Monocular 3-D Object Detection Based on Depth-Guided Local Convolution for Smart Payment in D2D Systems

Jun Li, *Member, IEEE*, Wei Song, Yongbin Gao, Huixing Wang, Yier Yan, Bo Huang, Jun Zhang, and Wei Wang, *Member, IEEE*

***Abstract*—3-D object detection from mobile phones in Device-to-Device (D2D) system provides a new smart payment tool for the next generation of fintech, which is more flexible and efficient than the traditional barcode. In this article, we propose a monocular 3-D object detection method based on depth-guided local convolution. The method combines the information of RGB image mode and depth mode by using a convolution kernel through depth image and works on a single RGB image locally. According to the multiscale input information, the convolution kernel is adaptively adjusted to capture the target objects of different scales, so as to improve the performance of 3-D object detection. In addition, we use the soft-non-maximum suppression algorithm instead of traditional non-maximum suppression to select the best prediction box. In order to further improve the accuracy of 3-D object detection, the depth estimation network and 3-D object detection network are jointly trained in this method to make the two networks constrain each other and achieve the best performance.**

***Index Terms*—Depth-guided local convolution, joint training, monocular 3-D object detection, smart payment, soft-non-maximum suppression (soft-NMS).**

Manuscript received 8 May 2021; revised 27 August 2021; accepted 2 November 2021. Date of publication 16 November 2021; date of current version 24 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61802253 and Grant 61872102; in part by the Chenguang Talented Program of Shanghai under Grant 17CG59; in part by the International Collaborative Research Program of Guangdong Science and Technology Department under Grant 2020A0505100061; in part by the Guangzhou University–The Hong Kong University of Science and Technology (GZU-HKUST) Joint Research Program under Grant YH202110; and in part by the Guangzhou Municipal Science and Technology Project under Grant 202102010416. *(Jun Li and Wei Song contributed equally to this work.) (Corresponding author: Yongbin Gao.)*

Jun Li and Yier Yan are with the Research Center of Intelligent Communication Engineering, School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China (e-mail: lijun52018@gzhu.edu.cn; year0080@gzhu.edu.cn).

Wei Song is with the Department of Electronic Information and Communication Engineering, Applied Technology College of Soochow University, Suzhou 215325, China (e-mail: songw3015@suda.edu.cn).

Yongbin Gao, Huixing Wang, and Bo Huang are with the School of Electronic and Electronics Engineering, Shanghai University of Engineering Science, Shanghai 201620, China (e-mail: gaoyongbin@sues.edu.cn; 1172213872@qq.com; huangbosues@sues.edu.cn).

Jun Zhang is with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: eejzhang@scut.edu.cn).

Wei Wang is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: weiwa@cse.ust.hk).

Digital Object Identifier 10.1109/JIOT.2021.3128440

I. Introduction

3-D OBJECT detection from the monocular camera is an automation analysis method to build the next-generation and intelligent Internet of Things (IoT), it has been applied to many fields in real life, such as smart payment, autonomous driving, robotics, and augmented reality. Now, the growth of communication technology [1]–[4] can provide more efficient and secure guarantee for the IoT and intelligent payment. The difference between 3-D object detection and 2-D object detection lies in the additional depth information. Currently, the sensors used to obtain depth information mainly include a binocular camera, RGB-D camera, and LIDAR. However, due to their disadvantages, such as large amount of computation, environmental impact, and high cost, monocular cameras have been used as a more efficient alternative. Compared with the above sensors, a monocular camera has the advantages of low cost, small size, and flexible operation. Although the problem of monocular 3-D object detection has attracted a lot of attention, there are still a lot of problems that have not been well solved.

The typical monocular 3-D object detection methods can be divided into three categories: 1) image-based method; 2) LIDAR-based method; and 3) pseudo-LIDAR-based method. The image-based method mainly uses the idea of combining deep learning with geometric constraints and designs sensitive loss function to calculate the scale and position of objects in the real world, so as to realize 3-D object detection. However, the scale of objects with different distances often changes significantly in a single image. The traditional 2-D convolution is difficult to process the target objects with different scales at the same time, which will lead to the failure to obtain the local target objects and their scale information. It will reduce the accuracy of 3-D object detection. The method based on LIDAR mainly uses LIDAR sensors to collect data, further generates 3-D point cloud, and then processes the point cloud data. Finally, the detection framework based on point cloud data is used for 3-D object detection. However, due to the high cost and sparse output of LIDAR sensor, the 3-D object detection method based on LIDAR is still unable to be widely used. Considering the limitations of the image-based method and LIDAR-based method, people begin to study the method based on pseudo-LIDAR. Pseudo-LIDAR means to use a 2-D image to obtain a

depth map and then calculate the depth map and camera parameters to obtain 3-D point cloud data. It only contains the coordinate information of the points in the point cloud but does not contain the RGB semantic information. So it is referred to as "pseudo-LIDAR." The main idea is to estimate the depth of 2-D image and then convert the depth map into point cloud representation and use the algorithm framework based on point cloud data to detect 3-D objects. This method not only avoids the problem that the 2-D image cannot obtain the scale information of different objects but also can replace the real LIDAR point cloud data and has achieved remarkable results. However, it still has the following disadvantages.

1) The depth map estimated from a single image is often rough, it will lead to the generation of inaccurate point cloud data and then reduce the performance of 3-D object detection.
2) The method based on pseudo-LIDAR is to first obtain the depth values of pixels in 2-D images by using depth estimation and then calculate 3-D point cloud by combining with camera parameters. However, the method of combining camera parameters with depth values is essentially a geometric transformation, so the pseudo-LIDAR is only a kind of 3-D spatial geometric information obtained from 2-D images. This kind of geometric information often ignores the high-level semantic information in 2-D images. However, using semantic information, it is easy to detect more obvious objects, such as roadblocks, trees, and dust on the road. Therefore, a better representation from 2-D image to 3-D space should include both geometric information and semantic information in 3-D space.
3) The method based on pseudo-LIDAR usually processes the depth estimation task and 3-D object detection task by stages. However, this processing method often causes error accumulation and makes the detection accuracy not reach the best state.

