Real-Time and Efficient 6-D Pose Estimation From a Single RGB Image

Jun Cheng, *Member, IEEE*, Penglei Liu, Qieshi Zhang, *Member, IEEE*, Hui Ma, Fei Wang, and Jin Zhang

***Abstract*— 6-D pose estimation is an important branch in the field of vision measurement and is widely used in the fields of robotics, autonomous driving, and reality augmentation. The latest research trend in 6-D pose estimation is to train a deep neural network to directly predict the 2-D projection position of the 3-D keypoint from the image, establish the corresponding relationship, and, finally, use the perspective-n-point (PnP) algorithm to perform pose estimation. The current challenge of pose estimation is that, when objects are textureless, occluded, or scene-cluttered, the detection accuracy is reduced, and most of the existing algorithm models are large and cannot accommodate real-time requirements. In this article, we introduce a densely connected feature pyramid network (DFPN) that can efficiently integrate and utilize features. We combine the cross-stage partial network (CSPNet) with DFPN to design a new network for 6-D pose estimation, DFPN-6-D, a new approach for 6-D object pose estimation. DFPN-6-D can efficiently and accurately handle objects with textureless, occluded, and scene clutter and estimate their full 6-D poses in a single shot. Furthermore, we propose a new confidence calculation method and loss function for object pose estimation, which can fully consider spatial information. Finally, we propose a novel augmentation method for direct 6-D pose estimation approaches to improve performance and generalization ability in the case of occlusion, which is called 6-D augmentation. Our approach achieves a new state-of-the-art accuracy of 98.06 and 87.09 in terms of the ADD(-S) metric on the Linemod dataset and the Occluded-Linemod dataset, and our method also achieves the best result in terms of the different metric on the MULT-I dataset, the BIN-P dataset, and the T-LESS dataset, respectively, while still running end-to-end at over 65 frames/s. The experimental results demonstrate that our algorithm is robust to textureless materials and occlusion while running more efficiently than other methods. We also deploy our proposed method to a real robot to grasp and manipulate objects based on the estimated pose.**

***Index Terms*— 6-D pose estimation, augmentation method, densely connected feature pyramid network (DFPN), occluded, robotics, textureless.**

Manuscript received May 1, 2021; revised August 27, 2021; accepted September 12, 2021. Date of publication September 24, 2021; date of current version October 8, 2021. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2019B090915001; in part by the National Natural Science Foundation of China under Grant U1713213, Grant U1913202, and Grant U1813205; in part by Shenzhen Technology Project under Grant JCYJ20180507182610734 and Grant JSGG20191129094012321; and in part by the CAS Key Technology Talent Program. The Associate Editor coordinating the review process was Yan Zhuang. *(Corresponding author: Penglei Liu.)*

Jun Cheng, Penglei Liu, Qieshi Zhang, and Fei Wang are with Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, also with Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen, China, and also with The Chinese University of Hong Kong, Hong Kong, SAR, China (e-mail: jun.cheng@siat.ac.cn; pl.liu@siat.ac.cn; qs.zhang@siat.ac.cn; fei.wang@siat.ac.cn).

Hui Ma is with the CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, and also with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, SAR, China (e-mail: hui.ma@siat.ac.cn).

Jin Zhang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518061, China (e-mail: jin.zhang@szu.edu.cn).

Digital Object Identifier 10.1109/TIM.2021.3115564

I. Introduction

THE 6-D pose estimation uses a camera as a sensor to measure objects with computer vision methods [1], [4], [7], [8], [27]. It is an important part of the field of vision measurement and is widely used in industrial measurement and robots [3], [11]. Although 3-D object detection [47], [48] can also be used for industrial measurement, it cannot provide the 6-D pose information of the object to assist the robot in grasping. 6-D object pose estimation is used to calculate the transformation from the world coordinate system of the object to the camera coordinate system [2]. The existing 6-D pose estimation methods are mainly divided into two types: those based on depth information and those based on red-green-blue (RGB) information. Although the current methods for pose estimation using RGB-D cameras are robust, depth cameras are suitable only for indoor scenes and are power-hungry [4], [5], [10], [12]. In contrast, RGB cameras are suitable for a wider range of scenes and power savings [16], [17], [19], [26]. Therefore, in this work, we use only RGB images to estimate the 6-D pose of the object.

In this field, the latest pose estimation method establishes a correspondence between the known 3-D model of the object and the 2-D pixel position and then uses the perspective-n-point (PnP) algorithm [17] to calculate six pose parameters. This method is robust and accurate when the object is well textured but may fail when the object has no texture, the scene is cluttered, or the objects occlude each other [8], [11]. BB8 [16] and single shot multibox detector (SSD)-6-D [14] have achieved good results in terms of accuracy, but they cannot achieve real-time requirements due to their multistage steps. The existing benchmark datasets Linemod and Occluded-Linemod are small and cannot readily satisfy the training of large neural networks. Many existing data enhancement methods proposed for these datasets rely on 2-D image enhancement methods [31], [34] without considering the characteristics of 6-D pose estimation and without data enhancement in terms of occlusion.

Fig. 1. Example image of multiobject pose estimation. The green 3-D bounding boxes represent the ground-truth poses, and the 3-D bounding boxes of other colors are the poses estimated by our approach. The poses of objects estimated by our approach are in good agreement with the ground truth.

In this article, we propose a densely connected feature pyramid network (DFPN), which can efficiently and fully integrate features. Generally, when we use the network to detect objects, the shallow network has high resolution and learns the detailed features of the picture, while the deep network has low resolution and learns more semantic features. The main function of the feature pyramid network (FPN) is to fuse and represent features of different scales. BiFPN [28] is one of the most advanced feature networks. It adds a forward residual structure to the FPN so that features can be better used in forward propagation, but it does not consider the propagation efficiency of semantic features and detailed features in the vertical direction. As the result, some semantic features and detailed features will be lost in the process of vertical propagation. In order to improve the efficiency of feature utilization, we add a densely connected network structure to the vertical propagation part of BiFPN so that semantic features and detailed features can be fully utilized in vertical propagation. Our DFPN can effectively deal with the problem of insufficient features for occluded objects.

In the object detection area, researchers mostly employ the backbone network to extract features from raw images. CSPNet [22] is a current state-of-the-art network that optimizes the process of gradient backpropagation while maintaining the network size and computational cost. Herein, we propose a novel object pose detection architecture, DFPN-6-D, that combines DFPN mentioned above and CSPNet to efficiently predict the object pose. As such, DFPN-6-D is characterized by lightweight model size and capabilities of detecting occluded or textureless objects. Our DFPN-6-D is suitable for real-time pose estimation and achieves state-of-the-art performance using RGB as input on the widely used benchmark datasets Linemod and Occluded-Linemod. Our approach can detect the 3-D b-box of an object and can estimate its 6-D poses within a single shot. When we regress the 6-D pose, our approach is end-to-end and does not require subsequent processing steps. This makes our method very fast and achieves real-time requirements. Most existing methods calculate the offset between projection points and the ground truth as a measure of confidence; the problem is that the offset in the 2-D plane may be small, but, in 3-D space, the offset is very large, resulting in inaccurate confidence. To solve this problem, we propose a new confidence algorithm that projects the 3-D bounding box of the target object into the front view, end view, and vertical view, calculates the complete intersection over union (CIoU), and then uses the weighted average as the final confidence. The new confidence algorithm is named fusion CIoU (FCIoU).

Finally, we propose an augmentation method for 6-D pose estimation, which can enrich the existing Linemod and Occluded datasets. When we train the model, the process of training the model requires a large amount of label data, and the benchmark dataset Linemod contains insufficient labeled data, which results in the existing amount of data being unable to fully train the model. For this situation, we propose a method for data enhancement in the field of 6-D pose estimation, which can effectively expand the small dataset. To improve the accuracy of the model in detecting occluded objects, we introduced a random erasure method to enrich the Occluded-Linemod dataset.

In summary, our key contributions are listed as follows.

1) We propose DFPN that can efficiently and fully integrate features of forward propagation and vertical propagation, and effectively deal with occlusion problems.
2) We propose DFPN-6-D that can estimate the pose of objects quickly. DFPN-6-D offers high accuracy, scalability, efficiency, and robustness. At the same time, we propose a new confidence calculation method for network training, which can fully consider spatial information.
3) We propose a data enhancement method for 6-D pose estimation, which can enrich the Linemod and Occluded datasets.

