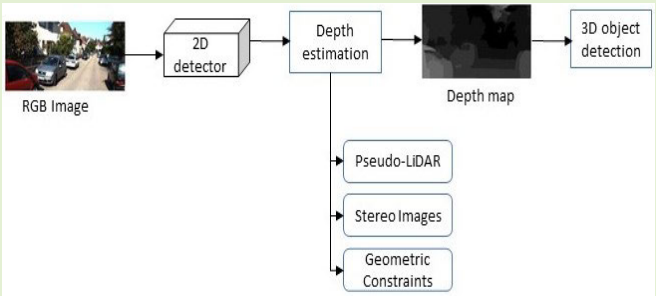
Deep Learning-Based Image 3-D Object Detection for Autonomous Driving: Review

Simegnew Yihunie Alaba, *Member, IEEE*, and John E. Ball, *Senior Member, IEEE*

***Abstract*—An accurate and robust perception system is key to understanding the driving environment of autonomous driving and robots. Autonomous driving needs 3-D information about objects, including the object's location and pose, to understand the driving environment clearly. A camera sensor is widely used in autonomous driving because of its richness in color and texture, and low price. The major problem with the camera is the lack of 3-D information, which is necessary to understand the 3-D driving environment. In addition, the object's scale change and occlusion make 3-D object detection more challenging. Many deep learning-based methods, such as depth estimation, have been developed to solve the lack of 3-D information. This survey presents the image 3-D object detection 3-D bounding box encoding techniques and evaluation metrics. The image-based methods are categorized based on the technique used to estimate an image's depth information, and insights are added to each method. Then, state-of-the-art (SOTA) monocular and stereo camera-based methods are summarized. We also compare the performance of the selected 3-D object detection models and present challenges and future directions in 3-D object detection.**



***Index Terms*— 3-D object detection, autonomous driving, camera, deep learning (DL).**

I. Introduction

AUTONOMOUS driving and robot navigation should obtain 3-D information on objects to understand the environment clearly. For fully autonomous driving, the perception system, such as 3-D object detection, needs to be robust to work in adverse weather, accurate to give precise information about the driving environment, and enable fast decision-making for high-speed driving [1]. Although 2-D object detection has shown significant performance improvement in the computer vision community due to the rapid growth of deep learning (DL), 3-D object detection is still a challenging problem due to the lack of 3-D information on sensors, scale changes, occlusions, and others. A robust perception system, including 3-D object detection, contributes to the development of fully autonomous driving, reducing fatalities caused by reckless human drivers. Building a perception system that is accurate to give precise information about the driving environment, fast to decide high-speed driving, and robust to work in inclement weather is crucial to achieving the goal of fully autonomous driving [1].

There are different 3-D sensors available for 3-D object detection, such as light detection and ranging (LiDAR), radio detection and ranging (radar), and depth sensors (RGB-D cameras) [2]. The LiDAR sensor is a good choice for distance measurement. It is also more robust to inclement weather than a camera. However, the LiDAR data are unstructured and sparse, making LiDAR processing more challenging. In addition, LiDAR is poor for color-based detection, and it is expensive. Radar is another 3-D sensor for distance measurement and velocity estimation, and is suitable for use in bad weather and night driving. However, it has low resolution, so radar-based object detection is poor. The camera sensor is inexpensive and rich in color and texture information. The major problem with a camera is the lack of high-accuracy depth information. Different DL-based methods have been developed to solve this problem. The monocular camera's lack of depth information can be partially solved using a stereo camera [3], [4] or structure from motion. Predicting stereo instance segmentation is another technique to solve the monocular depth problem for 3-D object detection [5]. In addition, a few works convert the image into Pseudo-LiDAR representation to solve the lack of depth information [6] (see details in Section IV).

Manuscript received 10 October 2022; revised 5 January 2023; accepted 6 January 2023. Date of publication 13 January 2023; date of current version 13 February 2023. The associate editor coordinating the review of this article and approving it for publication was Dr. Avik Santra. *(Corresponding author: Simegnew Yihunie Alaba.)*

The authors are with the Department of Electrical and Computer Engineering, James Worth Bagley College of Engineering, Mississippi State University, Starkville, MS 39762 USA (e-mail: sa1724@msstate.edu; jeball@ece.msstate.edu).

Digital Object Identifier 10.1109/JSEN.2023.3235830

The major contributions of this article are summarized as follows.

1) We provide an in-depth analysis of monocular and stereo image 3-D object detection methods.
2) We summarize 3-D bounding box (BBox) encoding techniques and object detection evaluation metrics. The BBox encoding and 3-D object detection evaluation techniques of each method are also provided in Section IV.
3) We categorize image 3-D object detection methods based on depth estimation techniques.
4) We present state-of-the-art (SOTA) image 3-D object detection methods for autonomous driving.

The rest of this article is organized as follows. Section II provides related work. Object detection, especially 3-D object detection, including object detection categories, 3-D BBox encoding techniques, and 3-D object detection evaluation metrics, is summarized in Section III. Section IV summarizes the image 3-D object detection methods and compares the selected ones. The challenges and future directions are presented in Section V. Section VI summarizes this survey article.

II. Related Work

The rapid growth of DL enables feature learning from images rather than handcrafted feature extractors, improving performance and facilitating the training process of object detection models. This work reviews DL-based image 3-D object detection models for autonomous driving. Most survey papers presented image 3-D object detection models with other works, such as LiDAR 3-D object detection methods. However, a tremendous number of papers are published each year. Thus, in this work, we present a detailed analysis of image 3-D object detection methods for autonomous driving. Therefore, we present SOTA methods and a comprehensive image 3-D object detection analysis for autonomous driving.

Kim and Hwang [7] reviewed a survey on monocular 3-D object detection, but the works are not specifically for autonomous driving. They presented deep DL-based monocular 3-D object detection methods and datasets. Feng et al. [1] reviewed 2-D and 3-D object detection and semantic segmentation for autonomous driving. The commonly used datasets and 2-D/3-D methods were reviewed. Jiao et al. [8] presented DL-based object detection methods, but not limited to autonomous driving. In addition, the survey focused more on 2-D object detection methods. Arnold et al. [2] briefly reviewed 3-D object detection methods, including LiDAR and image-based for autonomous driving. Similarly, Rahman et al. [9] presented 3-D object detection methods for autonomous driving. Li et al. [10] and Guo et al. [11] presented DL-based object detection, segmentation, and classification in autonomous driving. Fernandes et al. [12] also reviewed DL-based object detection and semantic segmentation for autonomous driving. Recently, Qian et al. [13] published a 3-D object detection method for autonomous driving. In addition to the current SOTA methods, we have included 3-D BBox encoding techniques and 3-D object detection evaluation techniques not covered by those survey papers. Recently, Alaba et al. [14] reviewed multisensor fusion 3-D object detection models, 3-D datasets, and sensors for autonomous driving (refer [14] for detailed analysis of more than 15 3-D datasets in autonomy, including stereo-based datasets).

III. Overview of Object Detection

This section presents object detection categories, evaluation metrics for object detection, and 3-D BBox encoding techniques.

A. Object Detection Categories

Image-based 3-D object detection models use 2-D object detection as a base model and use different techniques, such as regression, to extend to 3-D object detection. Thus, we review 2-D object detection models to understand 3-D object detection fully. DL-based general object detection methods can be classified into two groups: two-stage and one-stage. A two-stage object detection network has a region of interest (ROI) network for region proposal generation and the subsequent network for BBox regression and classification, as shown in Fig. 1. R-CNN [15], SPPNet [16], Fast R-CNN [17], Faster R-CNN [18], RFCN [19], and Mask R-CNN [20] are examples of two-stage 2-D object detection models. Girshick et al. [15] proposed R-CNN, a two-stage 2-D object detection network, as shown in Fig. 2.

The selective search algorithm [21] is used to generate 2000 region proposals (candidate boxes), and then, a CNN model is employed for feature extraction. The extracted features feed into support vector machines (SVMs) to classify an object within the region proposals. The major limitation of R-CNN was the redundant generation of 2000 BBoxs from each image, increasing the network's computational burden. He et al. [16] proposed spatial pyramid pooling networks (SPPNets) to overcome this problem by introducing a spatial pyramid pooling layer, which generates a fixed-length representation of an ROI. R-CNN and SPPNet train feature extraction and BBox regression networks separately. Thus, the training takes a long time to process.

Girshick [17] proposed the Fast R-CNN detector to solve the multistage training problem by simultaneously training the feature extraction and BBox regression networks. Fast R-CNN also uses a selective search algorithm for proposal generations. The selective search algorithm increases the computational burden of the model because of the redundancy of proposal generation. Thus, Fast R-CNN's detection speed is low for real-time applications. To solve this problem, Faster R-CNN [18] uses a region proposal network instead of the selective search algorithm to generate region proposals. Many improvements have been made based on Faster R-CNN, such as RFCN [19], Mask RCNN [20], Light head RCNN [22], and Feature pyramid Network [23]. Mask RCNN network combines the Faster R-CNN and the fully convolutional network (FCN) in one architecture with an additional binary mask to show pixels of the object in the BBox. There are also many 3-D object detection networks, such as Mono3D [24] (see Section IV for details on 3-D object detection).

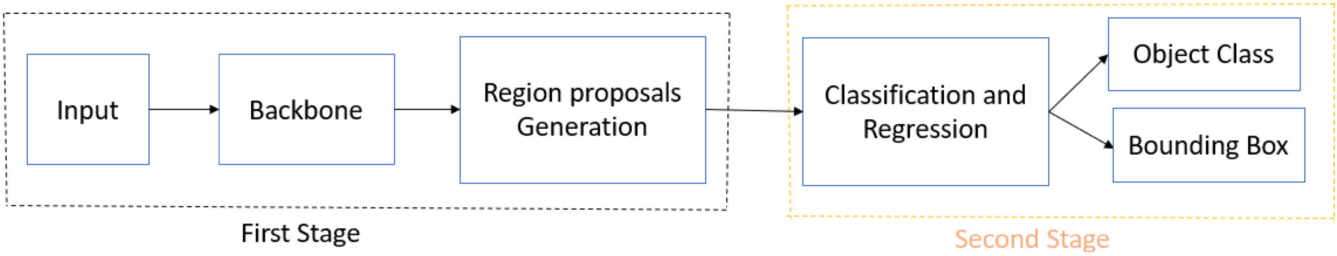


Fig. 1. Two-stage object detection architectural representation. The first stage generates the ROI, and then, the second stage predicts class probabilities and the BBox for each object. The backbone network and RPN can be designed as one network.

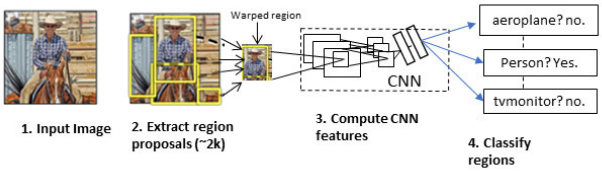


Fig. 2. R-CNN object detection system [15]. The system (1) takes an input image, (2) around 2000 bottom-up region proposals are extracted using a selective search algorithm, (3) for each proposal, features are computed using CNN and fed to the SVM classifier, and then, (4) linear SVMs classifies each region.