To solve the above problems, this article proposes a 3-D object detection method based on depth-guided local convolution, which combines the information of two different modes (RGB image and depth) and uses a depth map to generate a local convolution kernel, an adaptive convolution through selective kernel network (SKNet) [5] is performed on a single RGB image sample to complete the final 3-D object detection. In order to fully combine the feature information of RGB image and depth map, this method first uses two convolution networks to extract the features of RGB image and depth map, then uses the depth-guided local convolution module to fuse the output features of the two networks, and finally uses the output features for 3-D object detection. In order to get the best prediction box, the soft-non-maximum suppression (soft-NMS) [6] algorithm is used instead of the traditional non-maximum suppression (NMS) [7] algorithm to select a better prediction box. In order to solve the problem of error accumulation caused by staged training, the joint training method is used to train the depth estimation network and the 3-D object detection network, so that the two networks constrain each other, and then the network performance reaches the best state.

Inspired by the SKNet, we propose a depth-guided local convolution operation, the steps are as follows. First, the RGB image features are multiscale branched to generate multiple paths with different sizes of the kernel, which correspond to neurons with different sizes of receptive fields. Then, the multiple paths are fused to learn the adaptive weight with global information representation, and the adaptive weight is used to adjust the depth map features after the move operation, and then the adjusted depth map features are fused with the features of each path. Finally, the feature map of each path is fused, and the fused feature map is used as the input of the detection head to complete the 3-D object detection task.

Our main contributions of this article are listed as follows.

1) A monocular 3-D object detection method based on depth-guided local convolution is proposed. This method combines the relevant information between RGB image mode and depth mode. It generates a local convolution kernel in the depth map and acts on a single image. This operation will get a better representation from 2-D image to 3-D space that includes both geometric information and semantic information in 3-D space.
2) The soft-NMS algorithm is used to replace the traditional NMS algorithm, and the Gaussian function is used to punish all boxes that overlap the maximum score detection box to avoid the loss of the object detection box and obtain the best prediction box.
3) A joint training mechanism is proposed to train the depth estimation network and 3-D object detection network, so that the two networks constrain each other, and then the network performance reaches the best state.
4) Experiments on the KITTI dataset show the effectiveness of the proposed method. Compared with the existing monocular 3-D object detection methods, the accuracy is improved by 3.6%.

II. Related Works

A. Monocular 3-D Object Detection Based on Image

The existing monocular 3-D object detection methods usually make assumptions on the scene geometry and combine with a 2-D detection network to establish a 3-D parameter regression network, which is used as the constraint of training 2-D to 3-D mapping to simplify the calculation process of 3-D parameters. Deep3-DBox [8] extends the 2-D object detection network, obtains the 3-D dimension and heading angle of the target by regression method, then restores the 3-D pose of the object, and solves the translation matrix from the center of the target to the center of the camera, so as to minimize the error between the reprojection center coordinates of the 3-D detection box and the center coordinates of the 2-D detection box. According to the orthogonal transformation between the image and 3-D spatial features, OFT-Net [9] back projects the feature map of the image into the bird's-eye view of 3-D space and then processes the bird's-eye view feature map by residual network unit. By fusing a large amount of prior information, Mono3D [10] and Mono3D++ [11] calculate the energy loss function of the detection box and extract the accurate 3-D object detection box. In order to

introduce more prior information, Deep MANTA [12], ROI-10D [13], and 3D-RCNN [14] extract 3-D object information through CAD template matching to obtain better object geometry information, so as to improve the performance of 3-D object detection. MonoDIS [15], [16] uses multiple arrays to represent the 3-D geometric pose of the object and adopts the decoupling method to separate the parameter errors. This method makes the loss function of the network converge faster in the training process and avoids the interference of error transfer among the parameters. MonoPSR [17] uses the mature 2-D object detector to generate 3-D proposals for each object in the scene, which greatly reduces the difficulty of the final 3-D bounding box detection. At the same time, the point cloud is predicted in the object-centered coordinate system, and the total loss including the new projection alignment loss is designed to learn the local scale and shape information.

However, because 2-D image features are difficult to represent the 3-D spatial structure, the above methods can not only recover the accurate 3-D information from a single image. Therefore, our goal is to use depth information to guide the learning of feature representation from 2-D to 3-D and make up the gap between 2-D and 3-D representation.

B. 3-D Object Detection Based on LIDAR

With the development of deep learning, 3-D feature learning [18]–[20] can learn deep point-based and voxel-based features. Benefiting from this, the method based on LIDAR has achieved impressive results in 3-D detection. For example, PIXOR [21] obtains a 2-D bird's-eye view feature map with height and reflectivity as channels through point cloud, and then uses RetinaNet [22] with the fine-tuning structure for object detection and positioning, which maintains spatial information and object geometry and can achieve real-time object detection. STD [23] uses point cloud as input to generate an accurate proposal for each point and transforms its internal point features from sparse representation to compact representation to generate proposal features. VoxelNet [20] divides the point cloud into equidistant 3-D voxels and transforms a group of points in each voxel into a unified feature representation, which ensures the original 3-D features of the point cloud. Finally, after compression and extraction by convolution network, it is connected to the RPN layer for detection. Inspired by voxelNet, Yan *et al.* proposed a sparse embedded convolutional detection network (SECOND) [24], which connects two voxel feature coding architectures in series through sparse convolutional neural network, making full use of the sparsity of point cloud and improving the feature extraction effect in voxelNet. PointPillars [25] uses pointNets [18] to learn the representation of point cloud organized by vertical columns and can be used with any standard 2-D convolution detection architecture. FP [26] uses mature 2-D object detectors to learn directly from 3-D point cloud. Frustum convnet [27] aggregates point cloud features into truncated volume level feature vectors to reduce the proposed 3-D space and improve the accuracy of 3-D object detection. AVOD [28] and MV3D [29] generate features by fusing LIDAR point clouds and RGB images, take them as input, and predict directional 3-D bounding box.