II. Related Work

In this section, we briefly summarize existing projects related to 6-D pose estimation based on RGB images. We broadly divide existing projects into three categories: template-based methods, voting-based methods, and keypoint-based methods. Moreover, we briefly summarize the data enhancement methods in the field of 6-D pose estimation. Here, we briefly review some representative methods in each category.

A. Template-Based Method

The 6-D pose estimation method [6], [7] based on template matching refers to taking pictures of the target object from various angles, manually marking the pose of the object in each image, and then comparing the target image with all templates during pose estimation to find the most similar template. The posture information marked by the template is used as the posture information of the object in the target image. Therefore, we can transform the 6-D pose estimation problem into an image retrieval problem. The template-based method can effectively detect textureless objects [9], [12].

However, when the target object is occluded, the complete information of the object cannot be obtained, so the object cannot effectively match the template, which leads to the poor effect of the method.

B. Voting-Based Method

To solve the abovementioned problem, some approaches [23] adopt a pixelwise voting scheme, in which each pixel of the target object predicts a vector pointing to a keypoint. After each pixel has voted for the keypoint, the random sample consensus (RANSAC) algorithm is used to determine the final keypoint. This method can effectively deal with the occlusion problem and has good robustness. The dense 2-D–3-D correspondence method means that each pixel contained in the object predicts its corresponding 3-D model point. Dense 2-D–3-D correspondence can be obtained using UV maps [24] or by regressing the coordinates in the 3-D model space of the object. Then, RANSAC is used to calculate the final keypoints, establish a 2-D–3-D correspondence, and, finally, calculate the 6-D pose through the PnP algorithm. Although these methods achieve good 6-D pose estimation accuracy, all of them are also relatively slow in terms of object pose estimation, and the runtime linearly increases with additional objects. This is because pixel-by-pixel voting necessitates calculations for each pixel, which requires considerable time.

C. Keypoint-Based Method

Most of the recent works in the field of 6-D pose estimation with images rely on a method that detects the 2-D keypoint of the object of interest in the image and then establishes a correspondence with the keypoint of the 3-D model. For objects with rich textures, traditional methods can detect the local texture of the object as a keypoint and still have good robustness in the case of occlusions and cluttered scenes [7], [9], [20]. Although these methods can effectively process objects with rich textures, they perform poorly when processing poorly textured objects and low-resolution images [11], [15], [18]. To solve the shortcomings of object pose estimation based on keypoints in the traditional field, many researchers use convolutional neural networks (CNNs) as keypoint detectors to detect the keypoints of objects in RGB images. The intensive prediction method is used to predict key points [23]. Although the prediction accuracy is very good, the speed is insufficient. Some methods, such as [14] and [17], infer 6-D pose information by predicting the projection coordinates of objects in 2-D space, which is equivalent to applying 2-D detection tasks to 6-D pose estimation tasks. There are many advanced algorithms for 2-D target detection tasks, such as faster-region convolution neural network (RCNN) and You Only Look Once (YOLO) [13] and a series of their variants. These algorithms have achieved good results in 2-D target detection tasks. This inspired us to design a CNN to complete the 6-D pose estimation of the object by predicting the projected coordinates of the object in the 2-D space. Some algorithms have achieved good results in terms of speed, such as YOLO-6-D [17], which is a single-shot deep CNN that takes images as input and directly estimates the object pose without any postprocessing. As it is based on the YOLO-V2 network [13], it inherits the advantages of YOLO-V2's speed, but it is not effective in estimating small objects or occluded objects.

D. 6-D Augmentation Methods

Recently, there have been many data enhancement methods in image detection, such as image rotation, enlargement, reduction, changing the color of the image, and so on [29], [30], but there are few data enhancement methods in the field of 6-D pose estimation [17], [31], and there are only rotation, scaling, and color changes, no multiangle transformations, and no data enhancement on the occlusion datasets.

For these reasons, these existing methods are not yet suitable for multiple objects and real-time requirements, which limits their application in many real-world scenarios. To effectively address weak texture and occlusion problems while taking into account the running speed, we combined CSPNet and DFPN to form the DFPN-6-D network for pose estimation. In addition, we propose a data enhancement method for 6-D pose estimation, which can enrich the existing Linemod and Occluded datasets. Validated by experiments, our method achieves the requirements of speed and accuracy at the same time.

III. Approach

In this section, we first introduce the main ideas for our proposed DFPN, an efficient DFPN, and then introduce the DFPN-6-D network that is an end-to-end trainable network that predicts the 6-D pose in real time. Finally, we introduce FCIoU that is a new confidence calculation method for object pose estimation.

A. Network Architecture

1) DFPN: One of the main difficulties in object detection is how to effectively represent and process multiscale features. Earlier detectors often directly performed predictions based on the pyramidal feature hierarchy extracted from backbone networks. As shown in Fig. 2(a), Liu *et al.* [38] proposed a feature pyramid network (FPN), which is a top-down pathway to combine multiscale features, but FPN [38] disseminates information from top to bottom only and cannot disseminate information from bottom to top. To solve this problem, Liu *et al.* [39] added an additional path that can spread information from bottom to top on the basis of FPN and finally proposed the PANet structure, which can effectively solve the problem of one-way information flow. As shown in Fig. 2(b), although PANet shows good accuracy in the field of object detection, it requires more parameters and computational costs. To improve the operating efficiency of the model and reduce the parameters, Google researchers proposed a multiscale feature fusion method [28] and then proposed the BiFPN [as shown in Fig. 2(c)], offering efficient bidirectional cross-scale connections and weighted feature fusion. BiFPN achieves better accuracy and lower cost than PANet. BiFPN [28] is one of the most advanced feature networks, but it considers only the problem of forward feature propagation, without considering the problem of vertical propagation of features, which results in loss of features during vertical

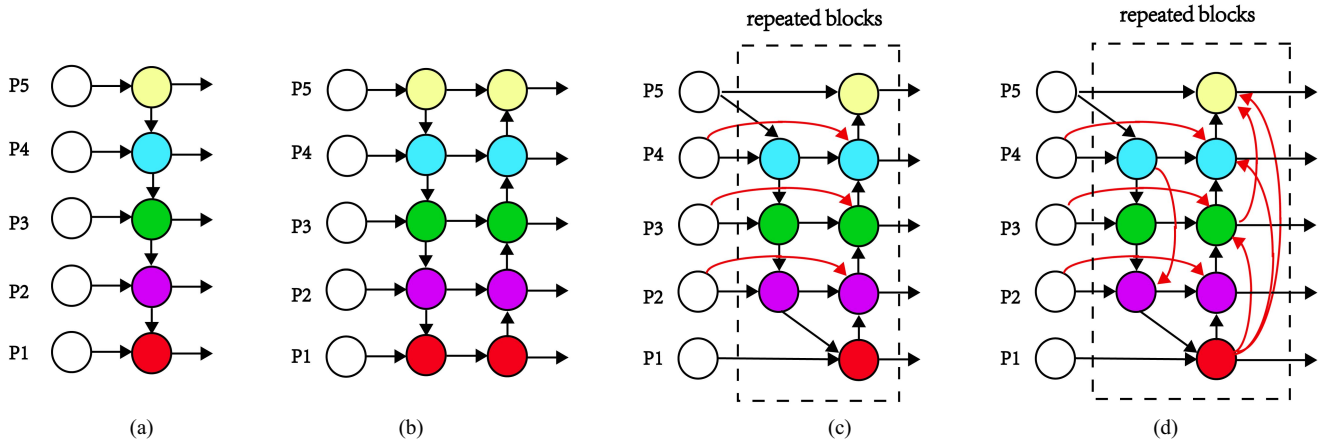


Fig. 2. Feature network design. (a) FPN introduces a top-down pathway to fuse multiscale features from level 1 to 5 (P1–P5). (b) PANet adds an additional bottom-up pathway on top of FPN. (c) BiFPN adds a residual structure to the lateral propagation direction of PANet. (d) DFPN is our proposed network. It adds a residual structure to the longitudinal propagation direction of PANet to achieve better accuracy and efficiency tradeoffs.