On the other hand, one-stage object detection networks directly learn the class probabilities and BBox coordinates in a single pass through the network without generating region proposals for each image. Fig. 3 shows the one-stage object detection general architecture. Redmon et al. [25] developed you only look once (YOLO), which is the first one-stage DL object detector. The network uses a single neural network to divide the image into regions and simultaneously predict each region's BBoxs and class probabilities, as shown in Fig. 4.

YOLO is faster than the two-stage object detection networks, but its accuracy is lower because of the class imbalance problem, a common problem for one-stage networks. YOLO struggles with small objects and groups of object detection. YOLO version 2 [26] improves YOLO by adding batch normalization on convolutional layers, increasing the resolution of images from 224 $\times$ 224 to 448 $\times$ 448, using anchor boxes instead of fully connected layers to predict BBoxs adopting multiscale training and others. The next versions of YOLO [27], [28] further improve detection speed and solve the accuracy bottlenecks. Similarly, Liu et al. [29] put forth a single-shot multibox detector (SSD), which is a one-stage detection network that improves the YOLO [25] accuracy bottlenecks and a small object detection problem by introducing aspect ratios and a multiscale feature map to detect objects at multiple scales. Then, Lin et al. [30] introduced RetinaNet to improve one-stage object detection by introducing focal loss (see the details from the paper [30]) as a classification loss function. The network's accuracy is comparable to the two-stage object detection while maintaining a high detection speed. Zhao et al. [31] proposed M2det, a multilevel feature pyramid network that enables the construction of multiscale and multilevel features, which helps to detect objects of different scales. Zhang et al. [32] introduced a RefineDet to further increase the accuracy of one-stage object detection. MoVi-3-D [33], [34] and AutoShape [35] are image 3-D one-stage object detection networks (see the details in Section IV).

One-stage object detection networks are fast, but their detection accuracy is lower than two-stage detectors due to class imbalance problems. On the other hand, two-stage detectors are slower than one-stage detectors; however, they have better detection accuracy. The RPN reduces redundant detections of two-stage detectors. However, one-stage detectors directly detect class probabilities and BBox estimation in a single pass without RPN, so the redundancy reduces the detection accuracy.

B. 3-D Bounding Box Encoding

Using perspective projection, one can estimate the 3-D BBox from the 2-D BBox. There are four commonly used 3-D BBox encoding techniques: the eight-corner method [36], the four-corner-two-height method [37], the axis-aligned 3-D center offset method [38], and the seven-parameter method [39], [40], as shown in Fig. 5. Mousavian et al. [38] proposed an axis-aligned 3-D center offset 3-D BBox encoding technique that combines DL with geometric constraints. The 3-D BBox is described by its center $T = [\Delta x, \Delta y, \Delta z]^T$, dimensions $D = [\Delta h, \Delta w, \Delta l]$, and orientation R $(\Delta\theta, \Delta\phi, \Delta\alpha)$, where $\Delta\theta$, $\Delta\phi$, $\Delta\alpha$, Δh, Δw, and Δl represent the azimuth angle, the elevation angle, the roll angle, the height, the width, and the length of the box, respectively. The elevation and roll angles are considered zero. Therefore, we can represent the 3-D BBox as $[\Delta x, \Delta y, \Delta z, \Delta h, \Delta w, \Delta l, \Delta\theta]$. The eight-corner box encoding method [36] regresses the oriented 3-D boxes from eight corners of 3-D proposals $(\Delta x0, \ldots, \Delta x7, \Delta y0, \ldots, \Delta y7, \Delta z0, \ldots, \Delta z7)$, which is a 24-D vector representation. Then, Ku et al. [37] developed four corners and two heights, representing the top and bottom corner offsets from the ground plane. The two heights are determined from the sensor height. Therefore, the 3-D BBox is represented as $(\Delta x1, \ldots, \Delta x4, \Delta y1, \ldots, \Delta y4, \Delta h1, \Delta h2)$.

Although the eight-corner encoding method gives better results than the axis-aligned method, it does not consider the physical constraints of a 3-D BBox [36]. Because of this, it forces the top corner of the BBox to align with the bottom corners. The four-corner and two-height encoding technique solves this problem by adding corner and height offset from the ground plane between the proposed BBoxs and the ground-truth boxes. Moreover, voxelnet [39] and SECOND [40] adopted the seven-point 3-D BBox encoding technique. The seven points are $(x, y, z, w, l, h, \theta)$, where x, y, and z are the

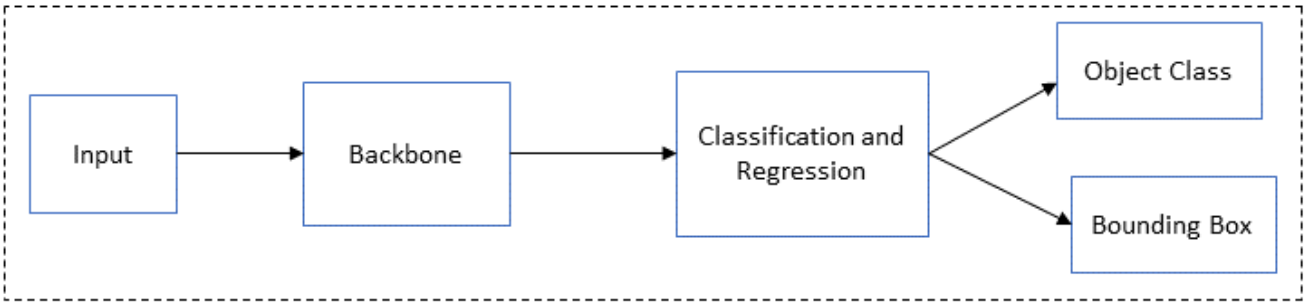


Fig. 3. One-stage object detection model architectural representation. The model learns the class probabilities and BBox regression in a single pass through the network instead of two passes like the two-stage model.

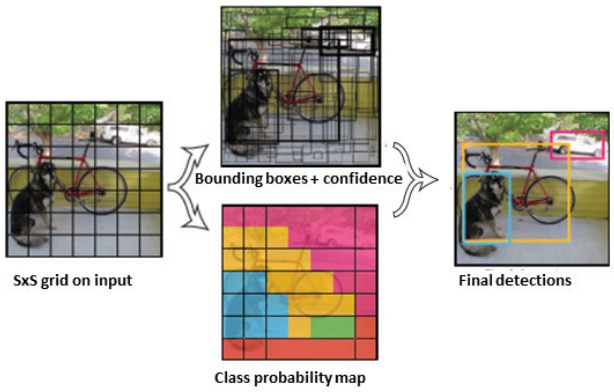


Fig. 4. YOLO model [25]. The model divides the image into an $S \times S$ grid. The model predicts BBoxs, a confidence score for each grid cell, and class probabilities for those boxes.

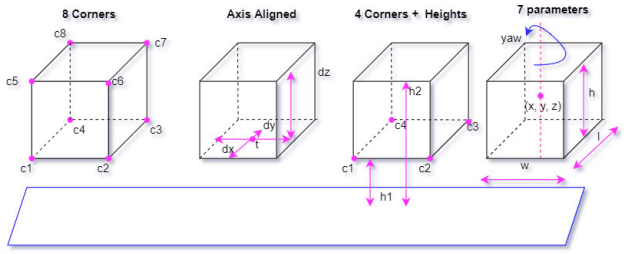


Fig. 5. Diagrammatic comparison between the eight-corner box encoding method [36], the four-corner and two-height encoding method [37], the axis aligned box encoding method [38], and the seven-parameter encoding method [39], [40].

center coordinates; w, l, and h are the width, length, and height, respectively. θ is the yaw rotation around the z-axis. The elevation and roll angles are considered zero. This encoding method is further adopted by pointpillars [41], WCNN3D [42], and monocular 3-D [24]. This technique is widely used in 3-D object detection. The regression operation between ground truth and anchors using the seven-point technique can be defined as

$$\Delta x = \frac{x^{\text{gt}} - x^a}{d^a}, \quad \Delta y = \frac{y^{\text{gt}} - y^a}{d^a}, \quad \Delta z = \frac{z^{\text{gt}} - z^a}{d^a}$$
$$\Delta w = \log \frac{w^{\text{gt}}}{w^a}, \quad \Delta h = \log \frac{h^{\text{gt}}}{h^a}, \quad \Delta l = \log \frac{l^{\text{gt}}}{l^a}$$
$$\Delta \theta = \sin(w^{\text{gt}} - w^a)$$

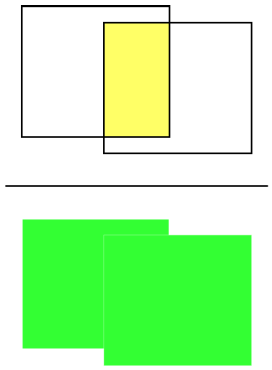


Fig. 6. Pictorial representation of IOU, best viewed in color. Top: intersection. Bottom: union.

where the superscripts gt and a represent the ground truth and the anchor boxes, respectively. $d^a = ((w^a)^2 + (l^a)^2)^{1/2}$ is the diagonal of the anchor box.

Energy minimization methods use a different 3-D BBox encoding technique. For example, Mono3D [43], 3DOP [44], and DeepStereoOP [4] represent a 3-D BBox as (x, y, z, θ, c, t), where (x, y, z) and θ denote the center of the 3-D BBox and the azimuth angle, respectively. c represents the object class, such as cars and pedestrians, and t denotes the set of 3-D box templates learned from the training data, which shows the physical size variation of each class.

C. Evaluation Metrics for Object Detection

One commonly used evaluation metric for object detection is average precision (AP) [45], which is an average detection precision under different recalls for each object category. The mean AP (mAP) is used as a final evaluation metric for performance comparison of overall object categories. The intersection over union (IOU) threshold value, a geometric overlap between the prediction and the ground-truth BBoxs, is used to measure the object localization accuracy. The graphical representation of IOU is shown in Fig. 6 (the yellow region represents the intersection of the predicted box and the ground-truth BBox, whereas the green region represents the union of the two). Equation (1) shows the mathematical expression of IOU. The representative threshold value may vary from object to object. For example, in the KITTI [45] dataset, a car's 3-D BBox requires an IOU of 0.7, and

pedestrians and cyclists require an IOU of 0.5

$$\text{IOU} = \frac{\text{bbox}_{\text{pred}} \cap \text{bbox}_{\text{gt}}}{\text{bbox}_{\text{pred}} \cup \text{bbox}_{\text{gt}}} \tag{1}$$

where $\text{bbox}_{\text{pred}}$ is the predicted BBox and bbox_{gt} is the ground-truth BBox. In addition, the F1 score and the precision–recall curve are used as evaluation metrics for classification. Precision shows the ratio of the true positives (TPs) to the total datasets' actual values, whereas recall reveals the ratio of the TPs to the predicted values. The balance of the precision–recall is important for AP and mAP. AP approximates the precision/recall curve shape by averaging precision for R equally spaced recall levels [46]

$$\text{AP}|R = \frac{1}{|R|} \sum_{r \in R} \rho_{\text{interp}}(r). \tag{2}$$

For the KITTI dataset, it is calculated for 11 equally spaced recall levels [46], [47], i.e., $R_{11} = (0, 0.1, 0.2, \ldots, 1)$. When the recall interval is zero, the correctly matched prediction gives 100% precision at the bottom recall bin [46]. The interpolation function $\rho_{\text{interp}}(r)$ is defined as

$$\rho_{\text{interp}}(r) = \max \rho(\tilde{r}), \tilde{r} : \tilde{r} \geq r \tag{3}$$

where $\rho(r)$ is the precision at recall r. The maximum precision value at recall greater or equal to r is considered rather than the mean of the whole observed precision values for each point r.