C. Monocular 3-D Object Detection Based on Pseudo-LIDAR Point Cloud

Recently, there are many methods [30]–[32] to obtain pseudo-LIDAR point cloud in RGB image and use the existing LIDAR-based detection algorithm to directly apply to monocular 3-D object detection. Their basic idea is to estimate the depth of RGB image to obtain the corresponding depth map and then transform the depth map into 3-D point cloud data representation by combining with camera parameters, which is referred to as the pseudo-LIDAR point cloud. The method proposed by Weng and Kitani [32] detected the 2-D object proposal in the input image, which extracts the cone of view from the pseudo-LIDAR for each proposal and then detects a directional 3-D bounding box for each cone. Ma *et al.* [33] proposed a multimodal feature fusion module, which fuses the complementary RGB information into the generated point cloud representation to enhance the discrimination ability of the point cloud. However, the conversion from depth to LIDAR depends heavily on the accuracy of the depth map and cannot use the semantic information of the RGB image. Therefore, our method uses a depth map as a guide to perform local convolution in RGB image, so as to learn better 2-D to 3-D representation.

III. Framework

Our overall framework is shown in Fig. 1. The 3-D object detection framework based on depth-guided local convolution consists of four parts, including depth estimation module, feature extraction module, depth-guided local convolution module, and 2-D–3-D detector module. Specifically, given an RGB image, we estimate the corresponding depth map and extract features from the RGB image and depth map, respectively. Then, the RGB image features and depth map features are input into the depth-guided local convolution module, and SKNet [5] is combined with the depth map features after the move operation to realize the depth-guided local convolution operation and obtain the target object features of different scales. Then, the obtained features are input into the 2-D–3-D detection head module for 3-D object detection, and the soft-NMS is used to estimate the best prediction box.

A. Depth Estimation Module

The whole network framework of this article consists of two parts: 1) depth estimation network and 2) 3-D object detection network. The depth map output by depth estimation network is used as the input of the 3-D object detection network. In order to provide better depth map input for 3-D object detection network, we use CBAM [35] instead of the original full image encoder in depth ordinal regression network [34] and then use the channel attention mechanism and spatial attention mechanism to capture the complete feature information and location information of the image, which improves the representation ability of global context information. Specifically, first, the feature information of larger pixels is better captured by global

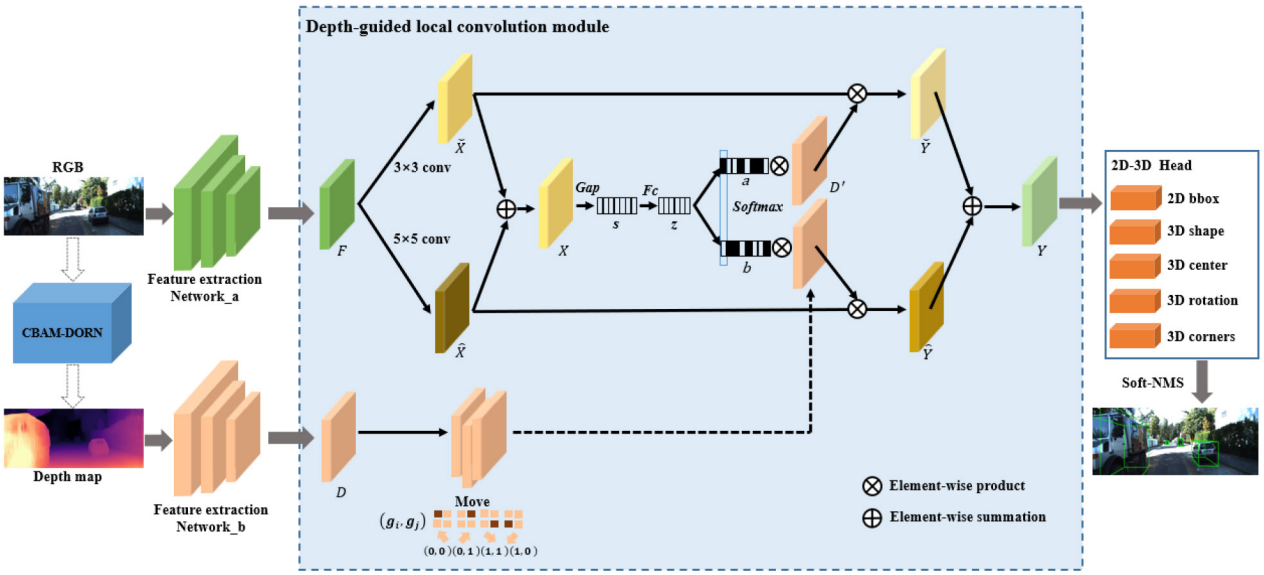


Fig. 1. Overall network framework. The input includes an RGB image and a depth image. Our network framework consists of four parts: depth estimation module, feature extraction module, depth-guided local convolution module, and 2-D–3-D detector module.

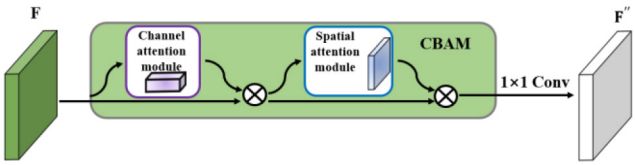


Fig. 2. Full image encoder based on CBAM.

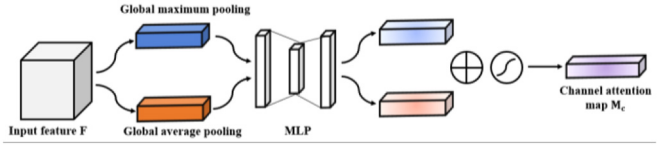


Fig. 3. Channel attention module.

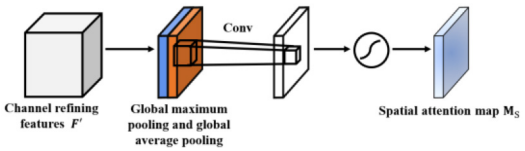


Fig. 4. Spatial attention module.

maximum pooling and global average pooling, and then the attention map generated by the spatial attention mechanism is multiplied by the original feature map instead of a simple copy operation to retain the complete location information. After the above improvements, the accuracy of the depth estimation network is improved, and a higher quality depth map is obtained, so as to provide the optimal depth map input data for the 3-D object detection network.