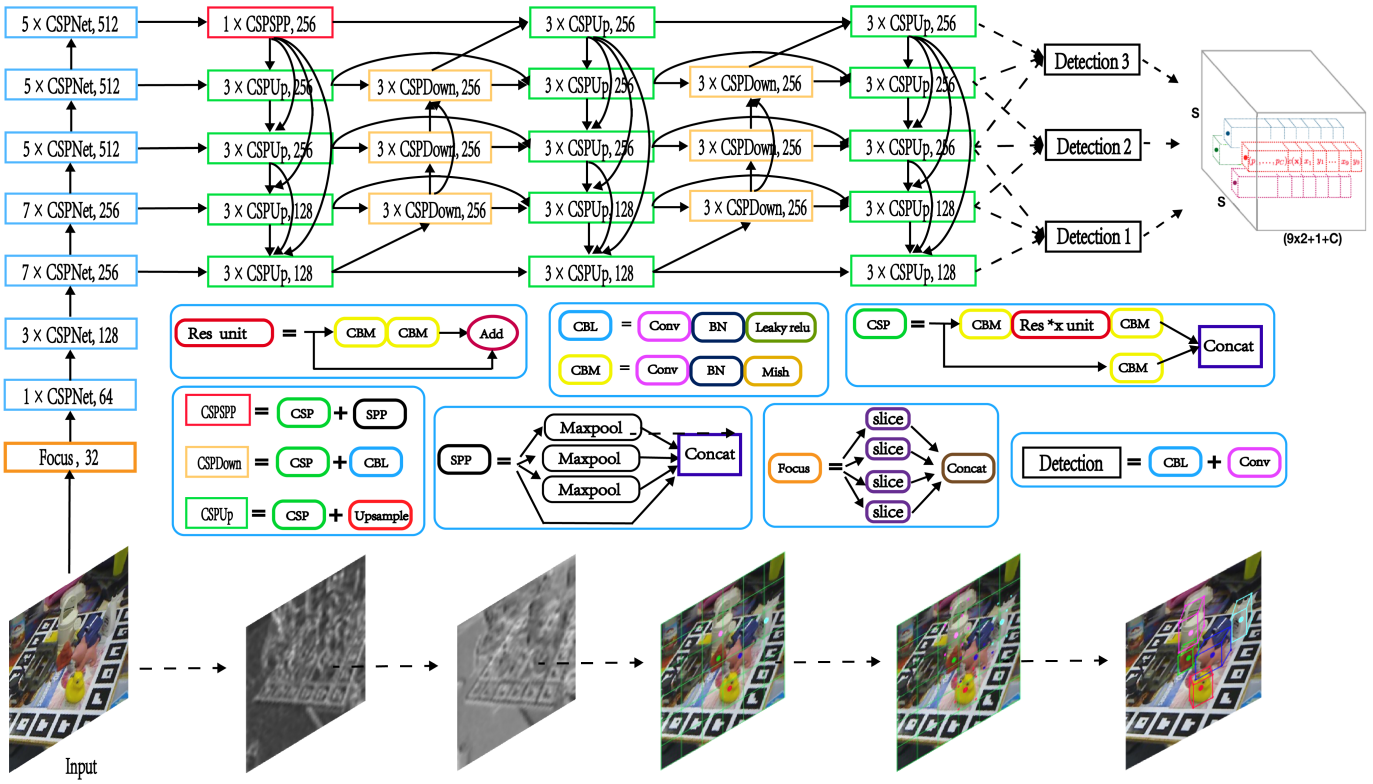


Fig. 3. We combined CSPNet with DFPN to design a new network for 6-D pose estimation, DFPN-6-D. Conv represents the convolutional layer. CBL represents the convolutional layer + BN layer + Leaky relu. CBM represents the convolutional layer + BN layer + Mish activation layer. Res unit represents a residual structure composed of two CBM modules. CSP represents a residual structure composed of the Res unit module and CBM module. SPP represents a pyramid structure. Focus represents the focus module. The bottom of the figure shows the whole process from the input map to the feature map to the output structure.

propagation and the inability to effectively use all feature information.

Therefore, to improve the utilization rate of input and extract effective features more fully, we apply the ideas of residual networks and densely connected convolutional networks to the feature network. Finally, a new feature fusion extraction network is proposed. The new feature network is named DFPN, as shown in Fig. 2(d). DFPN can improve the utilization of features in forward and vertical propagation.

2) DFPN-6-D: CSPNet [22], as a backbone network used in YOLO-V4 [29], shows good performance, and the processing speed is greatly improved. Therefore, when designing the backbone network, we integrated the design ideas of the CSPNet network and then combined it with DFPN. Finally, the DFPN-6-D network for pose estimation is designed, as shown in Fig. 3. The SPP block is added to the backbone network, as it can significantly increase the receptive field; at the same time, it can separate out the most significant context

features and hardly reduce the operation speed of the network. The existing approaches are not effective when estimating the pose of small target objects, so we added the focus layer and multiscale detection layer to the network to improve the detection effect of small objects. The focus layer concatenates the higher resolution features with the low-resolution features by stacking adjacent features into different channels instead of spatial locations. The focus layer can turn the $640 \times 480 \times 3$ feature map into a $320 \times 240 \times 12$ feature map, which can be concatenated with the original features. Compared with the original RGB three-channel mode, the spliced image has 12 channels. Then, the new image is subjected to a convolution operation, and finally, a double downsampled feature map without information loss is obtained. The eight vertices and center points of the bounding box of the 3-D model are selected as keypoints to establish a 3-D–2-D correspondence with their projection points, similar to BB8 [16] and YOLO-6-D [17]. The keypoint that we selected can be used for any rigid 3-D object with any shape. Our model takes an RGB image as input, processed by the neural network shown in Fig. 3, and finally obtains an $S \times S \times (9 \times 2 + 1 + C)$ tensor, where $S \times S$ means that the input image is divided into $S \times S$ grids. Each grid contains the coordinates of nine key points, the class probabilities of the object and confidence. When training our network, we need to know only the eight vertices and center points of the 3-D bounding box of the object, not a detailed mesh or an associated texture map. The training process of our network is similar to that of YOLO and YOLO-6-D.

B. FCIoU, a New Confidence Function

As in YOLO-V2 [13], YOLO-V4 [29], and faster-RCNN [25], an excellent network can not only accurately predict the location of the object but also predict the confidence of the object. In the field of 2-D detection, the calculation method of confidence is mainly based on calculating the intersection over union (IoU) score. The higher the value of IoU is, the greater the possibility that the area contains objects. In the case of 6-D pose estimation, the objects are located in 3-D space. If we wish to calculate the equivalent IoU score between them, we need to calculate the overlap volume between the two 3-D bounding boxes. This is very time-consuming and difficult.

In YOLO-6-D, a new confidence calculation method is used to project the predicted 3-D bounding box vertex onto a 2-D plane and then calculate the distance between it and the ground-truth 2-D projection point. The average offset of nine projection points is calculated as a measure of confidence. The problem is that, although the offset in the 2-D plane may be small, the offset is large when mapped to the 3-D space, resulting in inaccurate confidence.

In this article, a new confidence algorithm is proposed to project the 3-D bounding box of the target object into the front view, end view, and vertical view, calculate CIoU, and then use the weighted average as the final confidence. The new confidence algorithm is named FCIoU, as shown in Fig. 4. The algorithm predicts the projection point of the target in 2-D space and calculates the weighted value of the CIoU between the three predicted faces and the ground truth as a measure of confidence. The advantage of this is that it takes full advantage of the relationship between the 3-D information and each projection point. IoU [40] is defined as

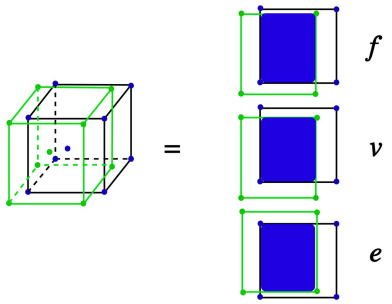


Fig. 4. New confidence algorithm that projects the 3-D bounding box of the target object into the front view, end view, and vertical view and calculates the CIoU.

$$\text{IoU} = \frac{\left|B \bigcap B^{gt}\right|}{\left|B \bigcup B^{gt}\right|} \tag{1}$$

where $B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$ is the ground truth and $B = (x, y, w, h)$ is the predicted box. The loss function of CIoU [40] is defined as

$$L_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{2}$$

where b and b^{gt} denote the central points of B and B^{gt}, $\rho(\cdot)$ is the Euclidean distance, c is the diagonal length of the smallest enclosing box covering the two boxes, a is a positive tradeoff parameter, and v measures the consistency of the aspect ratio. They are defined as

$$v = \frac{4}{\pi^2}\left(\arctan\frac{\omega^{gt}}{h^{gt}} - \arctan\frac{\omega}{h}\right)^2 \tag{3}$$

$$\alpha = \frac{v}{(1 - \text{IoU}) + v}. \tag{4}$$

Then, the loss function of FCIoU is defined as

$$L_{\text{FCIoU}} = a \cdot L_{\text{CIoU}-f} + b \cdot L_{\text{CIoU}-v} + c \cdot L_{\text{CIoU}-e} \tag{5}$$

where a, b, and c are the weights of each projection surface, $a = b = c = (1/3)$, $L_{\text{CIoU}-f}$ indicates that the L_{CIoU} algorithm is used in the front view in Fig. 4, and $L_{\text{CIoU}-v}$ and $L_{\text{CIoU}-e}$ indicate that the L_{CIoU} algorithm is used in the vertical view and the end view, respectively.

