IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 27, 2021, accepted January 13, 2022, date of publication January 18, 2022, date of current version January 28, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3144407*

# Autonomous Vehicles Perception (AVP) Using Deep Learning: Modeling, Assessment, and Challenges

**HRAG-HAROUT JEBAMIKYOUS, (Member, IEEE), AND RASHA KASHEF, (Member, IEEE)**
Department of Electrical, Computer, and Biomedical Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada
Corresponding author: Rasha Kashef (rkashef@ryerson.ca)

**ABSTRACT** Perception is the fundamental task of any autonomous driving system, which gathers all the necessary information about the surrounding environment of the moving vehicle. The decision-making system takes the perception data as input and makes the optimum decision given that scenario, which maximizes the safety of the passengers. This paper surveyed recent literature on autonomous vehicle perception (AVP) by focusing on two primary tasks: Semantic Segmentation and Object Detection. Both tasks play an important role as a vital component of the vehicle's navigation system. A comprehensive overview of deep learning for perception and its decision-making process based on images and LiDAR point clouds is discussed. We discussed the sensors, benchmark datasets, and simulation tools widely used in semantic segmentation and object detection tasks, especially for autonomous driving. This paper acts as a road map for current and future research in AVP, focusing on models, assessment, and challenges in the field.

**INDEX TERMS** Autonomous vehicle, deep learning, deep reinforcement learning, semantic segmentation, object detection, LiDAR, point cloud.

## I. INTRODUCTION

As technology constantly evolves, autonomous vehicles are becoming more popular, accessible, and affordable for more people in different countries and from different economic classes. Increasing accessibility results in a safer transportation experience, fewer deaths, and minimal injuries due to human-made mistakes that cause catastrophic accidents. To ensure the safety of individuals, it is necessary to deploy highly efficient and accurate learning models trained on a broad range of driving scenarios to precisely detect the surrounding objects under different weather and lighting conditions. This learning procedure via training will adjust the vehicle's decision-making process and control mechanism to take the necessary actions.

Autonomous Vehicle Perception (AVP) in driving systems collects the necessary information about the surrounding environment of the moving vehicle. The perception data is then fed to a learning model to obtain an optimum decision. The two main methods used in the perception of autonomous vehicles: Semantic Segmentation and Object detection; both tasks work primarily with images. Semantic segmentation is the process of assigning each pixel in an image to a particular class. These class labels could be a person, bicycle, tree, etc. Semantic segmentation is considered as an image classification task at a pixel level. Object detection is the task of identifying and locating an object of interest in an image and drawing a bounding box around that object.

Machine learning is used in many classification and categorization tasks in AVP [1]. Recently, Deep learning has widely been adopted in semantic segmentation and object detection. For example, the two semantic segmentation networks, Efficient Neural Network (ENet) [2], [3] and Segmentation Network (SegNet) [4], [5], utilize a compact encoder-decoder architecture. Both networks consist of an encoder and a corresponding decoder network followed by a pixel-wise classification layer. The commonly used object detection model used in the literature is called YOLOv2 (You Only Look Once) [6], [7]. This model performs object detection in one stage, making it fast. Another widely used Object Detection model is called Faster-RCNN [8], [9]; it consists of two networks: a Region Proposal Network (RPN) that ranks region boxes and proposes the boxes that are most likely to contain objects. Another network to detect objects using the generated proposals.

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

To the best of our knowledge, no research highlights the current state-of-the-art modelling methods in AVP, focusing on deep learning, assessment criteria, and challenges. Thus, the main contributions of this paper are:

1) Surveying the most recent research work on the two main methods used in the perception of autonomous vehicles: Semantic Segmentation and Object Detection.
2) Providing a comprehensive overview of the various deep learning method used in the AVP
3) Discussing the various evaluation metrics in AVP
4) Presenting and discussing the current challenges in AVP, including datasets, evaluation metrics, and modelling.

The rest of this paper is organized as: Section II discusses sensors types, Section III introduces Autonomous Vehicle Perception, Section IV presents related work on Semantic Segmentation, Section V discusses the state-of-the-art modelling methods in Object Detection, Section VI focusses on Deep Learning for AVP, the benchmark datasets are discussed in section VII, the performance evaluation metrics are introduced in section VIII; finally, conclusion and future directions are presented in section IX.

## II. ACTIVE AND PASSIVE SENSORS

Unlike human drivers that rely primarily on their auditory and visual systems to drive a car, the autonomous vehicle's perception relies on multiple sensors to overcome the limitations of individual sensors. The sensors can be divided into two categories: Active sensors, such as Radar, LiDAR, and Sonar emit energy into the surrounding environment and measure the reaction of the environment when the energy bounces back off each object in that environment to produce outputs, and Passive sensors, such as Stereo and Monocular Cameras receive the emitted energy from the surrounding environment to produce outputs. Most of the research work on autonomous vehicle perception is mainly focused LiDAR, Camera, and Radar sensors.

### A. CAMERAS

Any autonomous vehicle must be equipped with cameras because cameras can collect the richest information about the car's surrounding environment and objects. Monocular Cameras can provide shape and texture information, which is needed to detect and classify the lanes' shape and color (e.g., Broken white or double yellow), traffic light color classification, traffic sign recognition, and other object detection and classification tasks. However, this type of camera cannot provide the depth information needed to estimate the detected object's position and size. Thus, Stereo Cameras can retrieve the relative depth of each point.

### B. LIDAR

The LiDAR (Light Detection and Ranging) sensors emit laser pulses and receivers that receive the returned pulses. This sensor is widely used in autonomous vehicles to detect and recognize the object's class and accurately measure the distance and location of the object, regardless of the lighting and weather conditions. It measures the time taken to send and receive the pulse, which helps accurately determine the object's distance in each direction. It sends thousands of pulses per second to create a point cloud map (or a depth map), which provides a 360-degree view of the surroundings. LiDAR cannot be used as a standalone sensor because it is a depth-based sensor, and it cannot recognize the readings on the traffic signs, nor can it classify the colors of the traffic lights. Hence LiDAR Sensors will always be used in coordination with Camera sensors.

### C. RADAR

Radar sensors, which stands for Radio Detection and Ranging, have an antenna that emits radio signal in a specific direction and a receiver that detects the radio signal that has bounced off objects in the surrounding environment. The distance between the antenna and an object is determined by calculating the radio signal's time to and from the object. Radars can function better than other sensors in unpleasant weather conditions, such as snow, fog, and rain, and detect the car ahead. However, Radar tends to be less accurate than Camera and LiDAR and provides insufficient details for the perception of autonomous vehicles. Thus, it cannot be used to detect and classify objects accurately. Due to its limitations, Radars are used in very defined areas, and it is usually coupled with Camera and LiDAR sensors. The computer then pieces the gathered data from different sensors to create a coherent picture of the surrounding environment.

## III. AUTONOMOUS VEHICLE PERCEPTION

Perception is the ability of an autonomous system to extract important information from the environment. It is a fundamental task to enable autonomous driving; it provides crucial information about the driving environment, including the free drivable areas, the locations, velocities, and prediction of the future state of the surrounding obstacles. Autonomous vehicles use LiDAR and Camera sensors for their perception, as described in the previous section, to accurately detect obstacles and take the appropriate actions for a given scenario to avoid potential accidents. The essential tasks for a safe driving experience are Semantic Segmentation and Object Detection; these tasks are summarized next.

## IV. SEMANTIC SEGMENTATION

Autonomous vehicles rely heavily on semantic segmentation to navigate through routes. It operates by assigning each pixel in the image a particular class, and all the pixels that belong to a specific class are assigned a single color. As shown in Figure 1, vehicles are painted red, vegetation is painted green, buildings are painted grey, etc.