The mAP is calculated for the overall performance evaluation for 11 recall points. Some works, such as MonoPair [48], [46], calculate mAP using 41 points instead of 11 recall points but averaging only 40 (1/40, 2/40, 3/40, . . . , 1), without zero recall points to eliminate the glitch at the lowest recall bin [46]. The other common performance evaluating metrics are AP3D metric, average orientation similarity (AOS) metrics [45], and the localization metrics (AP_{BV}) [36] for bird's-eye view representation. AOS measures the 3-D orientation and detection performance by weighting the cosine similarity between the estimated and ground-truth orientations

$$\text{AOS} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} s(\tilde{r}), \tilde{r} : \tilde{r} \geq r \tag{4}$$

where $r = (\text{TP}/(\text{TP} + \text{FN}))$ is the recall based on the PASCAL [47] dataset. TP is a true positive, and FN is a false negative. The orientation similarity $\in [0, 1]$ at recall r is normalized by the cosine similarity

$$s(r) = \frac{1}{|D(r)|} \sum_{i \in D(r)} \frac{1 + \cos \Delta_\theta(i)}{2} \delta_i \tag{5}$$

where $D(r)$ denotes the set of all object detections at recall rate r, $\Delta_\theta^{(i)}$ is the difference in angle between estimated and ground-truth orientation of detection i, and the $\delta(i)$ term penalizes multiple detections.

On the other hand, the nuScenes [49] AP method defines a match by thresholding the 2-D center distance d on the ground plane rather than IOU. This helps to decouple the effect of object size and orientation for detection

$$\text{mAP} = \frac{1}{|C||D|} \sum_{c \in C} \sum_{d \in D} \text{AP}_{c,d} \tag{6}$$

where $D = \{0.5, 1, 2, 4\}$ m and C is the set of classes. For the nuScenes dataset, they measure a set of TPs for each prediction matched with the ground-truth box. Then, for each TP, the mean TP (mTP) is computed for the overall classes

$$\text{mTP} = \frac{1}{|C|} \sum_{c \in C} \text{TP}_c. \tag{7}$$

Finally, the nuScenes detection score (NDS) is computed

$$\text{NDS} = \frac{1}{10} \left[5\text{mAP} + \sum_{\text{mTP} \in \text{TP}} (1 - \min(1, \text{mTP})) \right]. \tag{8}$$

The NDS is an evaluation metric for the nuScenes dataset.

The Waymo open dataset [51] uses a 3-D object detection evaluation metric, APH, by incorporating heading information into common evaluation metrics, such as AP

$$\text{AP} = 100 \int_0^1 \max \left\{ p(r') | r' \geqslant r \right\} dr \tag{9}$$

$$\text{APH} = 100 \int_0^1 \max \left\{ h(r') | r' \geqslant r \right\} dr \tag{10}$$

where $p(r)$ is the precision/recall curve. In addition, $h(r)$ is computed similar to $p(r)$, but each TP is weighted by heading accuracy, which can be defined as $\min(|\tilde{\theta} - \theta|, 2\pi |\tilde{\theta} - \theta|)/\pi$, where $\tilde{\theta}$ and θ are the predicted heading and the ground-truth heading in radians within $[-\pi, \pi]$, respectively (refer the Waymo open dataset [51] for details). Most autonomy datasets follow either the KITTI or nuScenes evaluation metric.

IV. Image 3-D Object Detection Methods and Comparison of Various Methods

Image-based object detection methods use images as input. In this section, we review the monocular image and stereo image-based methods. 2-D object detection is successfully implemented for many applications, but it is not enough for autonomous driving applications. The autonomous vehicle (AV) must clearly understand the driving environment for reliable driving. Because of the lack of accurate depth information, 3-D object detection is more challenging for image-based methods. Different methods have been proposed to estimate depth from 2-D images to detect objects in 3-D using the estimated depth. Some of these methods use two-stage object detection methods by first generating object proposals and performing regression for 3-D BBox detection and classification. The classic object detection methods use handcrafted methods to generate 2-D box proposals [52], [53], [54], [55]. Others use the ability of deep neural networks to learn complex features from images to generate 2-D box proposals [56], [57]. Similarly, the box proposals can be generated from geometric constraints [38], [58], Pseudo-LiDAR [50], [59], or stereo depth estimation [3], [43].

Image-based 3-D object detection is more challenging due to the lack of depth information. Most depth estimation

Fig. 7. Generating Pseudo-LiDAR representation from given stereo or monocular images by predicting the depth map and projecting it into a 3-D point cloud coordinate system [50].

techniques can be categorized into Pseudo-LiDAR, stereo images, or geometric constraint-based, such as the object's shape and key points to estimate the depth. The Pseudo-LiDAR methods generate point cloud data from images and use 3-D LiDAR-based methods for detection. Although these methods outperform image-only methods, their accuracy is still lower than LiDAR-based methods because of the image-to-LiDAR generation error. The stereo image-based methods use the left and right image disparity to estimate the depth estimation. These methods also improve the 3-D object detection performance than the single-image methods. Some works also generate stereo images from a single image by generating a virtual image, which outperforms single-image methods. Other works use geometric constraints to estimate the depth information of a single image.

A. Pseudo-LiDAR Method

Some works convert monocular or stereo images into a LiDAR representation called Pseudo-LiDAR to solve the lack of depth information, such as [6], [24], [50], [59], [60], and [61]. Pseudo-LiDAR is a LiDAR representation of images by predicting the depth of each image pixel, called the depth map. Wang et al. [50] showed that the representation of the data plays a big role rather than the quality of the data on 3-D object detection by converting monocular images into LiDAR representation (Pseudo-LiDAR). The stereo depth estimation was done by using pyramid stereo matching network (PSMNet) [62], DISPNET [63], and SPS-STEREO [64], but they use DORN [65] as a monocular depth estimator. Then, the depth map is projected into a 3-D point cloud to produce Pseudo-LiDAR by mimicking the LiDAR signal, as shown in Fig. 7. The LiDAR-based detectors can directly process the Pseudo-LiDAR data. AVOD [66] and Frustum PointNet [67] LiDAR-based models were used for the experiment. The experimental result on the KITTI [45] dataset showed that Pseudo-LiDAR representation is more than adequate for 3-D object detection than image-only implementations.

Similarly, Ma et al. [59] convert RGB images into Pseudo-LiDAR and use pointNet as a backbone network to get objects' 3-D locations, dimensions, and orientations for each ROI. The proposed model consists of 3-D data generation and box estimation stages, as shown in Fig. 8. In the first stage, 2-D detection and point cloud representation are generated using two deep CNN backbones. In the second phase, two modules are designed for background points' segmentation and aggregation of RGB information to improve detection. Then, each ROI's 3-D location, dimension, and orientation are predicted using PointNet as a backbone. The proposed multimodal features fusion module is also used to fuse the complementary RGB image cues and the generated point clouds to improve performance. Xu and Chen [68] developed a fusion-based model for 3-D object detection by estimating the object class, 2-D location, orientation, dimension, and 3-D location based on a single monocular image. They use the MultiBin [38] architecture to obtain the pose of a 3-D object and then compute the point cloud representation. The estimated depth is encoded as a front view feature and fused with the RGB image to improve the input. Finally, features extracted from the original input are combined with the point cloud to increase the detection performance. Although converting images to Pseudo-LIDAR takes extra processing, Pseudo-LiDAR methods significantly improve performance over image-only methods.

Weng and Kitani [24] proposed a two-stage detection network based on Pseudo-LiDAR representation by using DORN [65] as a monocular depth estimator. They used instance mask 2-D proposals rather than BBoxs to reduce the number of points not belonging to the object in the point cloud. They train the network with an extended two-stage 3-D LiDAR detection algorithm Frustum PointNets [67]. The 2-D–3-D BBox consistency constraint was proposed to reduce noise in the Pseudo-LiDAR representation and handle local misalignment. The noise instance mask 2-D proposal representation and 2-D–3-D BBox consistency constraint improve the performance over [50], [68] by 6% and 21.2%, respectively. Similarly, OCM3D [69] is an object-centric monocular 3-D object detection model designed to reduce the noise level of Pseudo-LiDAR data by building voxels for each object proposal. The 3-D spatial points' distribution adaptively determines the voxel size and allows the point clouds' noise to organize effectively in the voxel grid. The model outperforms the previous models, such as RTM3D [70] on the KITTI [45] dataset.

Chong et al. [71] put forth Monodistill, a monocular 3-D object detection model. The LIDAR data were projected into the image plane and then trained on the LiDAR Net 3-D detector. Finally, LIDAR Net served as a teacher network for knowledge distillation to the baseline monocular model. The experimental result on the KITTI [45] dataset shows the method boosts the base model's performance. Reading et al. [72] proposed a categorical depth distribution network (CDDN) for monocular 3-D object detection. The frustum feature network projects image information into 3-D space and constructs a frustum feature grid. Then, the pointpillars [41] detection head performs 3-D object detection.

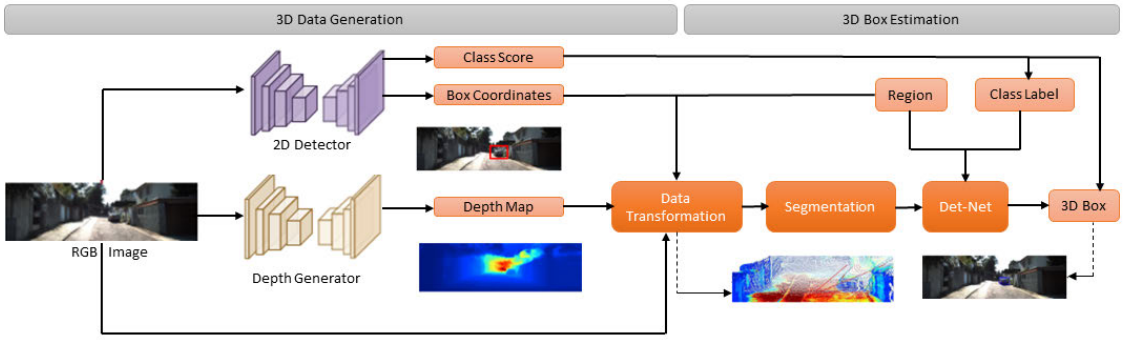


Fig. 8. Monocular 3-D object detection [59]. In the first phase, the two backbones generate 2-D detection and point cloud generation by estimating depth from RGB images. In the 3-D box estimation step, the pointNet backbone network generates each ROI's 3-D location, dimension, and orientation.

The model used the KITTI [45] and Waymo [51] datasets for the experiment. Vianney et al. [61] proposed a supervised and unsupervised preprocessing scheme to generate refined Pseudo-LiDAR data from depth maps before feeding into a 3-D object detection network. Qian et al. [60] put forth an end-to-end framework based on a differentiable change of representation (CoR) network to train the depth estimation and 3-D object detection.