The output of our network is an $S \times S \times D$ tensor, the image is divided into $S \times S$ grids, and each cell has a D-dimensional vector, $D = 9 \times 2 + C + 1$, as we have $9(x_i, y_i)$ control points, C class probabilities, and one confidence value.

Finally, we minimize the following loss function to train our complete network:

$$\text{Loss} = \lambda_{\text{pt}} \cdot L_{\text{pt}} + \lambda_{\text{FCIoU}} \cdot L_{\text{FCIoU}} + \lambda_{\text{id}} \cdot L_{\text{id}} \tag{6}$$

where L_{pt}, L_{FCIoU}, and L_{id} denote the coordinate, confidence, and the classification loss, respectively.

C. 6-D Object Pose Estimation

The DFPN-6-D network is very efficient and needs to call the network only once to estimate the pose of the 6-D object. At test time, we multiply the object's category probability by the score calculated by the confidence function to obtain the object's confidence score. When the network is being evaluated, each segmented cell generates predictions, and the low-confidence prediction cells are pruned using a threshold. To obtain a more accurate and robust pose estimation, we average the confidence scores of nine grids in the 3×3 neighborhood of the grid with the highest confidence and use the average as the final confidence score of the grid.

At runtime, the network predicts the projection positions of the vertices and center points of the 3-D bounding box. When the network predicts the 2-D projection point, we use the PnP algorithm to estimate the pose of the object and calculate the 3-D rotation R and 3-D translation T of the object.

IV. 6-D Augmentation

A. Linemod Dataset Augmentation

Linemod and Occluded-Linemod are the standard benchmark datasets for 6-D pose estimation, and the amount of annotation data used in this work is very limited. The Linemod dataset consists of 13 sequences, each of which contains approximately 1200 annotated examples, and Occluded-Linemod is a subset of Linemod in which the amount of data is equally small. Due to insufficient dataset samples, large neural networks cannot readily converge during the training process. Therefore, data enhancement on a small dataset can help the network learn more features. In the field of 2-D images, the effect of data augmentation can be achieved by rotation and scaling, but these methods cannot be used directly in the field of 6-D pose estimation. Some image transformations cause image mismatch with the ground-truth 6-D pose, resulting in bad data. To solve this problem, we propose a 6-D augmentation method, which can rotate and scale the image and ensure that the changed image and the ground-truth 6-D pose still match.

We take the center point of the image as the origin of the coordinate system and establish the coordinate system shown in Fig. 5. The process of 6-D pose estimation transforms the object coordinate system to the camera coordinate system. When the image is rotated by θ around the z-axis, the 3-D rotation R and translation T of the object must also be rotated by θ around the z-axis. Similarly, when the image is rotated around the x-axis or the y-axis, the 3-D rotation R and translation T of the object must be rotated at the same angle. We can also rotate the image around the z-axis first and then around the x-axis as long as the 6-D pose of the object is rotated in the same way

$$\Delta r_z = (0, 0, \theta)^T, \theta \in (-\pi, \pi) \tag{7}$$

$$\Delta r_x = (\theta, 0, 0)^T, \theta \in \left(-\frac{\pi}{6}, \frac{\pi}{6}\right) \tag{8}$$

$$\Delta r_y = (0, \theta, 0)^T, \theta \in \left(-\frac{\pi}{6}, \frac{\pi}{6}\right). \tag{9}$$

With the rotation matrix ΔR obtained from Δr_x, Δr_y, and Δr_z, the augmented rotation matrix $R_{z-\text{avg}}$ and translation t_{avg} can be computed with the following equations:

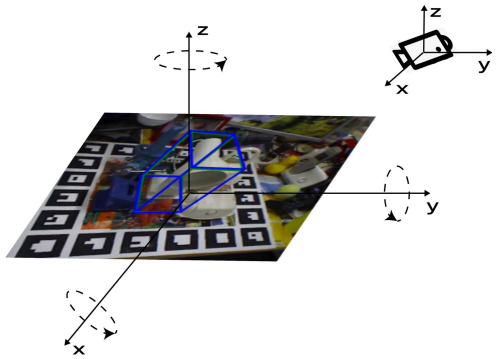


Fig. 5. Image can be rotated around the x-axis, the y-axis, and the z-axis to simulate images taken by the camera from different angles.

$$R_{\text{avg}} = \Delta R \cdot R \tag{10}$$

$$t_{\text{avg}} = \Delta R \cdot t. \tag{11}$$

Simulating the distance of the camera shooting object by zooming the image is also a method of data augmentation. We need only to adjust t_z in the translation matrix $t_{\text{avg}} = \left(t_x, t_y, t_z\right)^T$ to achieve the effect of scaling the image. The translation t_{avg} can be calculated as follows:

$$t_{\text{avg}} = \left(t_x, t_y, \frac{t_z}{f_{\text{scale}}}\right)^T. \tag{12}$$

In this work, the rotation range of the image along the z-axis is $(-\pi, \pi)$, the rotation range for the x-axis and y-axis is $[-(\pi/6), (\pi/6)]$, and the image scale range is $[0.8, 1.2]$.

B. Occluded Dataset Augmentation

In the task of 6-D pose estimation, the occlusion problem severely affects the detection accuracy of the CNNs. Many existing approaches have achieved good accuracy results without occlusion, but the accuracy is poor when the object is occluded. In the real world, occlusion problems can be found everywhere, so how to overcome the problem is an important challenge for 6-D pose estimation. To solve the occlusion problem, it is necessary not only to improve and optimize the CNN model but also to have a sufficient dataset. Here, we propose an occlusion data enhancement method for 6-D pose estimation, which we call 6-D occlusion augmentation. Since the amount of data in the existing occlusion dataset containing label information is very limited, the dataset cannot meet the training requirements of CNN. Therefore, we adopted a random erasure method [30] to enrich the existing occlusion dataset.

In Section VI-A, we enriched the Linemod dataset. Here, we use this dataset as the basis for extension to the Occluded dataset with random erasure. The random erasure method refers to randomly selecting a position in the picture, then randomly drawing a rectangular area of any size with this position as the center, and randomly assigning a new pixel value to all the pixel values in the selected area. In this manner, augmented images with various occlusion levels can be generated. Examples of the method are shown in Fig. 6.

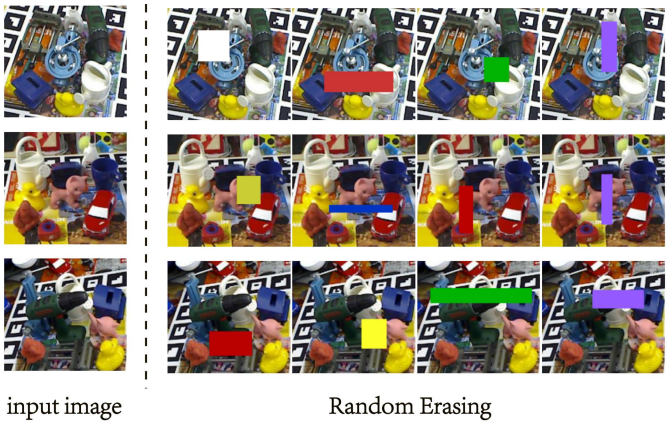


Fig. 6. Examples of random erasure. In CNN training, we randomly select a rectangular area in the image and replace all pixels in the area with random values. Images with various levels of occlusion are, thus, generated.