### A. SPATIAL AND SEMANTIC FEATURES

Spatial features can be represented in image or vector mode, containing spatial and location information. The spatial

**FIGURE 1.** Semantic Segmentation on Cityscapes dataset [40].

features can be defined as neighboring cells in the image mode, called regions. The spatial features can be defined as a line, point, or polygon in vector mode. In addition to the image and vector mode, spatial features can be found in the LiDAR data, collected via vehicles, satellites, drones, and other aerial devices. Spatial data is processed and analyzed using a Geographic Information System (GIS), a program or a combination of programs to enable users to manage, manipulate, analyze, customize, and create visual displays to make sense of the spatial data. Semantic features describe the visual contents of an image by correlating the content of the image scene with low-level features such as color. For instance, correlating the green color with trees, the blue color with sky and sea. In the autonomous driving scenario, the semantic features are the vehicles, road signs, traffic lights, lane markings, etc. The relationship between semantic features defines how the lanes work: when it can change lanes, where to stop, and which lanes to use to travel from points A to B.

### B. LITERATURE REVIEW

In [2], the authors tackled the problem of validating the performance of semantic segmentation algorithms under various operating conditions of autonomous vehicles, such as precipitation and illumination. Because even a slight variation in the environmental conditions could affect the classification performance and accuracy of the segmentation model, which can lead to catastrophic consequences. To solve this challenging problem, they proposed a pipeline that incorporated a Lidar sensor to test the performance of the semantic segmentation of a particular model in different real-world scenarios. They were able to distinguish the boundaries of the road around the vehicle. They automatically generated a large amount of ground truth road labels by testing the geometric properties of the surrounding Lidar points. They chose the 'Road' class from the semantic segmentation output to compare it with the ground truth generated by the Lidar sensor to prove the possibility of obtaining a measure of the classification performance and accuracy to validate the model. They also collected a weekly dataset of the area around their campus for 6 months to analyze the trained segmentation network performance and compare the validation accuracy of the model for datasets with different lighting and weather conditions. They used the NVIDIA DRIVE PX 2 computing platform, which is designed to accelerate the production of autonomous vehicles. They used the proposed validation pipeline to compare the performance of two different semantic models, namely ENet and Bonnet. By performing these comparisons, they concluded that the best model selection depends on the operating conditions, and the accuracy of the models varies depending on the dataset. The authors in [3] tackled the problem that current semantic segmentation models face: the edge of the detected object is not clear. The proposed method utilized EfficientNet as the backbone network, coord convolution is applied to low features to add the position information, because of this addition the performance of this method was higher than the existing semantic segmentation models, the experiment showed that the application of Direction Convolution led to a more accurate edge detection compared to existing techniques. The proposed method was validated on the 'Cityscape' dataset and resulted in a high performance, particularly on people and bicycles of different shapes. In [10], the authors tackled the need for a large computational resource for spatial-to-temporal approaches implemented in autonomous vehicles when tracking the various patterns of spatial positions for their motion. They proposed a temporal-to-spatial approach to cope with the vehicle's speed in autonomous navigation by sampling a 1-pixel line at each frame in the video. The temporal connection of lines from consecutive frames makes a road profile image consisting of vehicles, road, lane mark, roadside, etc., and turning and stopping of ego-vehicle. This approach reduces the processing data to a fraction of video to catch up with the vehicle driving speed. They used RGB-F images (where F is a channel that describes features around the sampling line) of the road profile to perform semantic segmentation to retrieve individual regions and their spatial relations on the road. They tested their proposed method on naturalistic driving video, and the results were promising. They used a single NVIDIA GTX 780Ti GPU to train and test the proposed model. A comparison of some of the current research work in semantic segmentation based on the used algorithm, available datasets, and the current challenges is provided in Table 1.

**TABLE 1.** Semantic segmentation approaches.

| Paper | Algorithm | Dataset | Problem |
|---|---|---|---|
| [2] | Enet & Bonnet | Cityscapes & USYD | Validating the performance of semantic segmentation algorithms under a variety of operating conditions |
| [3] | Efficient Net | Cityscapes | The edge of the detected object is not clear |
| [10] | Road Profile Semantic Segmentation | Self-collected | The need for large computational resources for spatial-to-temporal approaches |

## V. OBJECT DETECTION

Object detection is a fundamental task in any autonomous driving system, which identifies and locates object classes of interest in an image and creates a bounding box around those objects. Some popular object detectors include YOLOv2, YOLOv3 (Figure 2), and Viola-Jones algorithm. Others use more sophisticated deep learning-based models.

### A. POINT CLOUDS

Point clouds are a simple form of 3D models, a collection of points plotted in a 3D space. Each point represents several measurements, including the X, Y, Z coordinates, the color value stored in RGB format, and the luminance value, which determines the brightness of the point. Point clouds are created by scanning an object or structure using a laser scanner. Laser scanners work by sending laser pulses to the surface of an object and measuring the time taken for the pulse to return. These measurements are used to determine the exact position and the shape of the object. These points are then used to create a point cloud. As discussed in Section II, point clouds are collected using the LiDAR sensor in the autonomous driving scenario.

### B. RELATED WORK

A real-time classification based on the Real AdaBoost algorithm is introduced in [1]. Lidar 3D point clouds are used to compute various features of road objects. The proposed classifier achieved over 90% accuracy in a 50-meter range. This algorithm can be used for autonomous driving because it classifies an object in just $0.07 \times 10^{-3}$ seconds. The authors in [6] have tackled the problem of unreliable and noisy 3D maps generated by LIDAR sensors for precise mapping and localization of Autonomous vehicles due to the existence of moving objects in the map, which leads to bad localization. Their proposed system takes 3D points from LIDAR, camera images, and GPS/INS information as input and outputs a vehicle-free 3D point cloud map. They used YOLOv2 Vehicle Detection Network (YVDN) to find the bounding boxes of the vehicles in an image and used K-Frames forward-backward bounding box tracking algorithm to find the missing bounding boxes. The 3D points that fall into the detected bounding boxes are then removed from the LIDAR frame. They registered each vehicle-free LIDAR scan to a global coordinate based on the GPS data to reconstruct a vehicle-free 3D point map. They validated their proposed method on the Oxford RobotCar Dataset and proved that it could generate a precise vehicle-free 3D point cloud map. The network was trained on NVIDIA TITAN X GPU for 30 epochs. In [7], the authors built a system to detect the surrounding vehicles and warn the driver of potential collisions. The proposed method consisted of two parts is implemented in a Robot Operating System (ROS). The first part uses the YOLOv2 algorithm for vehicle detection in an autonomous vehicle environment and is configured to detect four different classes of vehicles: trucks, buses, vans, and cars. The second part uses two ROS nodes, the first node is used for distance assessment in the Carla simulator, and the second node is used for real-world distance assessment. The evaluation of the proposed method showed promising results. The algorithm runs at 40 FPS (close to real-time) on the NVIDIA GTX1060 (3Gb) graphics card. The authors [11] focused on object detection and tracking, an integral part of Advanced Driver Assistance Systems (ADAS). Object detection and tracking provide necessary information for collision avoidance, emergency braking, path planning, etc. The authors used two object detection algorithms: Viola-Jones and YOLOv3. The Viola-Jones algorithm was used to create nine object detectors classified under four groups: traffic light detector, pedestrian detector, traffic sign detector, and vehicle detector. Viola-Jones was compared with YOLOv3 based on their Precision, Recall, and processing speed. It was concluded that YOLOv3 achieved higher Precision and Recall and shorter processing time than Viola-Jones. They also used Median Flow tracking and Correlation tracking methods for object tracking. Median Flow tracking has a faster processing time, but both methods achieved similar results in terms of Multiple Object Tracking Accuracy (MOTA). They validated the proposed method on various datasets, such as German Traffic Sign Recognition (GTSR) Benchmark, INRIA Person, Udacity, and CARS Correlation tracking. Table 2 provides a comparison study for some related work in the literature of object detection.