Some methods convert stereo images to Pseudo-LiDAR representation to improve detection performance. Zhou et al. [73] put forth the SGM3D model that leverages the stereo representation to improve the performance of monocular 3-D object detection. The authors used a pretrained stereo-matching model PSMNet [62] for depth learning. Pixels are converted into 3-D pseudopoint clouds based on the estimated depth and camera instincts. The multigranularity feature alignment (MG-FA) module is proposed to get a consistent intermediate feature representation and the predictions per anchor between the output from the stereo-based and monocular approaches. An IOU matching-based alignment (IOU-MA) module is also introduced to reduce the mismatches between stereo and monocular predictions. Experimental results on the KITTI [45] and Lyft [74] datasets show a performance improvement. Pseudo-LiDAR++ [6] is an end-to-end depth learning approach using a stereo depth estimation network rather than disparity estimation. The graph-based depth correction algorithm concatenates the learned dense stereo depth and the sparse LiDAR signal for further depth refinement. The result improves 3-D object detection, especially faraway object detection.

Chen et al. [75] proposed the Disp R-CNN 3-D object detection model from stereo images, which has three stages. In the first stage, Mask R-CNN [20] detects images' 2-D BBoxs and instance segmentation. The instance disparity estimation network (iDispNet) estimates an instance disparity map in the second stage. Finally, an instance point cloud is generated from the instance disparity map and is the input to the detector head for 3-D BBox regression. The experimental result on the KITTI [45] dataset shows a promising result.

Converting monocular or stereo images into Pseudo-LIDAR improves 3-D object detection over image-only methods; however, the performance is lower than LiDAR-based methods because of the error from image to LiDAR conversion. Therefore, although converting image data into Pseudo-LiDAR representation takes extra processing, it is a good option when LIDAR data are not readily available.

B. Stereo Images' Method

These methods generate depth from stereo images [3], [4], [5], [43], [76], [77], [78], [79]. Mono3D [43] uses stereo images to estimate the depth and generate 3-D BBox object proposals by encoding object size priors, ground planes, a variety of depth-informed features, point cloud densities, and distance to the ground. The problem is articulated as an energy minimization function, and the Markov random field (MRF) is used to score 3-D BBoxs for proposal generation. Fast R-CNN [17] is used to predict the class proposal, and objects' orientation is estimated using the top object candidates. Chen et al. [44] extended the previous work [43] to generate class-specific 3-D object proposals (3DOPs) with very high recall for various IOU thresholds by assuming that objects should be on the ground plane and using only a single monocular image. They use semantic and object instance segmentation, context, shape features, and location priors to score 3-D BBoxs, as shown in Fig. 9. The limitation of 3DOP is that it should run separately for each object class to achieve a high recall. This operation increases the processing time because of many generated object proposals. To overcome this problem, Pham and Jeon [4] introduced a proposal reranking algorithm, DeepStereoOP, to rerank the generated 3DOPs. This algorithm helps achieve high recall and good localization using only a few candidate proposals. The two-stream CNN algorithm uses RGB features, depth features, disparity maps, and distance to the ground to rerank top-ranked candidates. The result shows that the DeepStereoOP algorithm is superior to the Mono3D [44] algorithm to get high recall with fewer proposals.

Chen et al. [3] presented a proposal generation algorithm using stereo imagery and contextual information. The 3DOPs are generated using an energy minimization function that encodes object size priors, ground plane information, and depth-informed features, such as free space, point cloud densities, and distance to the ground. The CNN scoring network uses appearance, depth, and context information to predict 3DOPs and poses simultaneously. The result outperforms the previous works, such as [4] and [44],

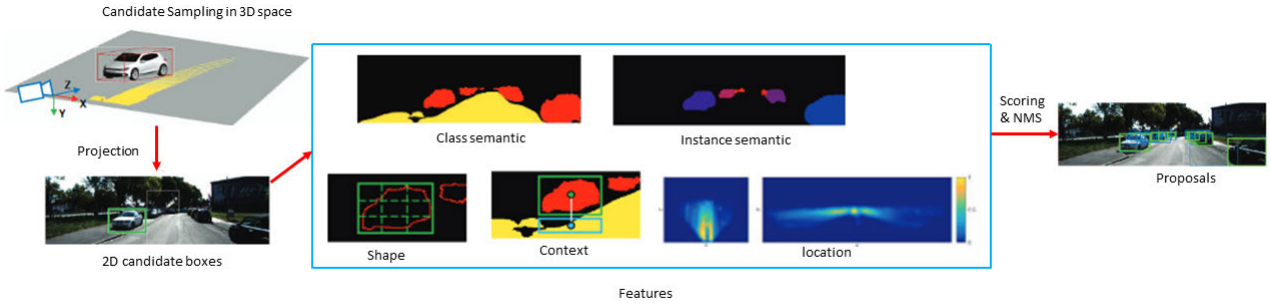


Fig. 9. Mono3D [44]. The 3-D BBoxs are sampled before being projected to the image representation. Scoring and NMS are done from multiple features: object shape, class semantic, location prior, instance semantic, and context.

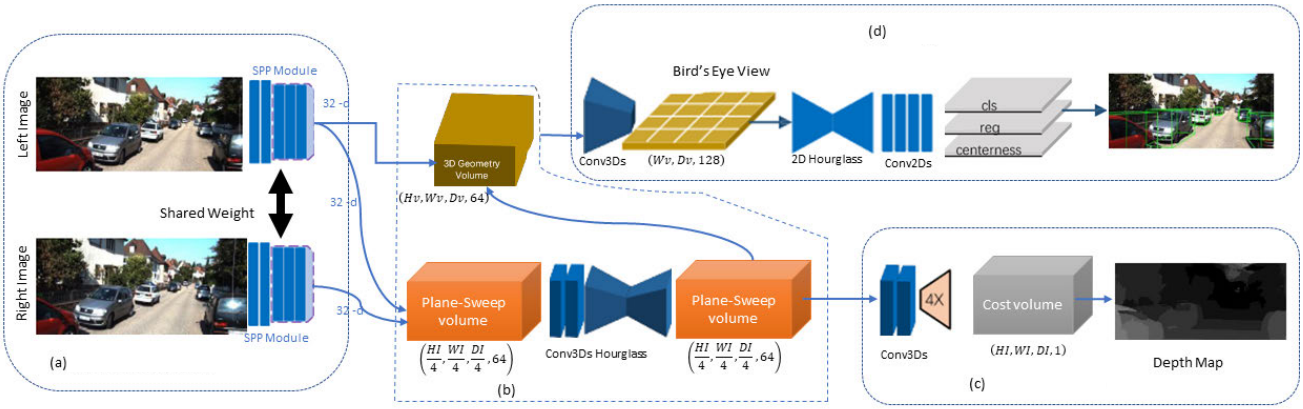


Fig. 10. DSGN architecture [79]. The model consists of four subnetworks. (a) 2-D feature extractor network captures pixel- and high-level features. (b) PSV and 3DGV are constructed in this subnetwork. (c) Depth estimation component on the PSV. (d) 3-D object detection component.

on the KITTI dataset. Königshof et al. [80] put forth 3-D object detection method using stereo image and semantic information. The semantic map and optional BBox suggestions are generated from the left image using ResNet-38 [81]. The model was trained and tested on the KITTI [45] dataset. Li and Chen [82] proposed S3D-RCNN, a two-stage joint stereo 3-D object detection and shape estimation model from a pair of stereo RGB images. The authors presented a global–local framework to decouple object pose estimation from object shape. The model showed a significant performance improvement on the KITTI [45] dataset.

Li et al. [83] developed an extended Faster R-CNN [17] based 3-D object detection method, Stereo R-CNN, to detect and associate objects in left and right images simultaneously by using the sparse, dense, semantic, and geometry information in stereo imagery. After generating left and right ROI proposals, left-right ROI features of object classes are concatenated and regress 2-D stereo boxes, viewpoint, and 3-D dimensions. They predicted a key point using only the left features combined with 2-D stereo boxes for 3-D box estimation. Peng et al. [78] put forth an instance-depth-aware module, Ida-3-D, as a depth estimation method of a 3-D BBox's center using instance-depth awareness, disparity adaptation, and matching cost reweighting. The channel and cost reweighting methods are essential to enhance features and weaken noisy signals using left-right coherence.

DSGN [79] is a one-stage end-to-end stereo-based 3-D object detection model that jointly estimates the depth and detects 3-D objects. The feature extractor component learns pixel- and high-level features from left and right images, as shown in Fig. 10. Then, the plane-sweep volume (PSV) and 3-D geometric volume (3DGV) are generated. The network's depth estimation component estimates the PSV depth. Finally, the 3-D object detection component predicts the object and BBox information. Chen et al. [84] proposed DSGN++, an extended version of DSGN, to improve the depth estimation technique. The three main aspects of DSGN++ models improved the DSGN model. First, the proposed depthwise plane sweeping (DPS) module extracts depth-guided stereo features. Second, the dual-view stereo volume (DSV) module allows multiple view connections of features, and top and front views. Finally, the proposed cross-modality data editing–copy-paste strategy ensures multimodal alignment, increases the foreground regions' dominance in 3-D, and improves data efficiency. The model was trained and tested on the KITTI dataset.

The confidence-guided stereo (CG-Stereo) 3-D object detection [85] model was proposed to improve the depth estimation accuracy. The model uses different decoders for foreground and background pixels while the depth estimation step. It also uses the confidence score of depth estimation network output to improve the depth estimation accuracy. The model outperforms previous models, such as DSGN [79] on the KITTI [45] dataset. Most of the existing stereo image-based depth estimation techniques provide predefined values. This estimation leads to a wrong prediction when the actual

depth does not match the predefined values. Garg et al. [86] proposed a model that can estimate arbitrary depth values instead the predefined discrete ones. The proposed continuous disparity network (CDN) also outputs a set of discrete values with probabilities and offsets, which turns the discrete distribution into continuous distribution for accurate disparity estimation. The model was trained with Wasserstein objective function on the KITTI [45] dataset. CDN-SDN was applied to Pseudo-LiDAR [6] and DSGN [79] models. The SDN backbone estimates depth in the Pseudo-LiDAR network, whereas the DSGN backbone, PSMNET, is replaced with the CDN backbone.

The triangulation learning network (TLNet) [76] uses 3-D anchors to construct object-level geometric correlation between stereo images. Then, the neural network learns the correspondence between stereo images to triangulate the target object near the anchor. The channel reweighting method is also proposed to enhance informative features and weaken the noisy signals by measuring left-right coherence, which overcomes the high computational burden of generating disparity maps in Mono3D [44] network. Stereo CenterNet [77] uses semantic and geometric information in stereo images to implement 3-D object detection. They use the anchor-free 2-D box association method by detecting only objects in the left images and computing the left-right associations by predicting the distance between them.