In the data enhancement process, N original pictures are input, the probability of each picture being randomly erased is p, and finally, only the objects in the $N \times p$ pictures are occluded to varying degrees. The advantage of this is that the CNN network robustness is improved. In the erasure process, a rectangular area is randomly selected in the target area, and the pixel value of the rectangular area is replaced with a random value. Assuming that the size of the input picture is $w \times h$ and the area of the picture is $S = w \times h$, we randomly generate an erased area I_s with an area of S_e, where the range of (S_e/S) is (s_1, s_2), the aspect ratio of the erased rectangular area is r_e, and the range is (r_1, r_2). The dimensions of the erased area I_s are $w_e = ((S_e/r_e))^{1/2}$ and $h_e = (S_e \times r_e)^{1/2}$. Then, we randomly select a point $P = (x_e, y_e)$ in the image. If $x_e + w_e \leq w$ and $y_e + h_e \leq h$, we set the region $I_s = (x_e, y_e, x_e+w_e, y_e+h_e)$ as the selected rectangular region. Otherwise, we repeat the above steps until the appropriate I_s value is selected. Finally, we select all the pixel values in the area to be erased and replace it with a random number; the range of the random number is $[0, 255]$. Algorithm 1 shows the detailed process.

Algorithm 1 Random Erasing Procedure

Require:
Input image I;
Image size w and h:
Area of image S;
Erasing probability p;
Erasing area ratio range s_w and s_h;
Erasing aspect ratio range r_1 and r_2;
Ensure: Erased image I'
$p_1 \leftarrow Rand(0, 1)$
if $p_1 > p$ **then**
$I' \leftarrow I$;
return I'.
else
$S_e \leftarrow Rand(s_w, s_h) \times S$;
$r_e \leftarrow Rand(r_1, r_2)$;
$H_e \leftarrow \sqrt{S_e \times r_e}$, $W_e \leftarrow \sqrt{\frac{S_e}{r_e}}$;
$x_e \leftarrow Rand(0, w)$, $y_e \leftarrow Rand(0, h)$;
if $x_e + W_e \leq W$ and $y_e + H_e \leq H$ **then**
$I_s \leftarrow (x_e, y_e, x_e + W_e, y_e + H_e)$;
$I(I_s) \leftarrow Rand(0, 255)$;
$I' \leftarrow I$;
return I'.
end if
end if

V. Experimental Evaluation

In this section, we evaluate our approach on the Linemod dataset [9], the Occluded-Linemod dataset [2], the MULT-I dataset [18], the BIN-P dataset [42], and the T-LESS dataset [12]. We compare the proposed method to other state-of-the-art methods for 6-D pose estimation. In this section, we introduce these popular benchmark datasets.

A. Linemod Dataset [9]

The Linemod dataset is a popular benchmark dataset for the 6-D pose estimation of objects. Most algorithms are evaluated on this dataset, so it is credible to use this dataset to verify our algorithm. The Linemod dataset consists of 13 different objects that are placed in different cluttered scenes. There are approximately 1200 images for each object. Although each image contains multiple objects, only one object has 6-D pose annotation information. In the division of the training dataset and the test dataset, we use the same division scheme as other methods [16], [17], [23] to ensure that subsequent experimental comparisons are fair.

B. Occluded-Linemod Dataset [2]

The Occluded dataset is a subset of the Linemod dataset. In this subset, each picture contains multiple objects, and each object has detailed annotation information. These objects are stacked and occluded by each other, and it is very difficult to estimate the pose of these objects. Therefore, this dataset can evaluate the performance of the algorithm in the case of multiobject stacking and occlusion. Similarly, we use the same division scheme as the Linemod dataset to ensure that subsequent experimental comparisons are fair.

C. MULT-I Dataset [18] and BIN-P Dataset [42]

The MULT-I dataset is a challenging dataset for multiple-instance detection containing heavy 2-D and 3-D clutter, as well as foreground occlusions. The dataset consists of six sequences where each sequence requires the detection and pose estimation of multiple instances of the same object in clutter and with different levels of mild occlusion. BIN-P dataset adds more occlusion and stacking on the basis of the MULT-I dataset. We can verify the robustness of our proposed algorithm in the case of occlusion and clutter on both datasets.

D. T-LESS Dataset [12]

The T-LESS dataset contains 30 industrial-related objects. The characteristic of the dataset is that there is no significant texture or discriminative color. The objects exhibit symmetries and mutual similarities in shape and/or size, and a few objects

are a composition of other objects. Our research field is robotic grasping in industrial environments, so this dataset is very suitable for verifying our algorithm.

E. Evaluation Metrics

We use four kinds of standard metrics to evaluate the accuracy of 6-D pose estimation: 1) the 2-D reprojection error; 2) ADD score [2], [14], [31]; and 3) F1-score and average recall (AR) [46]. When we use the 2-D reprojection error as a measure of 6-D pose estimation, we consider a pose estimate to be correct when the mean distance between the 2-D projections of the object's 3-D vertices using the estimate and the ground-truth pose is less than five pixels. When comparing 6-D poses using the ADD metric, ADD is defined as an average Euclidean distance between model vertices transformed with the predicted and ground-truth poses. If the average distance is less than 10% of the model's diameter, then the estimated object pose can be considered correct. The formula is defined as follows:

$$m = \underset{x \in M}{\text{avg}} \left\| (Rx + t) - \left(R'x + t'\right) \right\|_2 \tag{13}$$

where M represents a set of vertices of a particular model, R and t are the rotation and translation of the ground-truth transformation, and R' and t' correspond to the estimated rotation and translation, respectively. The ADD metric can be extended to solve symmetric objects, as shown in [9]

$$m = \underset{x_2 \in M}{\text{avg}} \min_{x_1 \in M} \left\| (Rx_1 + t) - \left(R'x_2 + t'\right) \right\|_2. \tag{14}$$

F1-Score is defined as follows:

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{15}$$

AR contains three error functions as follows:

$$\text{AR} = (\text{AR}_{\text{VSD}} + \text{AR}_{\text{MSSD}} + \text{AR}_{\text{MSPD}})/3. \tag{16}$$

The VSD, MSSD, and MSPD error functions represent visible surface discrepancy, maximum symmetry-aware surface distance, and maximum symmetry-aware projection distance, respectively. For details on the formula, please refer to the literature [46].

F. Training Procedure and Implementation Details

During training, DFPN-6-D needs to know only the center points and 3-D bounding box corners of the object, not a detailed 3-D model or texture map of the object. The 3-D bounding box of objects can be derived from other easy-to-obtain and approximate 3-D shape representations, so this method can effectively deal with textureless objects. We train the network to predict the projection position of the corner points of the 3-D bounding box, as well as the classification and confidence of the target object. The network predicts the offset of the 2-D coordinates of the object's center of mass relative to the coordinates (c_x, c_y) of the upper left corner of the associated grid unit and limits the offset to between 0 and 1. The predicted control point (b_x, b_y) is defined as

$$b_x = \sigma(t_x) + c_x \tag{17}$$

$$b_y = \sigma\left(t_y\right) + c_y \tag{18}$$

where $\sigma(\cdot)$ is chosen to be a 1-D sigmoid function in the case of the centroid and the identity function in the case of the eight corner points. The network must first detect the position of the center point of the object and then refine the positions of the other eight vertices.

The mean square error is used for coordinate and confidence losses, and the cross entropy is used for classification loss. To improve the stability of the model, we refer to the method in [17] and set λ_{FCIoU} to 0.1 to reduce the confidence loss of the cell that does not contain the object. For the cell containing the object, λ_{FCIoU} is set to 5.0, and λ_{pt} and λ_{id} are set to 1.

When multiple objects appear in the 3-D scene, some cells may contain multiple objects. When there are multiple objects in the same cell, to be able to predict the 6-D pose of multiple objects at the same time, and we allow up to three candidates per cell. For an input image of $416 \times 416 \times 3$, we set *a priori* boxes in each grid of the feature map of each scale, a total of $13 \times 13 \times 3 + 26 \times 26 \times 3 + 52 \times 52 \times 3 = 10647$ prediction candidates. Each predicted candidate has $(2 \times 9 + 1 + 13) = 32$ dimensions. This 32-D vector contains coordinates (18 values), confidence (1 value), and object category probability (for the Linemod dataset, there are 13 kinds of objects). Compared with YOLO-6-D using $13 \times 13 \times 5 = 845$ prediction candidates, our method attempts to predict that the number of candidates has increased by more than ten times, and it is performed at different resolutions, so the detection effect of small objects has a certain improvement. As in [17], we calculate the size of three anchor boxes by using the K-means approach in advance. During the training process, we designate the anchor frame closest to the size of the current object as the anchor frame responsible for this object, which is used to predict the 2-D coordinates of the object.