**FIGURE 2.** Object Detection obtained by YOLOv3 model [7].

**TABLE 2.** Object detection methods.

| Paper | Algorithm | Dataset | Problem |
|---|---|---|---|
| [1] | Real AdaBoost | Self-collected | Real-time object classification using Lidar |
| [6] | YOLOv2 Vehicle Detection Network | Oxford RobotCar | 3D maps are noisy due to moving objects which leads to inaccurate localization of Autonomous Vehicles |
| [7] | YOLOv2 Vehicle Detection Network | Self-collected | Detect surrounding vehicles & warn the driver of potential collisions |
| [11] | Viola-Jones, YOLOv3, Median Flow, | GTSR Benchmark, INRIA Person, Udacity, CARS | Object detection and tracking |

## VI. DEEP LEARNING FOR AUTONOMOUS VEHICLE PERCEPTION

Deep learning is the backbone of every autonomous driving system, it is being used by object detection and classification algorithms (Supervised Learning) to detect and classify obstacles around the vehicle. It is also used for decision-making (Deep Reinforcement Learning) based on the observed data. Autonomous vehicles extensively use Convolutional Neural Networks (CNN), one of the most famous deep learning models. A CNN model consists of three main layers: A Convolutional Layer is used to extract features from the input image by convolving (dot product) the input image with a filter of size M x M, and it outputs a feature map. A Pooling Layer is often placed after the convolutional layer to reduce the size of the feature map, reducing the computational cost of the model. A Fully Connected layer consists of neurons along with weights and biases. It connects each neuron to all the neurons in the previous and the next layer. It takes the flattened image as a vector as its input and outputs the classification results.

### A. DEEP LEARNING FOR SEMANTIC SEGMENTATION

In [4], they argue that the existing Semantic Segmentation methods partition the images into several semantically meaningful parts to classify each part into one of the predetermined classes, ignoring the different importance levels of classes. For example, bicycles, other cars, and pedestrians are much important than the buildings or the sky in the scene when driving autonomously, so they need to be segmented as accurately as possible to avoid catastrophic incidents. They proposed 'Importance-Aware Loss' IAL to tackle this problem, emphasizing the importance of critical objects in an autonomous driving scene. The IAL is designed based on a hierarchical structure, such that classes with different importance levels are located on a different level of the hierarchy. They also derived the forward and backward propagation of the IAL on four deep neural networks, namely, FCN, ENet, ERFNet, and SegNet. And tested these four networks on the 'CamVid' and 'Cityscapes' datasets, which obtained improved segmentation results on the pre-defined important classes. All semantic segmentation models are trained on two K80 GPUs. Road lane marking and road edge detection on Lidar-based autonomous cars are addressed in [5]. This includes obstacle avoidance capability but cannot detect road lane markings. They solved this problem by installing and calibrating a low-cost monocular camera on a Formula-SAE electric car with a Lidar sensor. They first tested the system on video recording of local roads to ensure the feasibility of SegNet semantic segmentation. Then they tested on the Formula-SAE car with Lidar readings. The obtained results from the semantic segmentation performed on the CamVid dataset proved that lane markings and road edges could be classified using the proposed method. The SegNet model ran on an NVIDIA GTX Titan X GPU with a 480 x 360 resolution and resulted in an image output at 10 FPS segmented at each video frame. The authors in [12] address the lack of research in the real-time RGB-D fusion semantic segmentation domain, despite accessible depth information. They proposed a real-time fusion semantic segmentation network named RFNet. The encoder part consists of two independent branches to extract the features of the input RGB and Depth images separately. They chose ResNet-18 as the backbone model to extract the features from the input images due to ResNet-18's residual structure and moderate depth. Its small operation footprint makes it compatible with real-time applications. After every layer of ResNet-18, the output features from the Depth branch are fed to the RGB branch after the AFC module. The SPP produces feature maps with multiscale information by collecting the fused RGB-D features from both branches. Finally, they used up-sampling modules to restore the resolution of the produced feature maps with a direct connection from the RGB branch and skip the Depth branch. They also used multi-dataset training to incorporate small obstacle detection to enrich the recognizable classes, which will help detect unforeseen hazards in real-world scenarios. They used the 'Cityscapes' and 'Lost and Found' datasets to test their model, outperforming previous state-of-the-art semantic segmentation models on the 'Cityscapes' dataset with high accuracy. The proposed RFNet operates at 41.6 HZ with half-resolution Cityscapes images and 22 HZ with full resolution on a single GTX2080Ti GPU, suitable for autonomous driving. The authors in [13] proposed an encoder-decoder-based deep CNN model in semantic segmentation of autonomous vehicle scenarios. The proposed model architecture is based on the VGG16 model. The encoder part of the architecture like VGG16 consists of 13 convolutional layers with $3 \times 3$ filters. After each convolutional layer, the convolutional stride and the spatial padding are fixed to 1 pixel. To decrease the size of feature maps, Max-pooling layers are used. They used residual learning by performing element-wise addition and shortcut connection to preserve the context and spatial information. On the other side, the decoder part has a similar structure as the encoder, but with only a few differences, such as the convolutional layers are replaced by de-convolutional layers and the Max-pooling layers by Up-sampling layers. They validated their proposed model on two popular benchmark datasets, namely, 'Cityscapes' and 'CamVid.' The experiments incorporated comparative analysis with popular networks such as ENet and SegNet, proving that their model outperformed both ENet and SegNet. The experiments were conducted on NVIDIA Titan X GPU. In [14], the problem of accurate road marking extraction is discussed. Addressing the complexity of road marking, they used a Dense Feature Pyramid Network (DFPN) based deep learning model, which concatenates the deep feature channels with shallow feature channels to help the shallow feature maps with abundant image details and high resolution utilize the in-depth features. Their deep learning model was trained end-to-end on mobile laser scanning (MLS) point cloud to extract the road markings. They optimized the model using the focal loss function.

Experiments proved that the proposed method outperformed the existing state-of-the-art methods in instance segmentation of road markings. To train the model, four GPUs were used by 400k iterations. In [15], a 3D Semantic Segmentation of point clouds in urban areas using deep learning is introduced. They conducted a comparative study on three novel deep learning-based semantic segmentation algorithms, PointCNN, PointNet, and SPGraph. The algorithms were trained on an outdoor aerial survey point cloud dataset and were evaluated based on the overall accuracy. The evaluation showed that SPGraph, PointNet, and PointCNN achieved 83.4%, 83%, and 72.7% accuracy for 3D semantic segmentation, respectively. Various semantic segmentation models trained on different datasets experience performance gaps when applied to actual scene images. Training Task Conversion (TTC) and domain adaptation have originally been proposed to solve this gap. But even with TTC and domain adaptation, the performance is not as good as the original task model. To solve the challenge of completing TTC while maintaining good performance, the authors in [16] proposed a deep learning model named DLnet for TTC from image dataset-based training to actual scene image-based training. Experimental results show that DLnet can achieve state-of-the-art performance on four popular datasets and four actual urban scenes. The DLnet is trained on Geforce GTX1080Ti GPU, which took 67 hours to train the model on the Cityscapes dataset and 97 hours on actual scene images. In [17], they proposed a self-attention mechanism and bi-directional Gated Recurrent Unit (GRU) to extract contextual information to achieve better semantic segmentation performance of urban traffic scenes by considering information distributed in the long-distance image plane, long-distance sequence information, and feature space correlation. They also proposed a cascade refinement supervised method using two loss functions to achieve precise segmentation. Experimental results on four semantic segmentation datasets, CamVid, Cityscapes, Mapillary, and KITTI, have demonstrated outstanding performance. The experiment was implemented on a computer with two GTX1080 Ti GPUs. A ERFNet-based multi-task instance segmentation network is proposed [18] to segment both road objects and road lanes. The ERFNet approach allows real-time segmentation even with limited hardware by allowing feature sharing, which reduces the computational requirements of the overall segmentation architecture. Results on a dataset derived from the large-scale BDD100K dataset and in real scenarios proved the robustness of the proposed approach in semantically segmenting objects on roads and road lanes effectively and in real-time. The segmentation network is trained and tested on an NVIDIA TITAN Xp GPU. In [19], the authors proposed a network called ADFNe,t, a neural network with accumulated decoder features based on ENet and ERFNet. ADFNet is a simple and efficient model that operates in real-time by only using the decoder information without the skip connections between the encoder and decoder.