Gao et al. [87] put forth an efficient stereo geometry network (ESGN) for 3-D object detection. The ResNet-34 [88] backbone is used to extract multiscale feature maps. Using a stereo correlation and reprojection module, the proposed 3-D efficient geometry-aware feature generation (EGFG) module constructs multiscale stereo volumes in camera frustum space. Then, multiple 3-D geometry-aware features are generated using a deep multiscale information fusion (multiscale BEV projection and fusion) module. A deep geometry-aware feature distillation scheme was proposed to help the stereo feature learning with LiDAR-based detectors. The experimental result on the KITTI dataset shows the ESGN model outperforms the YOLOStsereo3D [89] model. YOLOStereo3D [89] is faster than the ESGN model, but ESGN avoids object distortion in camera space by generating 3-D geometry-aware features. Guo et al. [90] proposed a stereo-based 3-D object detection model, LiDAR geometry aware stereo (LIGA-Stereo) detector. LiDAR-based models feature used to guide the learning of stereo-based models. A direct 2-D semantic supervision with an attached auxiliary 2-D detection head improved the learning efficiency. The experimental result on the KITTI [45] dataset shows the model outperforms the previous stereo-based models, such as DSGN [79].

Liu et al. [89] proposed YOLOStereo3D 3-D object detection model using stereo camera images. The authors described each anchor by 12 regressed parameters as $[x_{2d}, y_{2d}, w_{2d}, h_{2d}]$ for the 2-D BBoxs and $[c_x, c_y, z]$ for the 3-D centers of objects on the left image; $[w_{3d}, h_{3d}, l_{3d}]$ corresponds to the width, height, and length of the 3-D BBoxs, respectively. They concurrently applied photometric distortion augmentation [58] on binocular images and random flipping [83] during training. After extracting multiscale features from binocular images, the features passed through a multiscale stereo matching and fusion module. The Pseudo-LiDAR feature volume network (PLUMENet) [91] is a stereo image-based 3-D object detection model. Multiscale features are extracted from stereo images using a 2-D convolutional network. Then, the Pseudo-LiDAR feature volume is constructed in 3-D space. The 3-D occupancy grids and object BBoxs are predicted by multitasking headers (occupancy and detection headers) after a hybrid 3-D BEV network does 3-D reasoning. The experimental result on the KITTI [45] dataset shows the model outperforms the previous models, such as ZoomNet [5].

Zhang et al. [92] extended CenterNet [77] as a flexible framework for monocular 3-D object detection that explicitly decouples the truncated objects. The authors formulated the object depth estimation as an uncertainty-guided ensemble of multiple approaches and combined adaptively different key points to estimate the depth. The experimental results on the KITTI dataset [45] show the model outperforms models, such as RTM3D [70] and MoVi3D [33]. Chen et al. [93] proposed a pseudostereo 3-D detection method for 3-D object detection. The virtual view is generated from every single image to use as a stereo image with the input image. Three virtual view generation methods are proposed: image-level generation, feature-level generation, and feature clone for detecting 3-D objects from a single image. A disparitywise dynamic convolution is proposed to filter the features adaptively from a single image for generating virtual image features. The model is trained and tested on the KITTI [45] dataset.

Stereo image-based methods use 2-D left and right boxes to predict the BBoxs of objects in the 3-D space. Photometric alignment is usually used to optimize the 3-D BBox position further. The object-level geometric correlation between left and right images can be constructed using different techniques, such as 3-D anchors. The energy minimization function is also vital for generating 3DOPs. Some of the stereo-image-based methods use stereo matching and stereo instance segmentation to match detection between left and right images on ROIs and estimate instance-level disparity only for regions that contain objects of interest. The following methods use either stereo matching or stereo instance segmentation to match detection or estimate the disparity of ROI.

ZoomNet [5] applied adaptive zooming to resize BBoxs and adjust intrinsic camera parameters simultaneously to realize instance-level disparity estimation and construct point-cloud and Pseudo-LiDAR from each object instance rather than the full image. The Pseudo-LiDAR-based object detection has poor performance on distant objects because a distant object has low resolution because of the small number of points, the difficulty in distinguishing the relative positions between stereo images, and occlusions. This adaptive zooming helps analyze distant objects at larger resolutions, estimate better disparity, and have more uniform density point clouds. They also present pixelwise part locations to help solve the occlusion detection problem. Similarly, Pon et al. [95] proposed an object-centric stereo (OC Stereo) matching network, which solves the problems related to deep stereo matching methods.

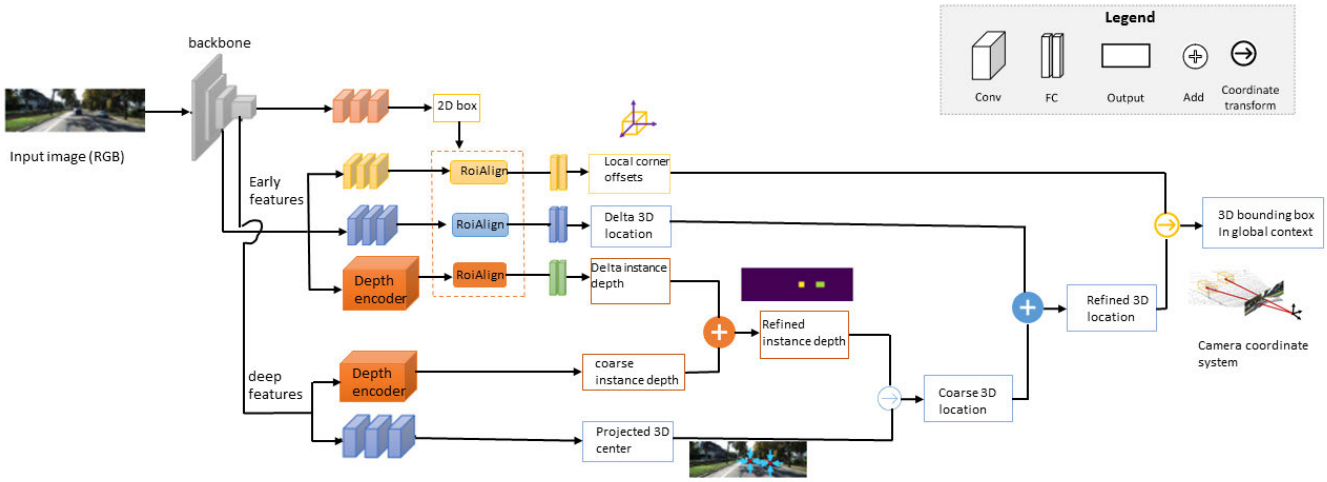


Fig. 11. MonoGRNet [94] 3-D object detection network best viewed in color. The model consists of four submodules: the brown subnetwork for 2-D detection, the orange subnetwork for instance depth estimation, the blue subnetwork for 3-D location estimation, and the yellow subnetwork for local corner regression.

They developed an object-centric depth representation to help solve streaking artifacts, the ambiguity between the object or the background pixels, and the pixel imbalance problem between near and far objects. The authors presented a fast 2-D box association algorithm to accurately match detection between left and right images by stereo matching on ROIs and considering only pixels belonging to objects. Disp r-CNN [96] is an iDispNet that estimates disparity only for regions that contain objects of interest rather than the entire image and learns a category-specific shape prior. This operation helps capture the smooth shape and sharp edges of object boundaries for more accurate 3-D object detection.

The lack of depth in image-based methods can be partially solved using stereo images. The 3DOPs are generated from stereo images using different techniques. Some methods, such as TLNET [76], use cost and channel reweighting to enhance features and weaken noises. Other methods formulated the object proposal as an energy minimization problem. Works such as DeepStereoOP [4] proposed a reranking algorithm to reduce redundant proposals and use only a few proposals. In addition, contextual information can be used together with stereo images for proposal generation.

C. Geometric Constraints Method

These works create 3-D proposals by adding additional geometric constraints, including object shape, ground planes, and key points [33], [38], [44], [58], [66], [70], [94], [97], [98], [99], [100], [101], [102], [103], [104]. Mousavian et al. [38] proposed Deep3DBox, a 3-D object detection method by incorporating geometry constraints. A hybrid discrete-continuous loss is used to estimate the 3-D object orientation and then apply regression on the 2-D BBox combined with the estimated geometric constraints to produce the object 3-D BBox. M3D-RPN [58] is a single end-to-end region proposal network for 3-D object detection using the correlation between 2-D scale and 3-D depth. The proposed depth-aware convolutional layer improves 3-D parameter estimation, enhancing 3-D scene understanding. Likewise, Mono3d++ [97] uses a joint method of predicting the vehicles' shape and pose using a 3-D BBox and morphable wireframe model from a single RGB image. The unsupervised monocular depth, a ground plane constraint, and vehicle shape priors optimize loss functions. The overall energy function integrates the loss and the vehicles' shape, and poses to improve vehicles' detection further. Integrating the loss function with the shape of vehicles may limit the model's performance because of the shape difference between vehicles.

Some methods use instance-level depth estimation using geometric reasoning. Others use a combination of key points and geometric information for depth estimation. For example, MonoGRNet [94] is a unified network for 3-D object detection from monocular RGB images using geometric reasoning and instance-level depth estimation. The model consists of the 2-D detection, instance depth estimation, 3-D location, and location corner estimation subnetworks, as shown in Fig. 11. The model was trained and tested on the KITTI [45] dataset. Barabanau et al. [100] also developed a combination of a key-point-based and geometric reasoning approach for 3-D object detection from monocular images. Similarly, Liu et al. [35] presented AutoShape, a one-stage real-time shape-aware monocular 3-D object detection model. The model employs geometry constraints for 3-D key points and their 2-D projections on images to enhance the detection performance. The proposed automatic annotation pipeline can autogenerate the shape-aware 2-D/3-D key points' correspondences for each object. The model was evaluated with the KITTI [45] car dataset. Likewise, Cai et al. [101] modeled the 3-D object detection task as a combination of a structured polygon prediction task and a depth estimation task. The depth estimation network uses an object's height to estimate the depth and then combines it with the structured polygon to obtain the 3-D boxes. Finally, fine-grained 3-D box refinement is proposed in BEV to improve the accuracy of the 3-D BBox.

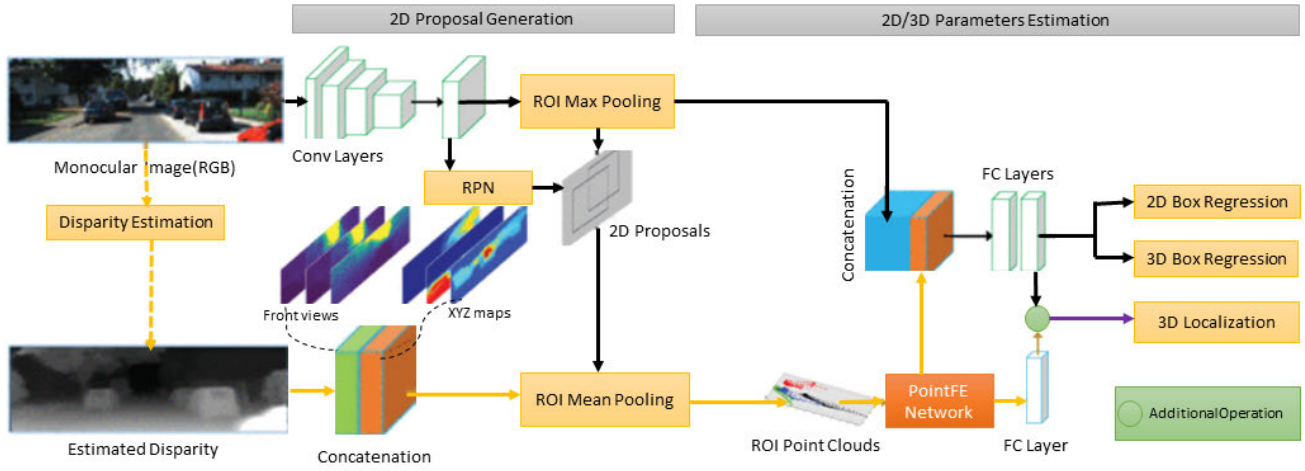


Fig. 12. MonoFENet architecture [105]. The disparity map is generated using the monocular disparity estimator. Then, the estimated disparity map is concatenated with the associated front view maps, such as distance, height, and depth. The ROI point clouds are generated using the ROI pooling layer before feeding to the proposed PointFE network. The point cloud features are fused with the RGB image features for 2-D and 3-D detections. Finally, the output of the pointFE network and 3-D detection head is fused to improve performance.