We use the Adam optimizer with an initial learning rate of $1e-4$ and a batch size of 4. During the training process, if the average point distance does not decrease within the last 15 evaluations on the test set, the learning rate decreases by a factor of 0.6. To prevent the training from being too slow, we set the minimum learning rate to $1e-7$. At the same time, to improve the network's ability to deal with scene clutter, we use the segmentation mask attached to the dataset to segment the training image and replace the background with random images in the Common Objects in Context (COCO) dataset [37].

G. Comparison With the State-of-the-Art Methods on the Linemod Dataset

To verify the effectiveness of our proposed 6-D pose estimation algorithm, we compare it with the most advanced 6-D pose estimation algorithm. Furthermore, to confirm that the DFPN network we proposed is more advantageous than the BiFPN network, we replace the DFPN network with the BiFPN network in the DFPN-6-D network that we designed and then compare it with DFPN-6-D. We use only RGB to estimate the 6-D pose of the object on the Linemod dataset, without reference to the depth information. We compare our approach with state-of-the-art methods.

The results in terms of 2-D reprojection error are shown in Table I, and the 6-D pose error is shown in Table II.

TABLE I

COMPARISON OF OUR APPROACH WITH STATE-OF-THE-ART ALGORITHMS ON LINEMOD IN TERMS OF THE 2-D REPROJECTION ERROR

	w/o refinement						w/ refinement	
Method	Brachmann [2]	BB8 [16]	YOLO-6D [17]	PVNet [23]	BiFPN-6D (Ours)	DFPN-6D (Ours)	BB8 [16]	Brachmann [2]
Ape	-	95.3	92.10	99.23	98.85	99.41	96.6	85.2
Benchvise	-	80.0	95.06	99.81	99.72	99.36	90.1	67.9
Cam	-	80.9	93.14	99.21	99.13	99.79	86.0	58.7
Can	-	84.1	97.44	99.90	98.89	98.78	91.2	70.8
Cat	-	97.0	97.41	99.30	97.98	99.75	98.8	84.2
Driller	-	74.1	79.41	96.92	96.20	98.92	80.9	73.9
Duck	-	81.2	94.65	98.02	98.91	98.95	92.2	73.1
Eggbox	-	97.9	90.33	99.34	97.28	99.40	91.0	83.1
Glue	-	89.0	96.53	98.45	98.22	98.28	92.3	74.2
Holepuncher	-	90.5	92.86	100.0	99.17	99.72	95.3	78.9
Iron	-	78.9	82.94	99.18	97.93	97.57	84.8	83.6
Lamp	-	74.4	76.87	98.27	98.57	98.89	75.8	64.0
Phone	-	77.6	86.07	99.42	98.87	98.85	85.3	60.6
Average	69.5	83.9	90.37	99.00	98.36	**99.05**	89.3	73.7

TABLE II

COMPARISON OF OUR APPROACH WITH STATE-OF-THE-ART ALGORITHMS ON LINEMOD IN TERMS OF THE ADD METRIC

	w/o refinement								w/ refinement		
Method	YOLO-6D [17]	Pix2Pose [32]	PVNet [23]	CDPN [33]	HybridPose [34]	EfficientPose [31]	BiFPN-6D (Ours)	DFPN-6D (Ours)	BB8 [16]	SSD-6D [14]	DPOD [24]
Ape	21.62	58.1	43.6	64.4	77.6	89.43	91.64	93.62	40.4	65	87.73
Benchvise	81.80	91.0	99.9	97.8	99.6	99.71	97.32	99.39	91.8	80	98.45
Cam	36.37	60.9	86.9	91.7	95.9	98.53	98.50	99.19	55.7	78	96.07
Can	68.8	84.4	95.5	95.9	93.6	99.70	98.78	98.82	64.1	86	99.71
Cat	41.82	65.0	79.3	83.8	93.5	96.21	97.41	98.77	62.6	70	94.71
Driller	63.51	76.3	96.4	96.2	97.2	99.50	98.02	98.86	74.4	73	98.80
Duck	27.23	43.8	52.6	66.8	87.0	89.20	91.42	95.25	44.3	66	86.29
Eggbox	69.58	96.8	99.2	99.7	99.6	100	97.68	99.93	57.8	100	99.91
Glue	80.02	79.4	95.7	99.6	98.7	100	98.95	99.07	41.2	100	96.82
Holepuncher	42.63	74.8	81.9	85.8	92.5	95.72	95.15	96.43	67.2	49	86.87
Iron	74.79	83.4	98.9	97.9	98.1	99.08	98.11	98.88	84.7	78	100.0
Lamp	71.11	82.0	99.3	97.9	96.9	100	98.79	99.56	76.5	73	96.84
Phone	47.74	45.0	92.4	90.8	98.3	98.46	97.49	98.19	54.0	79	94.69
Average	55.95	72.4	86.3	89.9	94.5	97.35	96.87	**98.06**	62.7	79	95.15

Our method outperforms all the other methods by a significant margin. Our approach achieves state-of-the-art pose estimation accuracy with the 2-D projection error and ADD metric without any refinement, and our method does not require complete 3-D computer aided design (CAD) model information or additional other information.

According to the projection error metric, from Table I, our method is 0.69% more accurate than BiFPN-6-D and better than PVNet. According to the ADD metric, from Table II, our method scores 2.91% higher than DPOD, 3.56% higher than HybridPose, 0.71% higher than EfficientPose, and 1.19% higher than BiFPN-6-D. Compared with other state-of-the-art methods, our proposed DFPN-6-D method achieves the best results on the Linemod dataset. Compared with BiFPN-6-D and DPFN-6-D, our method scores 1.19% higher than BiFPN-6-D. As indicated by the data in Tables I and II, our proposed DFPN is more effective than BiFPN. Fig. 7 shows the pose estimation result of our method on the Linemod dataset.

H. Comparison With the State-of-the-Art Methods on the Occluded-Linemod Dataset

We evaluate the algorithm proposed in this article on the Occluded-Linemod dataset, and the evaluation results are reported in terms of Mean Average Precision (mAP). Table III shows the mAP scores of our algorithm and the most advanced algorithm. The mAP score of our algorithm is 87.09%, which is 7.89% higher than that of HybridPose, 3.11% higher than that of EffectivePose, and 1.97% higher than that of BiFPN-6-D. Figs. 1 and 8 show the pose estimation result of our method on the Occluded-Linemod dataset. Our approach is effective for occluded objects. Through the figure and Table III, we can conclude that the 6-D pose estimation approach proposed in this article is the most advanced approach at present, confirming that the DFPN network proposed in this article is superior to the BiFPN network in terms of performance.

I. Comparison With the State-of-the-Art Methods on the MULT-I [18] Dataset

To prove the effectiveness of our approach, we evaluated our method on the MULT-I dataset presented in [31], which contains multiple objects of one category per test image, with much clutter and some cases of occlusion. Table IV shows the results in the form of F1-Score for each of the six objects. The F1-Score of our algorithm is 88.6%, which is 8.3% higher than that of Kehl *et al.* [42] and 25.3% higher than that of

Fig. 7. Pose estimation results of our method on the Linemod dataset. Green 3-D bounding boxes visualize ground-truth poses, while our estimated poses are represented by blue. The first row in the figure is the detection effect on small target objects. It can be concluded that our algorithm is effective for the pose estimation of small target objects; the second row in the figure is the detection effect on textureless objects, which can be concluded that our method is effective for the pose estimation of textureless objects; and the third row is the detection effect of ordinary objects.

TABLE III

DETECTION PERFORMANCE FOR MULTIPLE OBJECTS: COMPARISON OF THE STATE-OF-THE-ART mAP SCORES ON THE OCCLUDED-LINEMOD DATASET

Object	YOLO-6D [17]	PoseCNN [35]	Oberweger [36]	Hu [11]	Pix2Pose [32]	PVNet [23]	DPOD [24]	HybridPose [34]	EfficientPose [31]	BiFPN-6D (Ours)	DFPN-6D (Ours)
ape	2.48	9.6	12.1	17.6	22.0	15.8	-	53.3	59.39	64.78	68.17
can	17.48	45.2	39.9	53.9	44.7	63.3	-	86.5	93.27	92.42	93.24
cat	0.67	0.93	8.2	3.3	22.7	16.7	-	73.4	79.78	83.54	84.63
driller	7.66	41.4	45.2	62.4	44.7	65.7	-	92.8	97.77	97.11	98.34
duck	1.14	19.6	17.2	19.2	15.0	25.2	-	62.8	72.71	75.36	77.54
eggbox	-	22.0	22.1	25.9	25.2	50.2	-	95.3	96.18	95.32	95.59
glue	10.08	38.5	35.8	39.6	32.4	49.6	-	92.5	90.80	91.03	92.67
holepuncher	5.45	22.1	36.0	21.3	49.5	39.7	-	76.7	81.95	81.40	86.55
average	6.42	24.9	27.0	27.0	32.0	40.8	47.3	79.2	83.98	85.12	**87.09**

Fig. 8. Pose estimation results of our method on the Occluded-Linemod dataset. Green 3-D bounding boxes visualize ground-truth poses, while our estimated poses are represented by the other colors. Our method is effective for occluded objects.