Experimental results on the cityscapes dataset proved that the proposed ADFNet outperforms the state-of-the-art networks and the baseline network (ERFNet). The ADFNet model is trained on an NVIDIA GTX1080 Ti GPU. Table 3 shows the trade-off between the performance of semantic segmentation models (in terms of Accuracy, mIoU, and Precision) and the processing time of each image.

**TABLE 3. Semantic segmentation performance & processing time.**

| Paper | Evaluation Metrics | Processing Time |
|---|---|---|
| [10] | Accuracy: 97%, mIoU: 0.5931 | 0.3s |
| [14] | Average Precision: 0.383 | 1.2834s |
| [16] | mIoU: 83.4 | 62ms |
| [17] | mIoU: 76.77, Accuracy: 95.66 | 0.518s |

## B. DEEP LEARNING FOR OBJECT DETECTION

### 1) SUPERVISED LEARNING

Supervised learning is an essential part of all autonomous driving systems. It is the process of training the machine learning models on a large number of labelled images, the labels being the bounding boxes with specific colors around each class of object. After training, the model will be able to detect and classify each object in the surrounding environment of the autonomous vehicle. The output classification will then be fed to the decision-making system to take the optimum decision, which will ensure the safety of the driver and all other cars and pedestrians. To solve the performance limitations of self-driving cars equipped with a single sensor in severe weather conditions, the authors in [20] proposed a fusion scheme that uses a millimetre-wave radar as the main sensor and a camera as the auxiliary sensor. To match the observed values of the target sequence, they used the Mahalanobis distance, and the data fusion is based on Joint Probabilistic Data-Association (JPDA) method. The target detection algorithm is based on Faster R-CNN architecture, and it is tested on actual sensor data gathered from a vehicle while performing real-time perception. Experimental results showed that the proposed fusion of radar and camera performs better than single sensor perception in severe weather, reducing the missed detection rate in such scenarios. The detection and tracking of dynamic objects (e.g., bikes, vehicles, and pedestrians) in autonomous driving scenarios are of utmost importance for reliable decision-making and smart navigation of autonomous vehicles. However, current vision-based tracking systems have limitations, such as their lack of ability to re-track after the object is lost. The authors in [21] tackle such limitations by building a dynamic object tracking system in 3D space. The proposed system combines a LiDAR and monocular camera-based 3D position tracking algorithm to track the dynamic objects using a Siamese segmentation network and a re-tracking mechanism (RTM) to resume tracking the object after it reappears in the camera view using the YOLO object detection algorithm. The authors tested this method in a real-world autonomous driving environment and achieved a 10HZ update rate for real-time performance. To perform all experiments, an NVIDIA Geforce GTX 2080 was used. Over the last few years, the number of bicycle collisions on sidewalks has increased.

Many researchers have developed algorithms to detect people and bicycles to prevent such incidents. However, those algorithms cannot distinguish between bicycle pushers and bicycle riders because they mainly rely on the shape of the bicycle or people's shapes. To solve this problem, the authors in [22] have proposed a CNN-based algorithm called VGG-16, which uses video frame images to detect pedestrians, bicycle riders, as well as bicycle pushers. The algorithm is trained on 15,000 images of pedestrians, bicycle riders, and pushers. It is evaluated on a video recorded by the authors on public roads to assess bicycle riders' detection rate, which achieved 80.7%. Pedestrian detection has improved significantly with the advancement of convolutional neural networks, but the detection of small-scaled pedestrians and occluded pedestrians has been a challenging problem. The authors in [23] have proposed a pedestrians detection method with a coupled network to address these two problems. The first sub-network is a gated multi-layer feature extraction network to generate discriminative features for pedestrian candidates to detect large-scale variations of pedestrians. The second sub-network uses a deformable regional region of interest (ROI) pooling to solve the occlusion problem in pedestrian detection. Experimental studies on the CityPersons dataset have shown the effectiveness of the proposed coupled framework, which achieved missing rates of 40.78% and 34.60% on detecting small and occluded pedestrians, outperforming the second-best performing model by 6% and 5.87%, respectively. The experiments are performed on a single TITAN X Pascal GPU. In [24], the authors proposed an object detection and identification method. They utilized 3-D Lidar data to generate object region proposals. Then, they mapped those candidates onto the image space from which the proposals' ROI (Region of Interest) is selected and input to a CNN model based on the VGG16 model to perform object recognition. Then, they combined the features of the last three layers of the CNN to extract multiscale features from the Region of Interests to precisely identify the sizes of every object in the scene. They evaluated the proposed model on the KITTI dataset and reached the following conclusions:

- The processing time of each frame is 66.79ms, which is suitable for real-time processing.
- 3-D Lidar produces 86 candidate object-region proposals, compared to a sliding window that produces thousands of candidates per frame.
- The average identification accuracy of pedestrians and cars is 78.18% and 89.04%, respectively.