The full image encoder based on CBAM is shown in Fig. 2. The original feature F passes through the channel attention module and the spatial attention module in turn and finally passes through the 1×1 convolution layer to get the feature F''.

The channel attention module is shown in Fig. 3. First, the global maximum pooling and global average pooling are used to compress the input feature F in the spatial dimension to get two $C \times 1 \times 1$ channel attention vector, and C represents the channel number of the feature map. Then, the two vectors are sent into a shared multilayer perceptron (MLP) network composed of a hidden layer for calculation and generate two feature vectors with dimension $C \times 1 \times 1$. Further, the above two feature vectors are combined by element summation. Finally, the sigmoid function is used to map the combined feature vector to [0, 1], and then channel attention map M_c is obtained.

The channel attention module is shown in Fig. 4. First, the global maximum pooling and global average pooling operations are used for the feature F adjusted by the channel attention map in the channel dimension to obtain two feature maps of $1 \times H \times W$. The two feature maps are spliced into one feature map according to the channel. Then using a 7×7 convolution kernel to convolute the feature map, and it is mapped to [0, 1] by the sigmoid function, and a matrix of spatial attention weight is obtained which is the same dimension as the feature map. Finally, the spatial attention matrix is multiplied by F' to get the spatial attention map M_s.

After the channel attention map and spatial attention map are obtained, the channel attention map M_c and the input feature F are multiplied element by element to get F', then the spatial attention map M_s of F' is calculated, and the final feature F'' is obtained by elementwise multiplication. The whole process can be expressed as follows:

$$F' = M_c(F) \otimes F \tag{1}$$

$$F'' = M_s(F') \otimes F' \tag{2}$$

where $\otimes$ is elementwise multiplication.

B. Feature Extraction Module

As shown in Fig. 1, the feature extraction module is divided into two branches. The first branch is used for feature extraction of the RGB image, and the network backbone is ResNet-50 [36]. The last fully connected (FC) layer and pooling layers of the network are removed and are pretrained on the ImageNet classification dataset [37]. The second branch is used for feature extraction of the depth map. In order to reduce the computational cost, we only use the first three blocks of ResNet-50 as feature extraction network. For the next depth-guided local convolution module, each block of the two branches has the same number of channels.

C. Depth-Guided Local Convolution Module

As shown in Fig. 1, we use SKNet [5] to fuse RGB features with depth map features. First, the fusion layer feature $F \in \mathbb{R}^{C' \times H' \times W'}$ of RGB image features extracted by Feature extraction network_a is transformed to form two branch paths, and two feature maps $\check{X} \in \mathbb{R}^{C \times H \times W}$ and $\hat{X} \in \mathbb{R}^{C \times H \times W}$ with different scales are obtained. Specifically, two different transform functions are used. Each transform function passes through the convolution layer, batch normalization (BN) layer, and ReLu layer in turn. The convolution kernel sizes are 3×3 and 5×5, respectively. In order to improve the efficiency, we use the 3×3 convolution kernel with an expansion rate of 2 to replace the 5×5 convolution kernel.

After obtaining different scale RGB image features, we need to further obtain the global information of multiscale features, adjust the convolution kernel generated by depth map according to the global information, and then obtain different scale target objects. Therefore, we need to fuse the feature information of all branch paths. First, we combine the feature maps of the two branch paths by element summation

$$X = \check{X} + \hat{X} \tag{3}$$

Then, we use global average pooling to embed the global information into the feature map to generate the statistic $s \in \mathbb{R}^C$. Specifically, the cth element of s is obtained by compressing X in the space dimension of $H \times W$

$$s_c = \mathrm{Gap}(X_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_c(i, j) \tag{4}$$

Then, we need to get a feature vector $z \in \mathbb{R}^{d \times 1}$, which is used to fuse with the depth map features, so that the convolution kernel can carry the adaptive weight of different scale RGB image features and perform the adaptive selection of different spatial scale information. This step is implemented using an FC layer

$$z = Fc(s) = \delta(B(Ws)) \tag{5}$$

where δ is the ReLu function, B is BN, and $W \in \mathbb{R}^{d \times C}$. Here, the reduction rate γ is used to adjust the value of d

$$d = \max(C/\gamma, L) \tag{6}$$

where L represents the minimum value of d, which is set as $L = 32$ in the experiment.

Next, the feature vector z is used to generate the attention vector applied to the channel direction and then multiplied with the feature map of the branch path. Here, only two branch paths are created here, so two attention vectors need to be generated

$$a_c = \frac{e^{A_c z}}{e^{A_c z} + e^{B_c z}} \tag{7}$$

$$b_c = \frac{e^{B_c z}}{e^{A_c z} + e^{B_c z}} \tag{8}$$

where $A,\ B \in \mathbb{R}^{C \times d}$, and a and b represent the soft attention vectors of $\check{X}$ and $\hat{X}$ respectively. $A_c \in \mathbb{R}^{1 \times d}$ is the cth line of A, a_c is the cth element, and so is B_c and b_c.

In order to implement local convolution, we move the feature map extracted by feature extraction network_b in advance. The specific process is as follows: first, k is used to represent the size of convolution kernel, and then a moving grid $\{(g_i, g_j)\}$ containing $k \times k$ elements is defined, where $g \in (\mathrm{int})[1-k/2, k/2-1]$. We move the entire depth map feature map D to the direction and step size indicated by (g_i, g_j), and get the result $D(g_i, g_j)$. As shown in Fig. 1, we use D' to denote $D(g_i, g_j)$

$$D' = D\big(g_i, g_j\big). \tag{9}$$

Furthermore, the depth map feature map after moving [38] operation is fused with two attention vectors learned from RGB image features, respectively, and then fused with different scale RGB image features of two branch paths, respectively. The specific way of fusion is elementwise product operations. Finally, the feature maps of the two branch paths are fused by elementwise summation

$$Y = a_c \cdot D' \cdot \check{X} + b_c \cdot D' \cdot \hat{X} \tag{10}$$

where $Y = [Y_1, Y_2, \ldots, Y_C]$ and $Y_C \in \mathbb{R}^{H \times \mathrm{W}}$.