Ku et al. [102] estimated the region proposal network through geometric constraints and applied regression further for 3-D object detection. SMOKE [103] combined a single key point estimate with regressed 3-D variables to predict a 3-D BBox of individually detected objects rather than generating 2-D region proposals. Roddick et al. [104] proposed a 3-D object detection module by mapping image-based features into an orthographic 3-D space. An orthographic feature transforms the RGB image into an orthographic bird's-eye-view feature map. RTM3D [70] predicted the nine-perspective key points of a 3-D BBox and modeled the geometric relationship of 3-D and 2-D points to detect 3-D objects from monocular images. Similarly, MoVi-3D [33] is a one-stage deep architecture that leverages geometrical information to generate virtual views using prior geometrical knowledge to control the scale variability of the object because of depth.

GS3D [98] is an efficient model for getting a coarse cuboid for each predicted 2-D box to determine the 3-D BBox by refinement. This method improves the 3-D object detection and performs better than regression-based BBox prediction. ROI-10D [99] is an end-to-end network for 3-D object detection by lifting 2-D into 3-D to predict six degrees of freedom pose information (rotation and translation). The loss function measures the metrics misalignment of boxes and minimizes the error by comparing it with the ground-truth 3-D boxes.

Ding et al. [106] proposed a Depth-guided Dynamic Depthwise Dilated local convolution (D4LCN) network where local filters learn specific geometry from each RGB image using a depth map that is applied locally to each pixel and channel of each image. Some models, such as [34], avoid processing the image multiple times, which reduces the computational bottleneck of deep neural networks by generating perobject canonical 3-D BBox parameters using nonmaximal suppression (NMS) and nonlinear list square optimizer. Srivastava et al. [107] developed a 2-D to 3-D lifting method for AV's 3-D object detection. They generate BEV images from a single RGB image using generative adversarial networks (GANs) for image-to-image translation [108] and then do 3-D object detection using the generated BEV images.

de La Garanderie et al. [109] proposed a 3-D object detection model for AVs by using 360 panoramic imagery. This method is important to avoid blind spots in driving. The model was tested using the CARLA [110] urban driving simulator and the KITTI [45] dataset. Liu et al. [111] developed a deep-fitting scoring network for monocular 3-D object detection. The network generates 3-D proposals using the object's anchor-based dimension and orientation regression. Then, they use a fitting quality network (FQNet) to understand the spatial relationship between 3-D proposals and objects only using 2-D images. Chen et al. [48] proposed a pairwise spatial relationship-based 3-D object detection method. The object location is computed using uncertainty-aware predictions and 3-D distances for the adjacent pair. Finally, nonlinear least squares jointly optimize the system. By the same token, Bao et al. proposed MonoFENet [105] network for 3-D object detection by estimating the disparity from a monocular image. Fig. 12 shows the disparity image generated using the monocular-based disparity estimator. Then, the estimated disparity is transformed into a 3-D dense point cloud to feed into a point feature enhancement (PointFE) network and fuse with the image features for the final 3-D BBox regression.

Bao et al. [113] proposed a two-stage object-aware 3-D object detection model that uses both the regionwise appearance attention and the geometric projection distribution to vote the 3-D centroid proposals. The 2-D region proposals are generated using RPN from Faster R-CNN [18], and then, 3-D centroid proposals are estimated from generated ROIs' grid coordinates. Based on the proposed object-aware voting module, which comprises regionwise appearance attention and geometric projection distribution, the 3-D centroid proposals are voted for 3-D localization. Finally, 3-D BBoxs of objects are detected based on the proposed ROIs without learning the dense depth. Zhou et al. [114] put forth IAFA, an instance-aware feature aggregation model for 3-D object detection from

TABLE I

IMAGES 3-D OBJECT DETECTION METHODS COMPARISON BASED ON 3-D BBOX ENCODING TECHNIQUES, DEPTH ESTIMATION METHOD, AND YEAR OF PUBLICATION. ALL METHODS USE THE KITTI DATASET EXCEPT CDDN [72] AND SGM3D [73]. CDDN [72] USES KITTI AND WAYMO DATASETS, WHEREAS SGM3D [73] USES KITTI AND LYFT DATASETS. THE KITTI-BASED EVALUATION METRIC IS USED, EXCEPT FOR CDNN, WHICH USES BOTH KITTI AND WAYMO. SEE SECTIONS III-B AND III-C FOR 3-D BBOX ENCODING AND EVALUATION TECHNIQUES, RESPECTIVELY

Method	BBox Encoding technique	Depth Estimation Method	Year of Publication
Mono3D [43]	Energy minimization	Stereo Images	2015
3DOP [44]	Energy minimization	Stereo Images	2016
DeepStereoOP [4]	Energy minimization	Stereo Images	2017
Deep3DBox [38]	Axis-aligned	Geometric constraints	2017
Xu and Chen [68]		Pseudo-LiDAR	2018
Wang *et al.* [50]	Four corners & two heights and Seven points	Pseudo-LiDAR	2019
Ma *et al.* [59]	Seven points	Pseudo-LiDAR	2019
Weng and Kitani [24]	Seven points	Pseudo-LiDAR	2019
Pseudo-LiDAR++ [6]	Four corners & two heights and Seven points	Pseudo-LiDAR	2019
RefinedMPL [61]	Seven points	Pseudo-LiDAR	2019
TLNet [76]	Seven points	Stereo Images	2019
Stereo R-CNN [83]	Seven points	Stereo Images	2019
M3D-RPN [58]	Seven points	Geometric constraints	2019
Mono3d++ [97]	Energy minimization	Geometric constraints	2019
GS3D [98]	Seven points	Geometric constraints	2019
MonoGRNet [94]	Eight corners	Geometric constraints	2019
Barabanau *et al.* [100]	Eight corners	Geometric constraints	2019
MonoFENet [105]	Axis-aligned	Geometric constraints	2019
MonoPSR [102]	Seven points	Geometric constraints	2019
SMOKE [103]	Eight corners	Geometric constraints	2020
MonoPair [48]	Eight corners	Geometric constraints	2020
RTM3D [70]	Eight corners	Geometric constraints	2020
MoVi-3D [33]	Seven points	Geometric constraints	2020
D4LCN [106]	Seven points	Geometric constraints	2020
ZoomNet [5]	Seven points	Stereo Images	2020
Ida-3d [78]	Seven points	Stereo Images	2020
RAR-Net [112]	Seven points	–	2020
OC Stereo [95]	Four corners & two heights	Stereo Images	2020
Disp r-CNN [96]	Seven points	Stereo Images	2020
DSGN [79]	Seven points	Stereo Images	2020
Bao *et al.* [113]	Axis-aligned	Geometric constraints	2020
IAFA [114]	Seven points	Geometric constraints	2020
Qian *et al.* [60]	Seven points	Pseudo-LiDAR	2020
CDN-DSGN [86]	Seven points	Stereo Images	2020
CG-Stereo [85]	Seven points	Stereo Images	2020
CDDN [72]	Seven points	Pseudo-LiDAR	2021
Disp R-CNN [75]	Seven points	Pseudo-LiDAR	2021
Stereo CenterNet [77]	Seven points	Stereo Images	2021
YOLOStereo3D [89]	Seven points	Stereo Images	2021
Cai *et al.* [101]	Seven points	Geometric constraints	2021
GUP Net [115]	Seven points	Geometric constraints	2021
DDMP [116]	Seven points	Geometric constraints	2021
AutoShape [35]	Axis-aligned	Geometric constraints	2021
Zhang *et al.* [92]	Seven points	Stereo Images	2021
OCM3D [69]	Seven points	Pseudo-LiDAR	2021
PLUMENet-Large [91]	Seven Points	Pseudo-LiDAR	2021
LIGA-Stereo [90]	Seven points	Stereo Images	2021
SGM3D [73]	Seven points	Pseudo-LiDAR	2022
ESGN [87]	Seven points	Stereo Images	2022
Chen *et al.* [93]	Seven points	Pseudo-LiDAR	2022
DSGN++ [84]	Seven points	Stereo Images	2022

a single image. The model collects pixels that belong to the same object for contributing to the center classification and generates an attention map to aggregate useful information for each object. The authors used the coarse instance annotations from other networks as a supervision signal to generate the features aggregation attention maps. The model was trained on the KITTI [45] dataset.

Lu et al. [115] put forth a geometry uncertainty projection network (GUP Net) for monocular 3-D object detection. The input images are processed by the 2-D detection backbone, built on CenterNet [77], to get 2-D BBoxs (ROIs) and 3-D BBox information, i.e., angle, dimensions, and 3-D projected center for each box. Then, GPU Net predicts the depth information and its corresponding uncertainty by combining mathematical priors and uncertainty modeling. An efficient hierarchical task learning (HTL) strategy is proposed to reduce the instability caused by task dependency in geometry-based methods (error amplification). The error amplification causes amplification of the estimated depth. The HTL strategy controls the overall training process by making each task idle

TABLE II
BEV AND 3-D PERFORMANCE COMPARISON OF SELECTED IMAGE-BASED 3-D OBJECT DETECTION METHODS ON THE KITTI [45] VALIDATION BENCHMARK FOR CAR CLASS (*t* INDICATES EXPERIMENTS ON THE TEST BENCHMARK). R40 MEANS THAT THE MAP IS CALCULATED FOR 40 RECALL POINTS INSTEAD OF 11

Methods	$AP_{BEV}(IOU = 0.7)$			$AP_{3D}(IOU = 0.7)$		
	Easy	Moderate	Hard	Easy	Moderate	Hard
3DOP [3]	12.63	9.49	7.59	6.55	5.07	4.10
MonoPair [48] (r40)	24.12	18.17	15.76	16.28	12.30	10.42
TLNET [76]	29.22	21.88	18.83	18.15	14.26	13.72
Ma *et al.* [59]	43.75	28.39	23.87	32.23	21.09	17.26
Stereo RCNN [83]	68.50	48.30	41.47	54.11	36.69	31.07
Ida-3d [78]	70.68	50.21	42.93	54.97	37.45	32.23
Wang *et al.* [50]	74.90	56.80	49.00	61.90	45.30	39.00
Disp R-CNN [96]	76.51	58.63	50.26	63.57	47.15	39.73
OC-Stereo [95]	77.66	65.95	51.20	64.07	48.34	40.39
ZoomNet [5]	78.68	66.19	57.60	62.96	50.47	43.63
DSGN [79]	83.24	63.91	57.93	72.31	54.27	47.71
CDN-DSGN [86] (t)	83.30	66.20	57.70	74.50	54.20	46.40
PLUMENET-Large [91]	84.70	71.10	65.10	-	-	-
CG-Stereo [85]	87.31	68.69	65.80	76.17	57.82	54.63
LIGA-Stereo [90] (t)	88.15	76.78	67.40	81.39	64.66	57.22
DSGN++ [84] (t)	88.55	78.94	69.74	83.21	67.37	59.91

until its pretasks are well-trained. The experimental result on the KITTI dataset [45] outperforms methods, such as MoVi-3D [33] and RAR-net [112].