The authors in [25] designed a real-time pedestrian detection system for autonomous vehicles using CNN. They designed the system from scratch without using any available libraries. They evaluated their model on three datasets: INRIA, PETA-CUHK, and real-time video input and achieved accuracy ranging between 96.73% and 100%. The experiment was executed on a laptop with 8GB RAM and NVIDIA Geforce 940MX Graphics Processor. Deploying advanced Deep Convolutional Neural Network (DCNN) detectors in autonomous vehicles with limited memory and computing power is a challenging task [26]. It is necessary to design lightweight and robust detectors to solve this problem. Recently, a novel algorithm has been proposed named 'Group Convolution' to make the detection network faster and lighter by reducing the floating-point operations. But the existing guidelines do not indicate the optimal number of groups in the Group Convolution to maximize the detection speed. This paper introduced three new guidelines to indicate the optimum number of groups needed to design a fast and lightweight detector and named this detection network 'DenseLightNet'. The proposed method runs three times faster than the existing state-of-the-art detector YoloV3 and has a weight of 10.1MB compared to the YoloV3's 247MB. The algorithm was trained on the NVIDIA Titan X GPU. A Deep Neural Network (DNN) based object detector called Single-Shot Detector (SSD) is designed in [27]. The SSD architecture consists of a base network and an auxiliary network. VGGNet is used as a base network for good quality classification, and the auxiliary network is used to predict detection at multiple feature maps. A non-maximum suppression follows the base and auxiliary networks to decide the final detections. The proposed method was evaluated on the KITTI dataset, and it outperformed the original object detection model based on precision by 6%. The experiments were run on NVIDIA Geforce GTX TITAN X GPU and achieved 29.4 FPS, verifying its feasibility of running in real-time. In [8], a method for simultaneous detection of people, vehicles, lanes, and non-motor vehicles using RGB-D images is discussed. The task consists of two parts: the detection of vehicles, people, and non-motor vehicles as a general detection task, and lane detection as a segmentation task. They used two separate networks to improve the accuracy and speed, the first network is called LaneNet to segment the lanes, and the second is Faster-RCNN to detect the rest. They introduced a real-time synchronization method with multi-GPU for both networks' separate training and simultaneous detection. The detection frame rate of the system reached 15 FPS with four 1080Ti GPUs. The system was evaluated on a self-collected dataset, achieving high accuracy. They also tested the system in a real-time scenario on the streets of China, which proved that the system could be applied in real-time autonomous driving. In [9], the authors address the two main tasks involved in tracking and localizing vehicles and objects surrounding an autonomous vehicle: detecting and classifying obstacles. They proposed a region-based convolutional neural network named Faster-RCNN trained with PASCAL VOC dataset to detect and classify obstacles such as pedestrians, vehicles, animals, etc. This method was implemented on an NVIDIA GeForce GTX 980 Ti GPU and achieved a detection frame rate of 10 FPS on a VGA resolution image frame. The achieved fast frame processing rate ensures the usability of this system on highways. They validated the detection and classification performance of the system on the KITTI and iRoads datasets. They concluded that the performance did not vary on different shapes, views of an object, and different climate and lighting conditions. In [28], a model to predict the future trajectory of the objects using the Gated Recurrent Unit (GRU) is introduced. This model understands the behaviour of the surroundings in a mixed scene of bicycles, vehicles, and pedestrians. Since these objects have different behaviours, they applied different models to different categories. The proposed method takes three observed trajectories with different time steps as its input and predicts an accurate future trajectory. The model was then compared with GRU and LSTM and resulted in a smaller Mean Absolute Error (MAE) and converged faster than GRU and LSTM. In [29], the brake-lights recognition problem is presented, focusing on deep learning. The ''Brake Lights Patterns'' (BLP) are learned using a Multi-Layer Perceptron (MLP) based classifier that classifies the vehicles in an image as ''Normal'' or ''Brake''. The authors explored road segmentation and novel vanishing point ROI determination methods to speed up the detection and improve the system's robustness. The validation results conducted on on-road videos collected by the authors have shown the efficiency and robustness of the proposed method. In [30], a comparative study on object recognition using deep convolutional neural networks (CNN) in autonomous vehicle environments is presented. They used four well-known CNN models, Faster R-CNN Inception V2, Faster R-CNN Resnet 50, SSD Inception V2, AND Faster R-CNN Resnet 101. These models were pre-trained on the COCO dataset, and they were retrained with the new dataset using transfer learning. The new dataset was formed using GRAZ-01 and GRAZ-02 datasets and consisted of 517 images of 10 objects: Cars, Bicycles, Pedestrians, and 7 traffic signs. The experimental results have shown that Faster R-CNN outperformed the model models, with an accuracy of 85.1%. A deep learning model is proposed for 3D object proposal generation and detection from point cloud data called PointRCNN [31]. The framework is composed of two stages: The first stage generates a small number of high-quality 3D proposals in a bottom-up manner by segmenting the point cloud data into background and foreground points, unlike previous methods that used to generate proposals by projecting point cloud to bird's view or from RGB images. The second stage transforms the segmented points in the first stage to canonical coordinates to learn much better local spatial features. Those spatial features are combined with global semantic features for accurate confidence prediction and box refinement. The experiments performed on the KITTI dataset showed that the proposed PointR-CNN architecture outperforms state-of-the-art methods by only using point cloud as its input data. For self-driving, a deep learning system can use LiDAR point clouds and depth image-based rendering (DIBR) [32]. The DIBR is used to generate parallax map information and obtain the depth image, which is then combined with LiDAR point cloud to repair the objects in the point cloud image. They also combined the Histogram Equalization and Optimal Profile Compression (HEOPC) with the accuracy of deep learning to optimize the color image enhancement. Based on the restored point cloud image, they used a cutting algorithm to divide the areas of interest, such as cars, people, and bus and trained a MobileNet-YOLO model to identify those three objects. Detecting 3D objects in point clouds is challenging [33]. This problem was previously solved by projecting a 3D point cloud into 2D images. This means transforming the 3D detection problem into 2D detection. This method produces multiple 2D detection tasks, which increases the complexity and limits the performance of the 2D detection algorithm. To solve this problem, the authors proposed using a Convolutional Neural Network (CNN) model to perform the 2D detection task because CNN can predict multiple classes of objects using the same network without using an individual detector for each class. They concatenated two early rejection networks with binary outputs before the detection network to improve the detection efficiency. Extensive experiments have shown that the proposed method achieved a competitive performance, with at least ten times the speed of the latest 3D point cloud detection methods.

#### 2) REINFORCEMENT LEARNING

Reinforcement learning is also a machine learning paradigm commonly used in autonomous driving systems. It has an autonomous agent which learns to improve its performance at a given task by interacting with its environment without the help of an expert. The agent takes action and receives a reward from the environment based on the usefulness of the action taken. The performance is measured based on the reward function Q or R, and the agent's primary goal is to maximize the function reward function. Deep Reinforcement Learning-based obstacle detection and autonomous navigation, named Deep Q Network (DQN,) on a simulated car in an urban environment, has received popular attention in the last few decades [34]. The model takes input camera and laser sensor data placed on the car's front end. They also designed a prototype of a cost-efficient high-speed car to run the algorithm in real-time. They placed a Hokuyo Lidar sensor and a camera on the car and used an Nvidia-TX2 GPU to run the deep learning models. In [35], the authors worked on autonomous vehicle learning simulation results to drive in a simple environment containing static obstacles and lane markings. The algorithm takes an image of the street captured by the car front camera as an input. It computes the Q values representing the rewards that correspond to future actions taken by the autonomous vehicle. The actions are angles through which the vehicles steer at a fixed speed. The system enforces the car to act with the highest reward (Q value). The simulation results showed a high accuracy achieved by the model by following the lanes and avoiding obstacles. The algorithm is trained on an 8-core Xenon CPU x2, 256 GB RAM, and NVIDIA P100 GPU x5. Vehicle speed control using Reinforcement Learning methods is addressed in [36]. Their main motivation was the instability of the Q-learning algorithm in some games in the Atari 2600. They used an algorithm called Double Q-learning to control the vehicle's speed based on the surrounding environment. They proposed a new method that depends on the direct perception approach called the integrated perception approach to construct the environment. Both low dimensional data processed from the sensors and high dimensional data with road information from the video make up the input of the Double Q-learning model. Experimental results have shown that the Double Q-learning algorithm outperformed the traditional Q-learning algorithm in terms of policy quality and value accuracy. The total model score is 271.73% times that of Q-learning. A collision avoidance system for autonomous vehicles based on Reinforcement Learning can learn from mistakes and readdress its movement accuracy [37]. They used the Q-learning method to record and update the Q-values in a table for different movements, which will be used by the autonomous vehicle to determine how and where to move. A deep neural network was used to learn the Q-value table, which encounters many situations from different actions performed by the autonomous vehicle. The input to the model is 10000 images captured by a depth camera placed on the car's front end. The model was trained for 9000 epochs and achieved an obstacle avoidance rate of 95%. The autonomous braking problem is analyzed and discussed in [38] through precise decision-making and control to reduce accidents. They proposed a Deep Reinforcement Learning-based autonomous braking system in emergencies. They considered three key factors: accuracy, efficiency, and passengers' comfort. These factors were fully satisfied by the proposed system. They designed a multi-objective reward function for compromising the passengers' comfort, the degree of the accident, and the achieved rewards of different brake moments. To solve the autonomous braking problem, they adopted an actor-critic (AC) algorithm called Deep Deterministic Policy Gradient (DDPG), which improves the system's efficiency and makes it stable in continuous control tasks. They evaluated the proposed method through extensive simulations, which proved its efficiency in driving safety, decision-making accuracy, and learning effectiveness. Table 4 shows the trade-off between the performance of object detection models (in terms of Accuracy, Precision, and Recall) and the processing time. In Table 5, a comprehensive comparative study is provided among the state-of-the-art deep learning methods in semantic segmentation and object detection.