For different images, the depth-guided local convolution module performs local convolution on all pixels and enables neurons to adaptively adjust the size of the receptive field to obtain spatial information of different scales. This makes up for the defect that 2-D convolution cannot obtain local target scale information and fully integrates the semantic information of pseudo-LIDAR.

D. 2-D–3-D Detector Module

In this article, we use the single-stage detector [39], [40] based on the prior 2-D–3-D anchor box as the basic detector. The ground truth (GT) is defined with the following parameters.

1) 2-D bounding box $[x, y, w, h]_{\mathrm{2D}}$,where (x, y) denotes the coordinates of the center point of the 2-D box, and w and h denote the width and height of the 2-D box.
2) 3-D bounding box $[x, y, z]_{\mathrm{3D}}$ represents the position of the 3-D center in the camera coordinate system.
3) $[w, h, l]_{\mathrm{3D}}$ represents the size of the 3-D object, which are height, width, and length.
4) α_{3D} refers to the observation angle of the 3-D object, and the range is $[-\pi, \pi]$.

Input: we use the depth-guided local convolution module to output the feature map $Y_C \in \mathbb{R}^{H\times W}$ as input. 3-D to 2-D projection can be written as

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_P \cdot z_{3D} = k \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{3D} \tag{11}$$

where, $[x, y, z]_{3D}$ represents the horizontal position, height, and depth of the 3-D point in the camera coordinate system, and $[x, y]_P$ represents the projection coordinates of 3-D points in 2-D image coordinates.

Output: n_a is denoted as the number of anchors, and n_c is the number of categories. For each input position (i, j), the output parameters of the anchor are: $\{[t_x, t_y, t_w, t_h]_{2D}, [t_x, t_y]_P, [t_z, t_w, t_h, t_l, t_\alpha]_{3D}, t_C^{(m)}, s\}$, where $[t_x, t_y, t_w, t_h]_{2D}$represents the predicted 2-D boxes; $[t_x, t_y]_P$ represents the position of the 3-D angle of the projection on the 2-D plane, and $[t_z, t_w, t_h, t_l, t_\alpha]_{3D}$ is expressed as depth, predicted 3-D shape, and rotation angle. $t_C^{(m)} = \{[t_x^{(m)}, t_y^{(m)}]_P, [t_z^{(m)}]_{3D}\}$, $m \in \{1, 2, \ldots, 8\}$ represents eight projection 3-D angles. s is the classification score of each category. The actual output is an anchor-based conversion of 2-D–3-D boxes.

We use *a priori* 2-D–3-D anchor as the default anchor box. First, a 2-D–3-D anchor is defined in 2-D space, and then the corresponding prior in the training data set is used to calculate its part in 3-D space. Finally, the template anchor is defined by using the parameters of the two spaces. Furthermore, the template anchor is combined with the output based on a 2-D–3-D detector head, and the predicted 3-D box is obtained through data conversion.

Soft-NMS [6]: NMS is an important part of object detection. First, it sorts all detection boxes according to the score, selects the detection box with the largest score, and uses a predefined threshold to suppress all other detection boxes that significantly overlap with the largest detection box. This process is applied recursively to the remaining boxes. However, the traditional NMS has an obvious drawback: if a target is within a predefined overlap threshold, it will cause the target to be missed. Based on this, we use the soft-NMS algorithm instead of the traditional NMS algorithm. This algorithm attenuates the detection scores of all other objects to the weighting function that overlaps with the maximum score detection box. In this process, the target object is avoided to be eliminated, so as to obtain a more accurate 3-D bounding box. In the ideal case, when the maximum score detection box does not overlap with other detection boxes, the penalty function should have no penalty, while in the case of high overlap, there should be a very high penalty. Specifically, the Gaussian function is used to redefine the update principle of detection score in the traditional NMS algorithm

$$s_i = s_i e^{-\frac{iou(M,b_i)^2}{\sigma}} \tag{12}$$

where M represents the maximum score detection box, b_i represents the ith detection box in all initial detection boxes, $iou(M, b_i)$ represents the intersection and union ratio of the maximum score detection box and the initial detection box, and s_i is the test score.

E. Loss Functions

Loss Function: Total loss includes the classification loss, 2-D regression loss, 3-D regression loss, and 2-D–3-D angle loss

$$L = (1 - s_t)^\rho (L_{\text{class}} + L_{2d} + L_{3d} + L_{\text{corner}}) \tag{13}$$

where $\rho = 0.5$, and $L_{\text{class}}, L_{2d}, L_{3d}, L_{\text{corner}}$ represents the classification loss, 2-D regression loss, 3-D regression loss, and 2-D–3-D angle loss.

Specifically, the standard cross-entropy (CE) loss is used for classification

$$L_{\text{class}} = -\log(s_t). \tag{14}$$

In addition, 2-D and 3-D regression, using SmoothL1 regression loss

$$\begin{aligned} L_{2D} &= \text{Smooth}L1\left([x', y', w', h']_{2D}, [x, y, w, h]_{2D}\right) \\ L_{3D} &= \text{Smooth}L1\left([w', h', l', z', \alpha']_{3D}, [w, h, l, z, \alpha]_{3D}\right) \\ &\quad + \text{Smooth}L1\left([x', y']_P, [x, y]_P\right) \\ L_{\text{corner}} &= \frac{1}{8}\sum \text{Smooth}L1\left(\left[x'^{(m)}, y'^{(m)}\right]_{P,} [x^m, y^m]_P\right) \\ &\quad + \text{Smooth}L1\left(\left[z'^{(m)}\right]_{3D,} [z]_{3D}\right) \end{aligned} \tag{15}$$

where $[x^m, y^m]_P$ is the projection angle in the image coordinates of the GT 3-D box, and $[z]_{3D}$ is its GT depth.