Wang et al. [116] proposed a graph-based depth-conditioned dynamic message propagation (DDMP) model for monocular 3-D object detection. The model comprises two branches: the regression branch and the depth extraction branch. The regression branch receives the RGB images for feature extraction, and the depth extraction branch estimates the corresponding depth maps and extracts depth-aware features. The center-aware depth encoding (CDE) method is proposed to reduce the inaccurate depth prior issues. The context-aware and depth-aware features are integrated with a graph message propagation pattern via the DDMP module. Finally, 3-D object boxes were achieved using a 3-D detection head. The experimental result on the KITTI dataset [45] shows that the model outperforms the previous models, such as D4LCN [106].

Some works followed different approaches than we mentioned above to solve the 3-D objection problem from the input of 2-D images. Liu et al. [112] put forth RAR-Net, a reinforced axial refinement network monocular 3-D object detection model. The proposed model starts with an initial prediction and refines it gradually toward the ground truth, and only one 3-D parameter is changed in each step. The ϵ-greedy policy maximizes the reward by selecting the action with the highest estimated reward after each action is taken to refine the 3-D box of the monocular 3-D detection network. At each step, information from the image and 3-D space is fused; then, project the current detection into the image space to preserve information. This reinforcement learning-based learning can be used as a postprocessing stage and integrated into an existing monocular 3-D detection model to improve performance with some extra computational cost. The model was trained with the KITTI dataset [45] and showed promising performance. Mehtab et al. [117] proposed a 3-D vehicle detection model using LiDAR and camera sensors. The AV's size and orientation of 3-D BBoxs are estimated from the RGB images, whereas the LiDAR point cloud is used for distance estimation. As an image feature extractor, the authors used MobileNetV2 [118]. The model was trained and tested on the KITTI [45] and Waymo [51] datasets. Simonelli et al. [46] put forth self-supervised loss disentangling transformation for monocular 3-D object detection. The loss separates the groups of parameter contributions into separate terms from the original loss. The authors also applied the loss function IOU for 2-D detection and 3-D BBox predictions, and detection confidence. The model was trained on the KITTI [45] dataset.

The three depth estimation techniques perform different operations to estimate depth from 2-D images. The Pseudo-LiDAR methods transform the image into a LiDAR representation and use LIDAR-based models to leverage 3-D information of the LiDAR representation. On the other hand, the stereo-based models do not transform the image into another domain; instead, depth is generated from the left and right stereo images. The geometric constraints method uses additional geometric constraints, including object shape, ground planes, and key points to estimate the depth information from 2-D images. The 3-D BBox encoding technique, 3-D object detection evaluation method, datasets used for the experiment, and year of publication of each method are presented in Table I. Table II shows the BEV and 3-D performance comparison of the image-based 3-D object detection methods on KITTI [45] validation and test data benchmarks.

V. CHALLENGES AND FUTURE DIRECTIONS

Camera images, especially monocular images, are rich in texture and color information, which are essential for color-related tasks, such as object classification and lane detection. However, they do not provide highly accurate depth information for a complete understanding of the surrounding environment. Autonomous driving needs to be robust to drive in different weather conditions, but cameras are affected by bad weather. In addition, DL models evaluated on a different domain than trained perform poorly. We presented challenges and future research directions in image-based 3-D object detection for AVs.

1) *Semisupervised Learning:* One of the challenges of supervised learning is annotating and labeling data, which requires time and money. Data annotation and labeling problems can be solved using unsupervised learning. However, unsupervised models' detection and classification accuracy are lower than the supervised models. The potential solution to these problems is applying a semisupervised model using few labeled data and many unlabeled data to leverage the abundance of freely available images for different applications. Some teacher–student models, such as Zhang et al. [119], belong to a semisupervised 3-D object detection network for autonomous driving. The teacher model generates pseudolabels in the teacher–student model, and the student model trains the pseudolabels and the labeled dataset. Then, the teacher model may receive an update from the student model for better pseudolabel prediction. This model is mainly used in 2-D object detection, but the 3-D equivalents are limited.
2) *Multitask Learning:* The feature extractor part of DL networks can be common to multiple applications. Therefore, building a model with common feature extractor/lower architecture of the model with multiple decision layers to perform multiple tasks can save time, memory, and computational power. For example, Liang et al. [120] perform object detection and segmentation multitask learning. We expect many multitask learning works for AVs.
3) *Domain Adaptive Models:* DL models should perform the same/equivalent when tested with a different domain than they were trained. However, most DL models perform poorly when the training domain changes. Domain adaptive models are essential for autonomous driving to avoid country-specific changes, such as traffic sign variability and corner issues. Therefore, we need domain adaptive models to learn the driving environment changes and respond quickly to the changes.
4) *Lightweight Models:* DL models in AVs should fulfill the following three criteria [1].
    a) **Accurate** to precise information about the surrounding environments.
    b) **Robust** to work in different weather.
    c) **Real time** to perform high-speed driving. To achieve the above criteria, DL models should be robust enough to work under different weather and lightweight to be deployed in low-power and low-memory embedded hardware devices. Most existing 3-D object detection models are not lightweight as their 2-D equivalents. There are relatively lightweight 2-D object detection models, such as YOLO [121] and SSD [29], than 3-D object detection models.
5) *Multisensor Fusion:* Cameras are suitable for color-related detection and rich in texture too. Although different methods have been developed to solve the lack of 3-D information, 3-D object detection using cameras is challenging. In addition, cameras are not robust to adverse weather, which makes robust driving in different environmental weather challenging. Other sensors can provide better 3-D information, such as LiDAR, and more robust to adverse weather, such as radar. Therefore, fusing the camera images with LiDAR and/or radar can improve 3-D object detection by using the best out of different sensors (refer [14] for a detailed analysis of multisensor fusion methods and different sensor fusion techniques in 3-D object detection).
6) *Adding Temporal Cues to Spatial Information:* In existing 3-D object detection models, single-frame (spatial) data, which comprises finite information, are used. Including temporal information in the spatial information may improve the detection performance. BEVDet4D [122] has shown promising results in using temporal information in addition to spatial information.
7) *Balanced Dataset:* Most of the existing datasets have class imbalance problems where some of the classes have many samples, whereas others have few. The majority of classes influence models during decisions due to the high representation in the data. By taking time and collecting more data for those less represented classes or coming up with other solutions, such as proposing loss functions [123], [124], we can minimize or avoid the effect of class imbalance issues. In addition, generating synthetic data from simulators and training with real data may help to solve the class imbalance problem.

VI. Conclusion

This survey presented DL-based monocular and stereo camera image-based 3-D object detection for autonomous driving. The 3-D BBox encoding methods and the corresponding evaluation metrics were summarized. The general object detection categories as one- and two-stage and depth estimation methods of 3-D object detection are also reviewed. The depth estimation methods are grouped based on techniques, such as Pseudo-LiDAR, stereo image, and geometric constraint methods. Although 3-D object detection using camera images has shown significant performance improvement due to the rapid growth of DL, there are still issues to be solved for reliable and robust driving, such as driving in bad weather or at night. The camera sensor is rich in color and texture and inexpensive, but it cannot measure the distance from long range, cannot withstand bad weather, and does not give direct 3-D information [14], [42], [125]. 3-D sensors, such as LiDAR and radar, provide 3-D information about the driving environment and objects. LiDAR is more robust than a camera for inclement weather and a good choice for long-distance measurement and velocity estimation. However, it is not rich in color and texture. Similarly, radar is a robust sensor for inclement weather and the best choice for distance measurement and velocity estimation, but it has low resolution, making radar-based detection difficult. In addition, there is a possibility of sensor failure during autonomous driving. Thus, using multiple sensors for autonomous driving is essential to use redundant data from different sensors for reliable and robust driving to work under bad weather or sensor failure conditions. Lightweight and accurate 3-D

object detection models are necessary to improve the speed and accuracy of real-time processing. Finally, challenges and possible research directions were presented.

REFERENCES

[1] D. Feng et al., "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[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 Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019.

[3] X. Z. Chen, K. Kundu, Y. Zhu, S. Fidle, R. Urtasun, and H. Ma, "3D object proposals using stereo imagery for accurate object class detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1259–1272, May 2018.

[4] C. C. Pham and J. W. Jeon, "Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks," *Signal Process., Image Commun.*, vol. 53, pp. 110–122, Apr. 2017.

[5] Z. Xu et al., "ZoomNet: Part-aware adaptive zooming neural network for 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12557–12564.

[6] Y. You et al., "Pseudo-LiDAR++: Accurate depth for 3D object detection in autonomous driving," 2019, *arXiv:1906.06310*.

[7] S.-H. Kim and Y. Hwang, "A survey on deep learning based methods and datasets for monocular 3D object detection," *Electronics*, vol. 10, no. 4, p. 517, Feb. 2021.

[8] L. Jiao et al., "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019.

[9] M. M. Rahman, Y. Tan, J. Xue, and K. Lu, "Recent advances in 3D object detection in the era of deep neural networks: A survey," *IEEE Trans. Image Process.*, vol. 29, pp. 2947–2962, 2019.

[10] Y. Li et al., "Deep learning for LiDAR point clouds in autonomous driving: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3412–3432, Aug. 2021.

[11] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.

[12] D. Fernandes et al., "Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy," *Inf. Fusion*, vol. 68, pp. 161–191, Apr. 2021.

[13] R. Qian, X. Lai, and X. Li, "3D object detection for autonomous driving: A survey," 2021, *arXiv:2106.10823*.

[14] S. Alaba, A. Gurbuz, and J. Ball, "A comprehensive survey of deep learning multisensor fusion-based 3D object detection for autonomous driving: Methods, challenges, open issues, and future directions," *TechRxiv*, 2022, doi: 10.36227/techrxiv.20443107.v2.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[17] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 1–9.

[19] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," 2016, *arXiv:1605.06409*.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2961–2969.

[21] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[22] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head R-CNN: In defense of two-stage object detector," 2017, *arXiv:1711.07264*.

[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[24] X. Weng and K. Kitani, "Monocular 3D object detection with pseudo-LiDAR point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 857–866.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[26] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[27] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[28] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[29] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37.

[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[31] Q. Zhao et al., "M2Det: A single-shot object detector based on multi-level feature pyramid network," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 9259–9266.

[32] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4203–4212.