**TABLE 4. Object detection performance & processing time.**

| Paper | Evaluation Metrics | Processing Time |
|---|---|---|
| [1] | Accuracy: 90% | 0.07 ms |
| [9] | Mean Average Precision: 0.905 | <100 ms |
| [11] | Precision: 0.9, Recall: 0.879 | 0.83sec |
| [20] | Precision: 0.895, Recall: 0.881 | 64ms |
| [24] | Mean Average Precision: 0.8361 | 66.79 ms |
| [25] | Accuracy: 98.13% | 46.01 ms |
| [29] | Accuracy: 89% | 36 ms |

## VII. DATASETS

Since autonomous vehicle perception relies heavily on various deep learning models, the need for a large amount of data is obvious. Object detection and segmentation require accurately labeled data and many images and LiDAR point

**TABLE 5. A comparison between deep learning models.**

| Paper | Used Algorithm | Dataset | Problems |
|---|---|---|---|
| [12] | RFNet | Cityscapes & Lost and Found | Lack of real-time RGB-D fusion semantic segmentation work |
| [13] | VGG16 & Residual Encoder-Decoder | Cityscapes & CamVid | Residual Encoder-Decoder Network for Semantic Segmentation |
| [4] | FCN, SegNet, Enet, ERFNet | Cityscapes & CamVid | Semantic Segmentation methods give the same importance to all classes |
| [5] | SegNet | CamVid | Lidar-based autonomous vehicles are unable to detect road markings and edges |
| [14] | DFPN | Self-collected | Road Marking Instance Segmentation Using MLS Point Clouds |
| [15] | PointNet, PointCNN, SPGraph | Fused 3D point cloud | 3D Semantic Segmentation of Large-Scale Point-Clouds in Urban Areas Using Deep Learning |
| [16] | DLnet | Cityscapes, ADE20K, LIP, PASCAL-Context | Training Task Conversion while maintaining good performance in semantic segmentation |
| [17] | Bi-directional GRU | CamVid, Cityscapes, Mapillary, KITTI | Improve traffic scene semantic segmentation using contextual information |
| [18] | ERFNet | BDD100K | Multi-task instance segmentation with limited hardware. |
| [19] | ADFNet | Cityscapes | Improve the performance of semantic segmentation models. |
| [20] | Faster R-CNN | Self-collected | Improve the perception of self-driving cars in severe weather conditions. |
| [21] | Siamese, YOLO | Self-collected | Detection and tracking of dynamic objects. |
| [22] | VGG-16 | Self-collected | Improve the detection of bicycle riders, bicycle pushers, and pedestrians. |
| [23] | CNN (VGG-16 as the backbone) | CityPersons, Caltech | Improve the detection of small-scale and occluded pedestrians. |
| [24] | VGG16 | KITTI | Object detection and identification using 3D Lidar |
| [25] | CNN | INRIA, PETA-CUHK | Pedestrian detection using CNN programmed from scratch |
| [26] | Dense LightNet | City, Pascal VOC | Limited computing power for advanced DCNN |
| [27] | Single-Shot Detector | KITTI | On-road object detection using DNN |
| [8] | Faster-RCNN, LaneNet | Self-collected | RGB-D based real-time multiple object detection and ranging system |
| [9] | Faster-RCNN | KITTI, iRoads | On-road obstacle detection and classification using deep learning to track in a high-speed AV environment |
| [28] | GRU, LSTM | KITTI | The trajectory of Prediction of Immediate Surroundings Using Hierarchical Deep Learning Model |

**TABLE 5.** *(Continued.)* **A comparison between deep learning models.**

| | | | |
|---|---|---|---|
| [34] | Deep Q Network | Simulated the model | Deep Reinforcement Learning for obstacle avoidance and autonomous navigation |
| [29] | CNN (AlexNet) | Self-collected | Appearance-based Brake-Lights recognition |
| [35] | Deep Q Network | Simulated the model | Deep Reinforcement Learning for obstacle avoidance and lane detection |
| [36] | Double Q-Learning | Simulated the model | Instability of the Q-learning algorithm in speed control of vehicles in some games in the Atari 2600 |
| [30] | Faster R-CNN | GRAZ-01, GRAZ-02 | A comparative study on different CNN based object detection models |
| [37] | Q-Learning | Self-collected | A Reinforcement Learning collision avoidance system |
| [38] | Deep Deterministic Policy Gradient (DDPG) | Simulated the model | Deep reinforcement Learning-based autonomous braking decision-making strategy in an emergency |
| [31] | PointRCNN | KITTI | 3D Object Generation and Detection from Point Cloud |
| [32] | MobileNet-YOLO | KITTI | Self-driving Deep Learning System based on Depth Image Based Rendering and LiDAR Point Cloud |
| [33] | CNN | UWA 3D, CMU Oakland 3-D Point Cloud, Washington Urban Scenes 3D Point Cloud | 3D point cloud object detection with multi-view convolutional neural network |

cloud data to cover broader driving scenarios. This section presents the commonly used real and simulated datasets for object detection and semantic segmentation in autonomous driving.

### A. THE KITTI DATASET

One of the largest and widely used benchmark datasets in the autonomous driving research community [8], [12], [25], [27], [37], [38] is the KITTI dataset [39], which provides LiDAR point clouds, stereo color, and grayscale pictures, and GPS coordinates. The data was captured on the highways and rural areas of a mid-sized city in Germany called Karlsruhe. The tasks that can utilize this dataset include 3D Object Detection, Visual Odometry, Stereo Matching, and Optical Flow. The Object Detection part of the dataset consists of 7,481 training and 7,518 test images, with annotated boxes around the objects of interest.

### B. THE CITYSCAPES DATASET

One of the diverse datasets for the semantic segmentation task in autonomous driving [1], [2], [7] is called Cityscapes dataset [40], collected from 50 different cities in different seasons (spring, summer, fall) with various weather conditions. It consists of 20,000 images with coarse annotations and 5,000 images with fine annotations of 30 different classes.

### C. THE CAMVID DATASET

Another dataset for semantic segmentation in [7], [10], and [11] is The Cambridge – driving Labeled Video Database (CamVid) [41]. It provides per-pixel semantic segmentation of over 700 images, 367 training, 101 validation, and 233 test images of 32 semantic classes. Many papers use simulation tools to generate training data with specific conditions and to train autonomous driving systems.

### D. THE CARLA TOOL

A widely used open-source simulation tool is called CARLA [42], which provides flexible and adjustable sensor and environmental configuration to generate simulated data. Configurations could include adjusting the lighting and the weather with various virtual sensors such as a ray-casting LiDAR sensor, depth and RGB cameras with the ground, truth frames. Table 6 summarizes the used benchmark datasets along with their configurations and applications.

### E. THE PASCAL DATASETS

The main dataset is called PASCAL Visual Object Classes (VOC) project [26]. It provides standardized image datasets for object class recognition. The project ran from 2005 to 2012. The PASCAL VOC 2010 [43] dataset contains 20 classes, its train/validation data has 10,103 images with 23,374 ROI annotated objects and 4,203 segmentations. The PASCAL-Context dataset [44] used in [16] is a set of additional annotations of PASCAL VOC 2010. It goes beyond the original semantic segmentation task by providing annotations for the whole scene with 400+ labels.