F. Joint Training

In order to further improve the accuracy of the 3-D object detection network, in the training phase, we use the joint training method to train the depth estimation network and 3-D object detection network. Specifically, the real depth map of the KITTI dataset is used as the input to train the 3-D object detection network. After training, the parameters of the 3-D object detection network model are fixed. Then, the output depth map of the depth estimation network is sent to the 3-D object detection network with fixed parameters as input, and then the parameters of the depth estimation network are adjusted to make the depth estimation network reach the best state, and then the trained parameters of the depth estimation network model are fixed. Finally, a single RGB image is an input into the depth estimation network model with fixed parameters to get the depth map, which is further input into the 3-D object detection network for 3-D object detection. The joint training method solves the problem of error accumulation in segmented training, makes the two networks constrain each other, obtains the appropriate network parameters, and then makes the network reach the best state.

IV. Experiment

A. Experimental Setting

We use the KITTI 3-D object detection data set [41] to carry out our experiment, which includes 7481 training images,

TABLE I
COMPARISON RESULTS OF THE KITTI 3-D OBJECT DETECTION DATA SET

Method	Test set			Split1			Split2		
	Easy	**Moderate**	Hard	Easy	**Moderate**	Hard	Easy	**Moderate**	Hard
OFT-Net[9]	1.61	1.32	1.00	4.07	3.27	3.29	-	-	-
GS3D[42]	4.47	2.90	2.47	13.46	10.97	10.38	11.63	10.51	10.51
MonoGRNet[43]	9.61	5.74	4.25	13.88	10.19	7.62	-	-	-
MonoPSR[17]	10.76	7.25	5.85	12.75	11.48	8.59	13.94	12.24	**10.77**
SS3D[44]	10.78	7.68	6.51	14.52	13.15	**11.85**	9.45	8.42	7.34
MonoDIS[15]	10.37	7.94	6.40	11.06	7.60	6.37	-	-	-
Ours	**11.10**	**8.24**	**6.76**	**15.39**	**13.48**	10.69	**14.55**	**12.90**	10.57

7518 test images, point clouds, and calibration parameters. There are 80 2-D labeled objects and 256 3-D labeled objects, and the object categories are divided into car, pedestrian, and cyclist. Each 3-D GT box contains three difficulty categories: easy, moderate, and hard. KITTI has two train–val segmentation: 1) split1 contains 3712 training images and 2) 3769 validation images, while split2 contains 3682 training images and 3799 validation images. The network is implemented by using the pytorch framework. During training, the input image size is set to 512×1760, and the horizontal flipping method is used for data enhancement. The stochastic gradient descent (SGD) optimizer is used, the momentum is set to 0.9, the weight reduction parameter is set to 0.0005, and the learning rate is set to 0.01. The 11-point interpolated average precision (AP) metric $\text{AP}|_{R_{11}}$ is separately computed on each difficulty class and each object class. After that, the 40 recall position-based metric $\text{AP}|_{R_{40}}$ is used instead of $\text{AP}|_{R_{11}}$.

B. Comparison Results

Table I shows the comparison results between the proposed method and several advanced monocular 3-D object detection methods for cars in the KITTI dataset. Among them, OFT-Net [9] establishes orthogonal transformation of image features and 3-D spatial features according to the corresponding relationship between image and 3-D space. The feature map based on the image is backprojected into the aerial view of 3-D space, and then using residual network unit to process the aerial view feature map to realize 3-D object detection. GS3D [42] first uses 2-D detection results to obtain rough 3-D bounding boxes for the object. Then using the projection on the 2-D image obtain the 3-D structure information, and fusing surface features to further refine to get the accurate 3-D bounding boxes. MonoGRNet [43] detects 3-D objects in monocular images through geometric reasoning in observed 2-D projection and unobserved depth dimensions and decomposes monocular 3-D object detection task into four subtasks, including 2-D object detection, case-level depth estimation, projection 3-D center estimation, and local corner regression. Using the basic relationship of the pinhole camera model, the MonoPSR [17] uses a mature 2-D object detector to generate 3-D proposal for each object in the scene. At the same time, the point cloud is predicted in the object centered coordinate system, local dimension and shape information are learned, and the shape information is used to guide 3-D positioning, thus improving the 3-D positioning accuracy. SS3D [44] uses CNN to generate category scores and regress a set of intermediate values to estimate an initial 3-D borders, then, the redundant detections of 3-D borders are discarded by using NMS, the 3-D borders are further refined by the nonlinear least-square method. The main idea of MonoDIS [15] is to design an end-to-end 3-D point cloud learning encoder using 2-D convolution, which is suitable for point cloud learning. Learn the features of the pillars (vertical columns) of the point cloud to predict the 3-D bounding box for the object. It can be seen that in the most representative moderate category, the accuracy of our method is significantly improved compared with other methods, 8.24 is 3.6% higher than 7.94. Our model achieves the best 3-D detection results by using joint training, which verifies the effectiveness of joint training. We not only give the accuracy comparison results of $\text{AP}|_{R_{40}}$ but also give the accuracy comparison results of $\text{AP}|_{R_{11}}$ in split1/split2 for fair comparison. In addition, it can be seen from Table I that our results are not the best in the hard category. That is because of our proposed depth-guided local convolution module can obtain the 3-D point cloud and high-level semantic information of the specific target object, but in the case of severe occlusion and truncation, if the target object only shows small local details, it will increase the difficulty of obtaining the semantic information, which will affect the effect of our algorithm. However, in the research of 3-D object detection, researchers regard the moderate category as the most representative index. Therefore, in the moderate category, our results have the highest accuracy in several related algorithms.