[33] A. Simonelli, S. R. Bulò, L. Porzi, E. Ricci, and P. Kontschieder, "Towards generalization across depth for monocular 3D object detection," 2019, *arXiv:1912.08035*.

[34] E. Jörgensen, C. Zach, and F. Kahl, "Monocular 3D object detection and box fitting trained end-to-end using intersection-over-union loss," 2019, *arXiv:1906.08070*.

[35] Z. Liu, D. Zhou, F. Lu, J. Fang, and L. Zhang, "AutoShape: Real-time shape-aware monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15641–15650.

[36] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.

[37] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–8.

[38] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7074–7082.

[39] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[40] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[41] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.

[42] S. Y. Alaba and J. E. Ball, "WCNN3D: Wavelet convolutional neural network-based 3D object detection for autonomous driving," *Sensors*, vol. 22, no. 18, p. 7010, Sep. 2022.

[43] X. Chen et al., "3D object proposals for accurate object class detection," in *Proc. Adv. Neural Inf. Process. Syst.* Princeton, NJ, USA: Citeseer, 2015, pp. 424–432.

[44] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2147–2156.

[45] 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.

[46] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1991–1999.

[47] 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, 2010.

[48] Y. Chen, L. Tai, K. Sun, and M. Li, "MonoPair: Monocular 3D object detection using pairwise spatial relationships," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12093–12102.

[49] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.

[50] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8445–8453.

[51] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2446–2454.

[52] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, Sep. 2014, pp. 391–405.

[53] P. Krahenbuhl and V. Koltun, "Learning to propose objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1574–1582.

[54] T. Lee, S. Fidler, and S. Dickinson, "Learning to combine mid-level cues for object proposal generation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1680–1688.

[55] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," 2014, *arXiv:1406.2283*.

[56] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 924–933.

[57] Z. Lingtao, F. Jiaojiao, and L. Guizhong, "Object viewpoint classification based 3D bounding box estimation for autonomous vehicles," 2019, *arXiv:1909.01025*.

[58] G. Brazil and X. Liu, "M3D-RPN: Monocular 3D region proposal network for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9287–9296.

[59] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, "Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6851–6860.

[60] R. Qian et al., "End-to-end pseudo-LiDAR for image-based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5881–5890.

[61] J. M. U. Vianney, S. Aich, and B. Liu, "RefinedMPL: Refined monocular PseudoLiDAR for 3D object detection in autonomous driving," 2019, *arXiv:1911.09712*.

[62] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5410–5418.

[63] N. Mayer et al., "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4040–4048.

[64] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *Proc. Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, Sep. 2014, pp. 756–771.

[65] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

[66] H.-N. Hu et al., "Joint monocular 3D vehicle detection and tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5390–5399.

[67] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927.

[68] B. Xu and Z. Chen, "Multi-level fusion based 3D object detection from monocular images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2345–2353.

[69] L. Peng, F. Liu, S. Yan, X. He, and D. Cai, "OCM3D: Object-centric monocular 3D object detection," 2021, *arXiv:2104.06041*.

[70] P. Li, H. Zhao, P. Liu, and F. Cao, "RTM3D: Real-time monocular 3D detection from object keypoints for autonomous driving," 2020, *arXiv:2001.03343*.

[71] Z. Chong et al., "MonoDistill: Learning spatial features for monocular 3D object detection," 2022, *arXiv:2201.10830*.

[72] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8555–8564.

[73] Z. Zhou et al., "SGM3D: Stereo guided monocular 3D object detection," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10478–10485, Oct. 2022.

[74] R. Kesten et al., "Woven planet perception dataset 2020," 2019. [Online]. Available: https://www.woven-planet.global/en/data/perception-dataset

[75] L. Chen et al., "Shape prior guided instance disparity estimation for 3D object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5529–5540, Sep. 2021.

[76] Z. Qin, J. Wang, and Y. Lu, "Triangulation learning network: From monocular to stereo 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7615–7623.

[77] Y. Shi, Y. Guo, Z. Mi, and X. Li, "Stereo CenterNet based 3D object detection for autonomous driving," 2021, *arXiv:2103.11071*.

[78] W. Peng, H. Pan, H. Liu, and Y. Sun, "IDA-3D: Instance-depth-aware 3D object detection from stereo vision for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13015–13024.

[79] Y. Chen, S. Liu, X. Shen, and J. Jia, "DSGN: Deep stereo geometry network for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12536–12545.

[80] H. Königshof, N. O. Salscheider, and C. Stiller, "Realtime 3D object detection for automated driving using stereo vision and semantic information," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1405–1410.

[81] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the ResNet model for visual recognition," *Pattern Recognit.*, vol. 90, pp. 119–133, Jun. 2019.

[82] S. Li and K.-T. Cheng, "Joint stereo 3D object detection and implicit surface reconstruction," 2021, *arXiv:2111.12924*.

[83] P. Li, X. Chen, and S. Shen, "Stereo R-CNN based 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7644–7652.

[84] Y. Chen, S. Huang, S. Liu, B. Yu, and J. Jia, "DSGN++: Exploiting visual-spatial relation for stereo-based 3D detectors," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 8, 2022, doi: 10.1109/TPAMI.2022.3197236.

[85] C. Li, J. Ku, and S. L. Waslander, "Confidence guided stereo 3D object detection with split depth estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5776–5783.

[86] D. Garg, Y. Wang, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao, "Wasserstein distances for stereo disparity estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 22517–22529.

[87] A. Gao et al., "ESGN: Efficient stereo geometry network for fast 3D object detection," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Aug. 29, 2022, doi: 10.1109/TCSVT.2022.3202810.

[88] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[89] Y. Liu, L. Wang, and M. Liu, "YOLOStereo3D: A step back to 2D for efficient stereo 3D detection," 2021, *arXiv:2103.09422*.

[90] X. Guo, S. Shi, X. Wang, and H. Li, "LIGA-stereo: Learning LiDAR geometry aware representations for stereo-based 3D detector," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3153–3163.

[91] Y. Wang, B. Yang, R. Hu, M. Liang, and R. Urtasun, "PLUMENet: Efficient 3D object detection from stereo images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3383–3390.

[92] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3289–3298.

[93] Y.-N. Chen, H. Dai, and Y. Ding, "Pseudo-stereo for monocular 3D object detection in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 887–897.

[94] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A geometric reasoning network for monocular 3D object localization," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 8851–8858.

[95] A. D. Pon, J. Ku, C. Li, and S. L. Waslander, "Object-centric stereo matching for 3D object detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 8383–8389.

[96] J. Sun et al., "Disp R-CNN: Stereo 3D object detection via shape prior guided instance disparity estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10548–10557.
[97] T. He and S. Soatto, "Mono3D++: Monocular 3D vehicle detection with two-scale 3D hypotheses and task priors," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 8409–8416.
[98] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "GS3D: An efficient 3D object detection framework for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1019–1028.
[99] F. Manhardt, W. Kehl, and A. Gaidon, "ROI-10D: Monocular lifting of 2D detection to 6D pose and metric shape," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2069–2078.
[100] I. Barabanau, A. Artemov, E. Burnaev, and V. Murashkin, "Monocular 3D object detection via geometric reasoning on keypoints," 2019, *arXiv:1905.05618*.
[101] Y. Cai, B. Li, Z. Jiao, H. Li, X. Zeng, and X. Wang, "Monocular 3D object detection with decoupled structured polygon estimation and height-guided depth estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10478–10485.
[102] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3D object detection leveraging accurate proposals and shape reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11867–11876.
[103] Z. Liu, Z. Wu, and R. Tóth, "SMOKE: Single-stage monocular 3D object detection via keypoint estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 996–997.
[104] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3D object detection," 2018, *arXiv:1811.08188*.
[105] W. Bao, B. Xu, and Z. Chen, "MonoFENet: Monocular 3D object detection with feature enhancement networks," *IEEE Trans. Image Process.*, vol. 29, pp. 2753–2765, 2019.
[106] M. Ding et al., "Learning depth-guided convolutions for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1000–1001.
[107] S. Srivastava, F. Jurie, and G. Sharma, "Learning 2D to 3D lifting for object detection in 3D for autonomous vehicles," 2019, *arXiv:1904.08494*.
[108] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.
[109] G. P. de La Garanderie, A. A. Abarghouei, and T. P. Breckon, "Eliminating the blind spot: Adapting 3D object detection and monocular depth estimation to 360 panoramic imagery," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 789–807.
[110] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.
[111] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, "Deep fitting degree scoring network for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1057–1066.
[112] L. Liu, C. Wu, J. Lu, L. Xie, J. Zhou, and Q. Tian, "Reinforced axial refinement network for monocular 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*Glasgow, U.K.: Springer, Aug. 2020, pp. 540–556.
[113] W. Bao, Q. Yu, and Y. Kong, "Object-aware centroid voting for monocular 3D object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2197–2204.
[114] D. Zhou et al., "IAFA: Instance-aware feature aggregation for 3D object detection from a single image," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–18.
[115] Y. Lu et al., "Geometry uncertainty projection network for monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3111–3121.
[116] L. Wang et al., "Depth-conditioned dynamic message propagation for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 454–463.
[117] S. Mehtab, W. Q. Yan, and A. Narayanan, "3D vehicle detection using cheap LiDAR and camera sensors," in *Proc. 36th Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Dec. 2021, pp. 1–6.
[118] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[119] J. Zhang, H. Liu, and J. Lu, "A semi-supervised 3D object detection method for autonomous driving," *Displays*, vol. 71, Jan. 2022, Art. no. 102117.
[120] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7345–7353.
[121] W. Ali, S. Abdelkarim, M. Zidan, M. Zahran, and A. E. Sallab, "YOLO3D: End-to-end real-time 3d oriented object bounding box detection from LiDAR point cloud," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018, pp. 716–728.
[122] J. Huang and G. Huang, "BEVDet4D: Exploit temporal cues in multi-camera 3D object detection," 2022, *arXiv:2203.17054*.
[123] S. Y. Alaba et al., "Class-aware fish species recognition using deep learning for an imbalanced dataset," *Sensors*, vol. 22, no. 21, p. 8268, Oct. 2022.
[124] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9268–9277.
[125] S. Y. Alaba and J. E. Ball, "A survey on deep-learning-based LiDAR 3D object detection for autonomous driving," *Sensors*, vol. 22, no. 24, p. 9577, Dec. 2022.

Simegnew Yihunie Alaba (Member, IEEE) received the B.S. degree in electrical engineering from Arba Minch University, Arba Minch, Ethiopia, and the M.S. degree in computer engineering from Addis Ababa University, Addis Ababa, Ethiopia. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Mississippi State University, Starkville, MS, USA.

His research interests include image processing, computer vision, deep learning, and autonomous driving.

John E. Ball (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Mississippi State University, Starkville, MS, USA, in 1991 and 2007, respectively, and the M.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 1993.

He is currently an Associate Professor and the Endowed Chair of Electrical and Computer Engineering with Mississippi State University. His research interests include sensors, sensor processing, deep learning, and autonomous vehicles (AVs), especially in an unstructured environment.

Dr. Ball serves as an Associate Editor for IEEE SIGNAL PROCESSING LETTERS and the *Journal of Applied Remote Sensing*.