### F. THE BDD100K DATASET

The largest and the most diverse dataset for computer vision research in autonomous driving is the BDD100K dataset [45], used in [11]. As the name implies, the dataset consists of 100 thousand videos; each video is 40s long, 720p, and has a frame rate of 30fps. The videos were recorded in different states in the United States, covered different weather conditions and different times of the day. Each video comes with GPS/IMU information recorded by cell phones to show driving trajectories. The dataset is labelled at several levels: road object bounding boxes, lane markings, drivable areas, and full-frame segmentations.

### G. THE CITYPERSONS DATASET

The CityPersons dataset [46] is built upon the Cityscapes dataset [40] specifically for pedestrian detection; it is used in [23]. In the Cityscapes dataset, humans are labeled as a rider or person. In CityPersons, humans are classified based on their postures to four categories: rider, pedestrian, sitting person, and another person. It contains 5000 images with 35k person, and 13k ignored region annotations.

**TABLE 6.** Benchmark datasets.

| Dataset | # Images | Application | SENSORS | GROUP OF CLASSES |
|---|---|---|---|---|
| KITTI | 14999 | Object Detection | Two Cameras, LiDAR, GPS | Vehicle, Human, Void |
| CITYS CAPES | 25000 | Semantic Segmentation | Camera | Flat, Human, Vehicle, Constructio n, Object, Nature, Sky, Void |
| CamVid | 701 | Semantic Segmentation | Camera | Moving Objects, Road, Ceiling, Fixed Objects |
| PASCAL VOC 2010 | 10103 | Object Detection, Semantic Segmentation | Camera | Person, Vehicle, Animal, and Indoor objects |
| BDD100K | 100,000 videos | Object Detection, Semantic Segmentation | Camera, GPS/IMU | Vehicle, Sign, Person, Bike, Light, Rider, Lane Markings, Drivable Areas. |
| City Persons | 5000 | Object Detection | Camera | Rider, Pedestrian, Sitting Person, Other Person. |
| Caltech | 250,000 frames | Object detection | Camera | Pedestrian. |

### H. THE CALTECH DATASET

The Caltech dataset [47], [48] is built for the task of pedestrian detection. It is the largest dataset and is used in [23]. It consists of 10 hours of video with 640 x 480 resolution and is recorded at 30HZ from a vehicle driving through regular traffic in an urban environment. It includes 350,000 bounding boxes and 2300 unique pedestrian annotations in 250,000 frames. Temporal correspondences and occlusions are also annotated.

## VIII. PERFORMANCE EVALUATION METRICS

Performance evaluation is required to evaluate and optimize any machine learning model and compare it with other models. Different evaluation metrics are used in the literature; this section describes the most efficient and widely used metrics in semantic segmentation and objects detection tasks.

Intersection Over Union (IoU) matric, also known as Jaccard Index, is widely used to evaluate semantic segmentation and object detection models [10]. It computes the percent overlap between the ground truth bounding box (in object detection) or the target mask (in semantic segmentation) and the prediction output. As shown in Eq.1, IoU measures the number of common pixels between the prediction and target bounding boxes or masks and divides it by the total number of pixels present in both bounding boxes or both masks. Multi-class segmentation or detection tasks [2], [3], [7] use the mean Intersection Over Union (mIoU) metric for model evaluation, which first computes the IoU of each class and then computes the average overall classes.

$$IoU = \frac{Target \cap Predicted}{Target \cup Predicted} \tag{1}$$

Precision and Recall are also used as standard performance evaluation metrics [14], [24]. Precision represents the purity of the positive detections relative to the ground truth, which can be calculated using the following equation.

$$Precision\ (P) = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

The Recall represents the completeness of the positive predictions relative to the ground truth, which can be calculated using the following equation.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

Another commonly metric used to measure the detection accuracy is the mean Average Precision (mAP). It is calculated by computing the Average Precision (AP) of each class, then computes the average of all Average Precisions, as shown in the following equation.

$$Average\ Precision\ (AP) = \frac{\sum P\ \forall\ True\ Positive}{True\ Positive} \tag{4}$$

$$mean\ Average\ Precision\ (mA) = \frac{\sum AP\ \forall\ Classes}{Number\ of\ Classes} \tag{5}$$

## IX. CONCLUSION AND FUTURE DIRECTIONS

As the adoption of autonomous vehicles with different levels of autonomy increases, the need for precise and accurate perception systems increases drastically to ensure the safety of the passengers, pedestrians, and the surrounding vehicles' drivers. This survey concludes that deep learning models are essential for accurate object detection and semantic segmentation on images and point clouds because only deep learning models can learn the complex features and patterns in an image. It is also important to note the importance of training the models on versatile datasets collected under a variety of scenarios and weather conditions, which will play a crucial role in enabling the autonomous vehicle to make the right decision in a hazardous situation. This survey paper focuses on AVP tasks: Semantic segmentation and object detection as critical tasks in the perception process for autonomous vehicles; future work would involve other tasks in the autonomous driving system, such as planning, controlling, sensing, localization, perception, navigation, control, decision, and integrity monitoring. The future of autonomous driving relies on developing more robust algorithms trained on powerful computers like the system developed by Tesla called "Dojo", specifically designed for autonomous driving applications. Such computers would improve the machine learning models' efficiency, accuracy, and speed. Ensemble learning acts as a future direction in semantic segmentation, while hybrid learning is promoted for future research on object detection. The scalability of the existing methods is a great area of future investigation.

## REFERENCES

[1] M. Yoshioka, N. Suganuma, K. Yoneda, and M. Aldibaja, "Real-time object classification for autonomous vehicle using LIDAR," in *Proc. Int. Conf. Intell. Informat. Biomed. Sci. (ICIIBMS)*, Nov. 2017, pp. 210–211.

[2] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1951–1963, May 2020.

[3] J. S. Lee and T. H. Park, "Semantic segmentation with improved edge detail for autonomous vehicles," in *Proc. IEEE 16th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2020, pp. 520–525.

[4] B. Chen, C. Gong, and J. Yang, "Importance-aware semantic segmentation for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 137–148, Jan. 2019.

[5] K. L. Lim, T. Drage, and T. Braunl, "Implementation of semantic segmentation for road and lane detection on an autonomous ground vehicle with LIDAR," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Nov. 2017, pp. 429–434.

[6] M. Feng, S. Hu, G. Lee, and M. Ang, "Towards precise vehicle-free point cloud mapping: An on-vehicle system with deep vehicle detection and tracking," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 1288–1293.

[7] M. Gluhakovic, M. Herceg, M. Popovic, and J. Kovacevic, "Vehicle detection in the autonomous vehicle environment for potential collision warning," in *Proc. Zooming Innov. Consum. Technol. Conf. (ZINC)*, May 2020, pp. 178–183.

[8] J. Yang, C. Wang, H. Wang, and Q. Li, "A RGB-D based real-time multiple object detection and ranging system for autonomous driving," *IEEE Sensors J.*, vol. 20, no. 20, pp. 11959–11966, Oct. 2020.

[9] G. Prabhakar, B. Kailath, S. Natarajan, and R. Kumar, "Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving," in *Proc. IEEE Region Symp. (TENSYMP)*, Jul. 2017, pp. 1–6.

[10] G. Cheng, J. Y. Zheng, and M. Kilicarslan, "Semantic segmentation of road profiles for efficient sensing in autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 564–569.

[11] J. Ciberlin, R. Grbic, N. Teslić, and M. Pilipović, "Object detection and object tracking in front of the vehicle using front view camera," in *Proc. Zooming Innov. Consum. Technol. Conf. (ZINC)*, May 2019, pp. 27–32.