C. Ablation Study

Table II shows the results of the ablation study. We mainly studied ablation in five aspects: 1) *basic 3-DNet:* basic network model without depth information; 2) *+D:* add depth information (no move operation); 3) *+DLCM:* add depth-guided local convolution module (add depth information and move operation at the same time); 4) the method of segmented training (without joint training) was adopted; and 5) *ours (+ SN):* add the full model of soft-NMS, where σ is set to 0.5. Phased training means that the monocular depth estimation network and the 3-D object detection network are trained separately. The monocular depth estimation network is trained first,

TABLE II
ABLATION STUDY OF CAR IN SPLIT1

Method	$AP\|_{R_{11}}$ Easy	**Moderate**	Hard	$AP\|_{R_{40}}$ Easy	**Moderate**	Hard
3DNet	8.92	5.25	3.84	7.52	3.13	1.46
+D	11.42	9.87	6.45	8.21	3.96	2.14
+DLCM	13.57	11.24	8.74	9.76	5.45	3.75
Phased training	13.53	11.45	8.85	9.22	6.23	4.36
Ours(+SN)	**15.39**	**13.48**	**10.69**	**12.89**	**8.34**	**5.97**

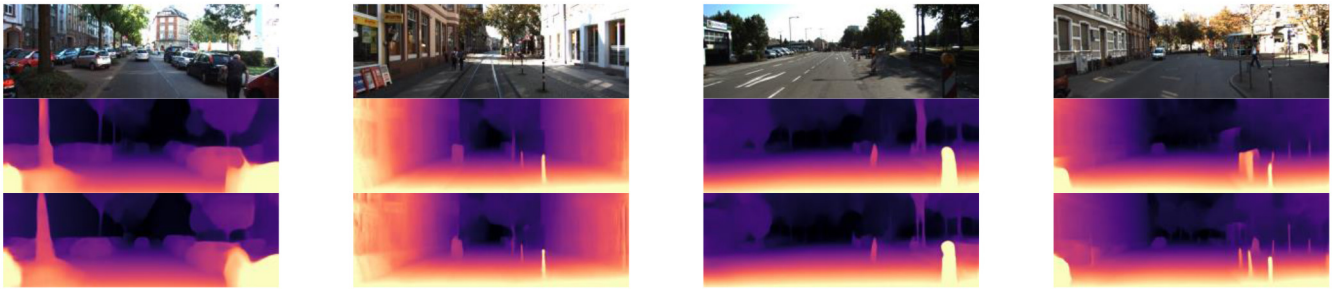


Fig. 5. Depth map visualization results. Four RGB images from the KITTI data set were used for visual comparison. The first line is the RGB original image, the second line is the depth map directly output by the depth ordinal regression network, and the last line is the depth map output after joint training.

Fig. 6. Visualization results of 2-D box and 3-D box of 3-D object detection. Four RGB images of the KITTI data set are used for visualization. The first line is the RGB original image, the second line is the 2-D box visualization result, and the last line is the 3-D box visualization result.

and then the 3-D object detection network is trained. However, this training method will cause the error accumulation of the two networks, that is, the error of the monocular depth estimation network will directly affect the accuracy of the 3-D object detection network, resulting in the network state is not the best, so we use the joint training method to optimize the two networks. It can be seen from Table II that with the addition of different components, the detection accuracy is continuously improved, showing the contribution of each component. In the moderate category, the accuracy of $AP|_{R_{40}}$ and $AP|_{R_{11}}$ of the depth-guided local convolution module in the model ranges from 5.25 to 11.24 and 3.13 to 5.45, respectively, which shows the effectiveness of the depth-guided local convolution module. After the addition of soft-NMS, the accuracy increases by 2.13 and 2.89 percentage points, respectively, which shows the effectiveness of soft-NMS. Among them, we make statistics on the accuracy of segmented training, and we can see that the accuracy of the joint training method is significantly improved.

D. Visualization

In this section, we mainly show the visualization results of two parts: 1) in the joint training, the trained 3-D object detection network model is used to visualize the depth map after adjusting the depth estimation network and 2) visualization results of 3-D object detection. In the first part, we mainly compare the depth map directly output by depth estimation network with the depth map output after joint training. Fig. 5 shows the results of depth map output directly by the depth ordinal regression model and the results of depth map output after joint training. It can be observed that the depth map output by the depth ordinal regression model after joint training performs better in the far target and object details and provides better depth map input for the 3-D object detection network.

The second part mainly shows the visualization results of the 2-D and 3-D boxes of the target, as shown in Fig. 6. It can be seen that our model can achieve good detection results for single target and multiple targets of pedestrians (pictures 1 and 3). In the case of truncation and occlusion, it can also achieve better detection results (pictures 2 and 4). This is mainly due to the use of depth information and adaptively adjusting the convolution kernel according to the multiscale input information to guide the 2-D convolution to better represent the 3-D spatial information, so as to capture the target objects of different scales, and the soft-NMS algorithm can avoid the loss of the object detection box and get the best prediction box.

V. Conclusion

In this article, we introduce a depth-guided local convolution method for monocular 3-D object detection. In our model, we combine the information of two modes (RGB image mode and depth mode) to generate a local convolution kernel by combining SKNet with depth map and applied to the single RGB image. According to the multiscale input information, we adaptively adjust the convolution kernel to capture the target objects of different scales, thus narrowing the gap between the 2-D image representation and the 3-D space representation. In addition, the soft-NMS algorithm is used to replace the traditional NMS algorithm, and the Gaussian function is used to punish the discontinuous overlaps properly, so as to avoid the loss of the detection box and obtain the accurate detection box. Finally, the joint training method is used to train the depth estimation network and the 3-D object detection network to achieve the best state of the two networks. A large number of comparative experiments are evaluated on the KITTI 3-D object detection data set, compared with the prestigious monocular 3-D object detection methods, the accuracy is improved by 3.6%.

References

[1] J. Li, S. Dang, M. Wen, X.-Q. Jiang, Y. Peng, and H. Hai, "Layered orthogonal frequency division multiplexing with index modulation," *IEEE Syst. J.*, vol. 13, no. 4, pp. 3793–3802, Dec. 2019.

[2] Q. Li, M. Wen, M. D. Renzo, H. V. Poor, S. Mumtaz, and F. Chen, "Dual-hop spatial modulation with a relay transmitting its own information," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4449–4463, Jul. 2020.

[3] X. Pei, H. Yu, M. Wen, S. Mumtaz, S. A. Otaibi, and M. Guizani, "NOMA-based coordinated direct and relay transmission with a half-duplex/full-duplex relay," *IEEE Trans. Commun.*, vol. 68, no. 11,pp. 6750–6760, Nov. 2020.

[4] B. Zheng *et al.*, "Design of multi-carrier LBT for LAA&WiFi coexistence in unlicensed spectrum," *IEEE Netw.*, vol. 34, no. 1, pp. 76–83, Jan./Feb. 2020.

[5] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 510–519.

[6] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, 2017, pp. 5561–5569.