Alykhan *et al.* [18]. Fig. 9 shows the pose estimation results of our method on the MULT-I dataset. From Table IV and Fig. 9, we can see that our method significantly outperforms the state-of-the-art methods on this dataset.

J. Comparison With the State-of-the-Art Methods on the BIN-P Dataset [42]

In order to prove the effectiveness of our method, we evaluated it on the BIN-P dataset provided in [42]. The BIN-P

TABLE IV

RESULTS ON THE DATASET OF MULT-I [18]. EVALUATE OUR APPROACH USING THE F1 METRIC

Object	LINEMOD [9]	Drost [41]	Tejan [18]	Douman [42]	DFPN-6D (Ours)
	F1 Score				
Coffee Cup	0.819	0.867	0.877	0.932	**0.954**
Shampoo	0.625	0.651	0.759	0.735	**0.873**
Joystick	0.454	0.277	0.534	**0.924**	0.918
Camera	0.422	0.407	0.372	0.903	**0.925**
Juice Carton	0.292	0.604	0.870	0.819	**0.851**
Milk	0.176	0.259	0.385	0.510	**0.797**
Average	0.498	0.511	0.633	0.803	**0.886**

TABLE V

RESULTS ON THE DATASET OF BIN-P [42]. EVALUATE OUR APPROACH USING THE AR

Method	DPOD [24]	Sunder [44]	Pix2Pose [32]	Leap [45]	CDPN [33]	DFPN-6D (Ours)
Average	13.0	21.7	22.6	34.2	47.3	**51.7**

Fig. 9. Pose estimation results of our method on the MULT-I dataset [18].

Fig. 10. Pose estimation results of our method on the BIN-P dataset [42].

dataset is similar to the MULT-I dataset, but the occlusion of objects in the BIN-P dataset is more serious. From Table V, we can learn that the AR [46] score of our algorithm is 51.7%, which is 17.4% higher than that of Leap [45] and 4.4% higher than that of coordinates-based disentangled pose network (CDPN) [33]. Fig. 10 shows the pose estimation results of our method on the BIN-P dataset. From Table V and Fig. 10, we can see that our method significantly outperforms the state-of-the-art methods.

K. Comparison With the State-of-the-Art Methods on the T-LESS Dataset [12]

Finally, we add an industrial dataset T-LESS [12] to verify our algorithm. The evaluation results are reported in terms of AR [46]. Table VI shows the recall scores of our algorithm and the most advanced algorithm. The AR score of our algorithm is 47.6%, which is 7.3% higher than that of Leap [45], 16.6% higher than that of Sunder [44], and 18.1% higher than that of Pix2Pose [32]. Fig. 11 shows the pose estimation result of our method on the T-LESS dataset. As shown in Fig. 11 and Table VI, we can see that our method significantly outperforms the state-of-the-art methods; meanwhile, it proves that our method is very suitable for the industrial measurement field.

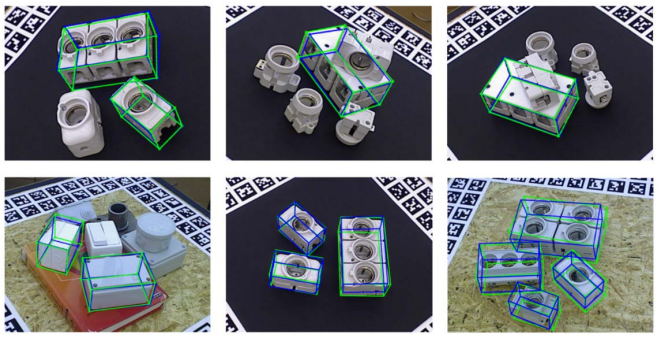


Fig. 11. Pose estimation results of our method on the T-LESS dataset. Green 3-D bounding boxes visualize ground-truth poses, while our estimated poses are represented by the other colors.

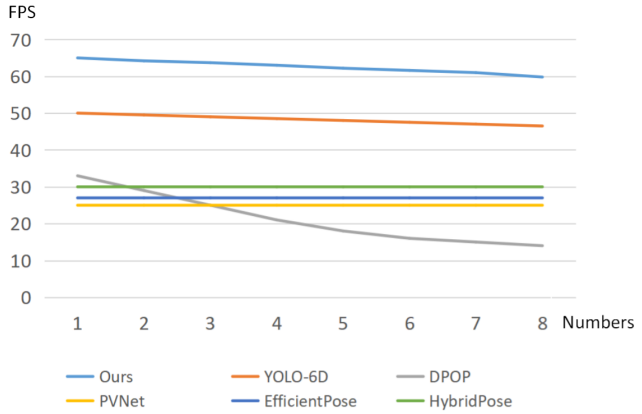


Fig. 12. Comparison result of the state-of-the-art approaches running speed in the case of different numbers of objects in the image. Our approach is the fastest. When there is only one object in the image, our approach can reach 65 frames/s, and when the image contains eight objects, our speed can reach 59 frames/s. Timing is measured on a Titan X (Pascal) GPU.

TABLE VI

RESULTS ON THE DATASET OF T-LESS [12]. EVALUATE OUR APPROACH USING THE AR

Method	DPOD [24]	Sunder [44]	Pix2Pose [32]	Leap [45]	CDPN [33]	DFPN-6D (Ours)
Average	8.1	30.4	34.3	40.3	49.0	**52.8**

L. Runtime Analysis

In Table VII, we show the efficiency of the proposed method for 6-D object pose estimation in comparison to the state-of-the-art approaches. We use the same graphics processing unit (GPU) compared with other methods; our method is the fastest, more than twice as fast as [31], [34] in speed and 15 frames/s faster than [17]. When detecting multiple targets, our algorithm does not take extra time, but [24] needs to spend an extra 5 ms on each object.

Fig. 13. (a) Platform consists of a UR5 robot arm, a Robotiq Gripper, and an RGB camera. (b) 6-D pose estimation of objects by our method. (c) Processes of moving the gripper (row 1) and grasping of objects (row 2, 3).

TABLE VII
COMPARISON OF COMPUTING SPEED WITH THE STATE-OF-THE-ART METHODS. WE CALCULATE THE AVERAGE TIME ON MULTIPLE DATASETS

Method	Speed	Refine	Param	FLOPs
Branchmann[2]	2 fps	100ms/object	-	-
Kehl[44]	2 fps	-	-	-
BB8[16]	3 fps	21ms/object	-	-
SSD-6D[14]	10 fps	24ms/object	271M	863B
Pix2Pose[32]	10 fps	-	-	-
Leap[46]	10 fps	-	-	-
CDPN[33]	20 fps	-	-	-
PVNet[23]	25 fps	-	-	-
EfficientPose[31]	27 fps	-	52M	325B
DPOD [24]	33 fps	-	-	-
HybridPose[34]	30 fps	-	126M	268B
Sunder[45]	37 fps	-	117M	231B
YOLO-6D[17]	50 fps	-	63M	157B
BiFPN-6D(Ours)	65 fps	-	53M	108B
DFPN-6D(Ours)	65 fps	-	53M	112B

Our method can achieve the real-time and accuracy of 6-D pose estimation. During the operation of the CNN, we do not need a complete 3-D model to refine and our pose estimation, so our method can efficiently process multiple objects. Even if the objects are occluded, the accuracy of detection can still be guaranteed. As shown in Fig. 12, when detecting multiple objects, the only computational cost increase of our algorithm is associated with the PnP algorithm. For each additional object, the calculation time increases by only 0.2 ms, while other methods have linear runtime growth.