[12] L. Sun, K. Yang, X. Hu, W. Hu, and K. Wang, "Real-time fusion network for RGB-D semantic segmentation incorporating unexpected obstacle detection for road-driving images," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5558–5565, Oct. 2020.

[13] Y. G. Naresh, S. Little, and N. E. Oconnor, "A residual encoder-decoder network for semantic segmentation in autonomous driving scenarios," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 1052–1056.

[14] S. Chen, Z. Zhang, R. Zhong, L. Zhang, H. Ma, and L. Liu, "A dense feature pyramid network-based deep learning model for road marking instance segmentation using MLS point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 784–800, Jan. 2021.

[15] C. Lowphansirikul, K.-S. Kim, P. Vinayaraj, and S. Tuarob, "3D semantic segmentation of large-scale point-clouds in urban areas using deep learning," in *Proc. 11th Int. Conf. Knowl. Smart Technol. (KST)*, Jan. 2019, pp. 238–243.

[16] Y. Cai, L. Dai, H. Wang, L. Chen, and Y. Li, "DLNet with training task conversion stream for precise semantic segmentation in actual traffic scene," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 25, 2021, doi: 10.1109/TNNLS.2021.3080261.

[17] M. Yan, J. Wang, J. Li, K. Zhang, and Z. Yang, "Traffic scene semantic segmentation using self-attention mechanism and bi-directional GRU to correlate context," *Neurocomputing*, vol. 386, pp. 293–304, Apr. 2020.

[18] L. C. L. Bianco, J. Beltran, G. F. López, F. Garcia, and A. Al-Kaff, "Joint semantic segmentation of road objects and lanes using convolutional neural networks," *Robot. Auton. Syst.*, vol. 133, Nov. 2020, Art. no. 103623.

[19] H. Choi, H. Ahn, J. Kim, and M. Jeon, "ADFNet: Accumulated decoder features for real-time semantic segmentation," *IET Comput. Vis.*, vol. 14, no. 8, pp. 555–563, Dec. 2020.

[20] Z. Liu, Y. Cai, H. Wang, L. Chen, H. Gao, Y. Jia, and Y. Li, "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 24, 2021, doi: 10.1109/TITS.2021.3059674.

[21] L. Zhao, M. Wang, S. Su, T. Liu, and Y. Yang, "Dynamic object tracking for self-driving cars using monocular camera and LIDAR," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10865–10872.

[22] K. Ishii, S. Tsuichihara, H. Takemura, and H. Mizoguchi, "CNN-based system to identify bicycle riders and pedestrians: Toward minor collision prevention on sidewalks," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2020, pp. 718–721.

[23] T. Liu, W. Luo, L. Ma, J.-J. Huang, T. Stathaki, and T. Dai, "Coupled network for robust pedestrian detection with gated multi-layer feature extraction and deformable occlusion handling," *IEEE Trans. Image Process.*, vol. 30, pp. 754–766, 2021.

[24] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications," *IEEE Sensors J.*, vol. 20, no. 9, pp. 4901–4913, May 2020.

[25] K. Pranav and J. Manikandan, "Design and evaluation of a real-time pedestrian detection system for autonomous vehicles," in *Proc. Zooming Innov. Consum. Technol. Conf. (ZINC)*, May 2020, pp. 155–159.

[26] L. Chen, Q. Ding, Q. Zou, Z. Chen, and L. Li, "DenseLightNet: A light-weight vehicle detection network for autonomous driving," *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10600–10609, Dec. 2020.

[27] H. Kim, Y. Lee, B. Yim, E. Park, and H. Kim, "On-road object detection using deep neural network," in *Proc. IEEE Int. Conf. Consum. Electronics-Asia (ICCE-Asia)*, Oct. 2016, pp. 1–4.

[28] P. Y. Hsu, M. L. Huang, and H.-H. Chiang, "Trajectory of prediction of immediate surroundings for autonomous vehicles using hierarchical deep learning model," in *Proc. IEEE Eurasia Conf. IoT, Commun. Eng. (ECICE)*, Oct. 2020, pp. 263–266.

[29] J.-G. Wang, L. Zhou, Y. Pan, S. Lee, Z. Song, B. S. Han, and V. B. Saputra, "Appearance-based brake-lights recognition using deep learning and vehicle detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 815–820.

[30] G. Ozturk, R. Koker, O. ELDOgAN, and D. Karayel, "Recognition of vehicles, pedestrians and traffic signs using convolutional neural networks," in *Proc. 4th Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, Oct. 2020, pp. 1–8.

[31] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.

[32] G.-H. Lin, C.-H. Chang, M.-C. Chung, and Y.-C. Fan, "Self-driving deep learning system based on depth image based rendering and LiDAR point cloud," in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-Taiwan)*, Sep. 2020, pp. 1–2.

[33] G. Pang and U. Neumann, "3D point cloud object detection with multi-view convolutional neural network," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 585–590.

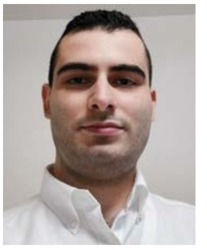
[34] A. R. Fayjie, S. Hossain, D. Oualid, and D.-J. Lee, "Driverless car: Autonomous driving using deep reinforcement learning in urban environment," in *Proc. 15th Int. Conf. Ubiquitous Robots (UR)*, Jun. 2018, pp. 896–901.

[35] T. Okuyama, T. Gonsalves, and J. Upadhay, "Autonomous driving system based on deep Q learnig," in *Proc. Int. Conf. Intell. Auto. Syst. (ICoIAS)*, Mar. 2018, pp. 201–205.

[36] Y. Zhang, P. Sun, Y. Yin, L. Lin, and X. Wang, "Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1251–1256.

[37] H.-T. Tseng, C.-C. Hsieh, W.-T. Lin, and J.-T. Lin, "Deep reinforcement learning for collision avoidance of autonomous vehicle," in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-Taiwan)*, Sep. 2020, pp. 2063–2068.

[38] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5876–5888, Jun. 2020.
[39] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[40] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[41] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Proc. ECCV*, Oct. 2008, pp. 44–57.
[42] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn. (CoRL)*, Oct. 2017, pp. 1–16.
[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
[44] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 891–898.
[45] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2636–2645.
[46] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3213–3221.
[47] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.
[48] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 304–311.



**HRAG-HAROUT JEBAMIKYOUS** (Member, IEEE) received the bachelor's degree in electronics engineering technology from Yorkville University, in 2018, and the M.Eng. degree from the Department of Electrical and Computer Engineering, Ryerson University. His research interests include using machine learning in the IoT, finance, and text analysis, and deep learning in autonomous vehicles and medical image analysis.

**RASHA KASHEF** (Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, in 2008. She was an Assistant Professor with the School of Computing, AAST Institute, from 2009 to 2011, a Research Associate at Microsoft Corporation, and a Postdoctoral Fellow with the Department of Applied Mathematics, University of Waterloo, from 2011 to 2013, where she also joined the Department of Management Science, from 2013 to 2016. She had been hired as an Assistant Professor with the Management Science Group, IVEY Business School, with a focus on data analytics, from 2016 to 2019. She is currently a Faculty Member with the Department of Electrical, Computer, and Biomedical Engineering, Ryerson University. She is a Professional Engineer in ON, Canada. Her research interests include the use of machine learning in big data analysis in different applications, including healthcare, revenue management, and software engineering. Her research expertise is in data science, machine learning, big data, the IoT, smart systems, operation research, management science, healthcare, autonomous systems, and distributed computing.