[7] A. Neubeck and L. V. Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3. Hong Kong, 2006, pp. 850–855.

[8] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 7074–7082.

[9] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3D object detection," 2018, *arXiv:1811.08188*.

[10] 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.*, Las Vegas, NV, USA, 2016, pp. 2147–2156.

[11] T. He and S. Soatto, "Mono3D++: Monocular 3D vehicle detection with two-scale 3D hypotheses and task priors," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8409–8416.

[12] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, "Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 2040–2049.

[13] 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.*, Long Beach, CA, USA, 2019, pp. 2069–2078.

[14] A. Kundu, Y. Li, and J. M. Rehg, "3D-RCNN: Instance-level 3D object reconstruction via render-and-compare," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 3559–3568.

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

[16] A. Simonelli, S. R. Bulo, L. Porzi, M. L. Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection: From single to multi-class recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Sep. 18, 2020, doi: 10.1109/TPAMI.2020.3025077.

[17] 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.*, Long Beach, CA, USA, 2019, pp. 11867–11876.

[18] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 652–660.

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," 2017, *arXiv:1706.02413*.

[20] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 4490–4499.

[21] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 7652–7660.

[22] 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.*, 2017, pp. 2980–2988.

[23] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-dense 3D object detector for point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 1951–1960.

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

[25] 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.*, Long Beach, CA, USA, 2019, pp. 12697–12705.

[26] 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 Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 918–927.

[27] Z. Wang and K. Jia, "Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection," 2019, *arXiv:1903.01864*.

[28] 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)*, Madrid, Spain, 2018, pp. 1–8.

[29] 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.*, Honolulu, HI, USA, 2017, pp. 1907–1915.

[30] 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.*, 2019, pp. 8445–8453.

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

[32] X. Weng and K. Kitani, "Monocular 3D object detection with pseudo-LiDAR point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, Seoul, South Korea, 2019, pp. 857–866.

[33] 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.*, Seoul, South Korea, 2019, pp. 6851–6860.

[34] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 2002–2011.

[35] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[38] M. Ding *et al.*, "Learning depth-guided convolutions for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Seattle, WA, USA, 2020, pp. 1000–1001.
[39] 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.*, Las Vegas, NV, USA, 2016, pp. 779–788.
[40] W. Liu *et al.*, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[41] 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.*, Providence, RI, USA, 2012, pp. 3354–3361.
[42] 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.*, Long Beach, CA, USA, 2019, pp. 1019–1028.
[43] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A geometric reasoning network for monocular 3D object localization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8851–8858.
[44] A. Limaye, M. Mathew, S. Nagori, P. K. Swami, D. Maji, and K. Desappan, "SS3D: Single shot 3D object detector," 2020, *arXiv:2004.14674*.

Jun Li (Member, IEEE) received the B.S. degree from South Central University for Nationalities, Wuhan, China, in 2009, and the Ph.D. degree from Chonbuk National University, Jeonju, South Korea, in 2016.

He is currently an Associate Professor with Guangzhou University, Guangzhou, China. He has published more than 50 papers in refereed journals and conference proceedings. His research interests include spatial modulation and OFDM with index modulation.

Dr. Li serves as a reviewer for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

Wei Song received the M.S. degree in soft engineer from Dalian University of Technology, Dalian, China, in 2005, and the Ph.D. degree from Chonbuk National University, Jeonju, South Korea, in 2010.

He is currently working with the Applied Technology College of Soochow University, Suzhou, China, as a Distinguished Professor. His research interests include spatial modulation, MIMO, STBC, and reconfigurable intelligent surface.

Yongbin Gao received the Ph.D. degree from Jeonbuk National University, Jeonju, South Korea, in 2017.

He is currently an Associate Professor with the School of Electronic and Electrical Engineering, and the Vice Director with the Next Generation Intelligent Research Center, Shanghai University of Engineering Science, Shanghai, China. He has published 30 SCI papers in prestigious journals, such as IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Information Science and Pattern Recognition Letters*, in the area of image processing, patter recognition, and computer vision.

Huixing Wang is currently pursuing the master's degree with the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China.

His research interests include 3-D object detection, computer vision, and machine learning.

Yier Yan received the B.S. degree in applied electronics from South-Central University for Nationalities, Wuhan, China, in 2003, and the Ph.D. degree in communication engineering from Chonbuk National University, Jeonju, South Korea, in 2010.

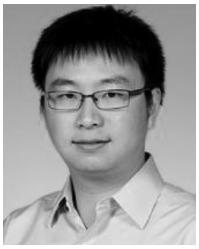
He is currently working with the School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou, China. His research interests include information theory, signal processing, and OFDM in MIMO system.

Bo Huang was born in Hubei, China, in 1985. He received the M.Sc. and Ph.D. degrees in computer science from Wuhan University, Wuhan, Hubei, China, in 2010 and 2014, respectively.

He is currently an Associate Professor with Shanghai University of Engineering Science, Shanghai, China. His current research interests include artificial intelligence, machine learning, data analytics, and the application of soft computing.

Jun Zhang received the B.S. and M.S. degrees in electronic and communication engineering from the Sun Yat-sen University, Guangzhou, China, in 1997 and 2000, respectively, and the Ph.D. degree in electronic and communication engineering from the South China University of Technology, Guangzhou, in 2003.

He joined the South China University of Technology as a Lecturer with the School of Electronic and Communication Engineering in 2003, where he is currently an Associate Professor. His research interests are speech signal processing and underwater acoustic communication.



Wei Wang (Member, IEEE) received the B.Engr. and M.Engr. degrees from the Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2007 and 2010, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, in 2015.

Since 2015, he has been with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology (HKUST), Hong Kong, where he is currently an Associate Professor and also affiliated with the HKUST Big Data Institute. His research interests cover the broad area of distributed systems, with focus on serverless computing, machine learning systems, and cloud resource management. He published extensively in the premier conferences and journals of his fields.

Dr. Wang has won the Best Paper Runner Up Awards of IEEE ICDCS 2021 and USENIX ICAC 2013. He routinely serves on the Program Committees of Leading Conferences and was named the Distinguished TPC Member of IEEE INFOCOM in 2018–2020.