M. Robotic Experiments

In order to verify the effectiveness of our proposed method, we set up a robotic arm grasping platform, as shown in Fig. 13(a). Our platform consists of a UR5 robot arm, an RGB camera, and a Robotiq Gripper. In addition, we also collected some industrial parts, such as screws and nuts for experiments. Fig. 13(b) shows the result of the 6-D pose estimation of the object by our proposed method. Fig. 13(c) shows the movement process of the robot arm grasping the object.

TABLE VIII
RESULTS ON ROBOTIC GRASPING EXPERIMENTS

Object	Prediction correct (%)	Reaching success (%)	Grasping/Holding success (%)
Nut	98	98	97
T-screw	100	100	100
L-machine screw	98	97	97
S-machine screw	97	96	94
Pipe joint	98	97	97
Screw	99	96	95
Average	98.3	97.3	96.6

TABLE IX
ABLATION STUDY TO EVALUATE THE INFLUENCE OF OUR PROPOSED METHOD

Backbone	Neck	FCIoU	ADD(%)	FPS
CSPNet	BiFPN	without	95.59	65
CSPNet	DFPN	without	96.83	65
CSPNet	BiFPN	with	96.87	65
CSPNet	DFPN	with	**98.06**	**65**

A successful grasp depends on three key steps: prediction, reaching, and the final grasp/hold, so we show the accuracies at each step in Table VIII. From Fig. 13 and Table VIII, we can conclude that our proposed method can be well applied to robotic arm grasping.

N. Ablation Study

To demonstrate the importance of our proposed method, we evaluated on Linemod dataset, and the results are shown in the Table IX. By comparing BiFPN-without and BiFPN-with, DFPN-without and DFPN-with, we can see that using the FCIoU loss function can improve the accuracy of the network. When the loss function is the same, by comparing BiFPN and DFPN, we can see that the performance of the DFPN network is better than the BiFPN network.

VI. CONCLUSION

We propose DFPN, which efficiently performs multiscale feature fusion and has significant robustness against occlusions, textureless features, and scene confusion. We combine

DFPN and CSPNet to form a DFPN-6-D network for 6-D pose estimation. DFPN-6-D can effectively detect small target objects and estimate their 6-D pose. DFPN-6-D establishes a 2-D–3-D correspondence by predicting the projection position of the 3-D bounding box angle of the target object in the 2-D space and then calculates the 6-D pose of the object by the PnP algorithm. In the field of using only 2-D images to estimate the 6-D pose, most methods consider only plane information when designing the loss function, but the 6-D pose estimation of the object needs to consider 3-D information. To solve this problem, we propose a new confidence calculation method, FCIoU, which can effectively extract and use potential spatial information from 2-D images. When we train the model, the process of training the model requires a large amount of label data, and the benchmark dataset Linemod contains insufficient labeled data, which results in the existing amount of data being unable to fully train the model. For this situation, we propose a method for data enhancement in the field of 6-D pose estimation, which can effectively enrich the small dataset. To improve the accuracy of the model in detecting occluded objects, we introduce a random erasure method to expand the Occluded-Linemod dataset. Our approach achieves a new state-of-the-art result on the Linemod, Occluded-Linemod, MULT-I, BIN-P, and T-LESS datasets while still running end-to-end at over 59 frames/s. In addition, we demonstrated that a UR5 robot can use our proposed approach to grasp and manipulate objects.

REFERENCES

[1] J.-P. Mercier, C. Mitash, P. Giguere, and A. Boularias, "Learning object localization and 6D pose estimation from simulation and weakly labeled real images," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3500–3506.

[2] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3D object coordinates," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 536–551.

[3] T. Cavallari *et al.*, "Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2465–2477, May 2019.

[4] X. Lu, J. Ji, Z. Xing, and Q. Miao, "Attention and feature fusion SSD for remote sensing object detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[5] Y. Cai *et al.*, "YOLOv4-5D: An effective and efficient object detector for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[6] K. Park, T. Patten, J. Prankl, and M. Vincze, "Multi-task template matching for object detection, segmentation and pose estimation using depth images," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7207–7213.

[7] P. Liu, Q. Zhang, J. Zhang, F. Wang, and J. Cheng, "MFPN-6D real-time one-stage pose estimation of objects on RGB images," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 12939–12945.

[8] G. He, X. Yuan, Y. Zhuang, and H. Hu, "An integrated GNSS/LiDAR-SLAM pose estimation framework for large-Scale map building in partially GNSS-denied environments," *IEEE Trans. Instrum. Meas.*, vol. 70, no. 9, pp. 1–9, Sep. 2020.

[9] S. Hinterstoisser *et al.*, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Proc. IEEE Asian Conf. Comput. Vis. (ACCV)*, Nov. 2012, pp. 548–562.

[10] L. Porzi, A. Penate-Sanchez, E. Ricci, and F. Moreno-Noguer, "Depth-aware convolutional neural networks for accurate 3D pose estimation in RGB-D images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5777–5783.

[11] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3385–3394.

[12] T. Hodan, P. Haluza, S. Obdrzalek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2017, pp. 880–888.

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

[14] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1530–1538.

[15] E. Munoz, Y. Konishi, C. Beltran, V. Murino, and A. D. Bue, "Fast 6D pose from a single RGB image using cascaded forests templates," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4062–4069.

[16] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 880–888.

[17] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 292–301.

[18] T. Alykhan, K. Rigas, D. Andreas, T. Danhang, and K. Tae-Kyun, "Latent-class Hough forests for 3D object detection and pose estimation," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2014, pp. 462–477.

[19] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2938–2946.

[20] D. Esslinger *et al.*, "Accurate optoacoustic and inertial 3-D pose tracking of moving objects with particle filtering," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 3, pp. 893–906, Mar. 2020.

[21] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 21–37.

[22] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 390–391.

[23] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4561–4570.

[24] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1941–1950.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2016.

[26] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 699–715.

[27] B. Cheng, W. Wu, D. Tao, S. Mei, T. Mao, and J. Cheng, "Random cropping ensemble neural network for image classification in a robotic arm grasping system," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 6795–6806, Feb. 2020.

[28] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[29] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Apr. 2020, pp. 1089–1106.

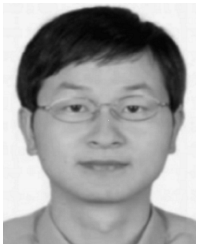
[30] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. IEEE AAAI Conf. Artif. Intell.*, vol. 34, no. 7, May 2020, pp. 13001–13008.

[31] Y. Bukschat and M. Vetter, "EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach," 2020, *arXiv:2011.04307*. [Online]. Available: http://arxiv.org/abs/2011.04307

[32] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7668–7677.

[33] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7678–7687.

[34] C. Song, J. Song, and Q. Huang, "HybridPose: 6D object pose estimation under hybrid representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 431–440.
[35] X. Yu, S. Tanner, N. Venkatraman, and F. Dieter, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," 2017, *arXiv:1711.00199*. [Online]. Available: https://arxiv.org/abs/1711.00199
[36] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3D object pose estimation," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Apr. 2018, pp. 125–141.
[37] T.-Y. Lin, M. Maire, and S. Belongie, "Microsoft COCO: Common objects in context," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Apr. 2014, pp. 740–755.
[38] T.-Y. Lin, P. Dollar, 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.
[39] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8759–8768.
[40] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 12993–13000.
[41] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010, pp. 998–1005.
[42] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim, "Recovering 6D object pose and predicting next-best-view in the crowd," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3583–3592.
[43] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local RGB-D patches for 3D object detection and 6D pose estimation," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 205–220.
[44] M. Sundermeyer *et al.*, "Multi-path learning for object pose estimation across domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13916–13925.
[45] J. Liu *et al.*, "Leaping from 2D detection to efficient 6DoF object pose estimation," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 707–714.
[46] T. Hodan *et al.*, "BOP challenge 2020 on 6D object localization," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 577–594.
[47] P. Liu, H. Zhao, P. Liu, and F. Cao, "RTM3D real time monocular 3D detection from object keypoints for autonomousm driving," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, Dec. 2020, pp. 644–660.
[48] L. Wang, X. Fan, J. Chen, J. Cheng, J. Tan, and X. Ma, "3D object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities," *Sustain. Cities Soc.*, vol. 54, Mar. 2020, Art. no. 102002.



Jun Cheng (Member, IEEE) received the B.Eng. and M.Eng. degrees from the University of Science and Technology of China, Hefei, China, in 1999 and 2002, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2006.

He is currently a Professor with the CAS Key Laboratory of Human-Machine Intelligence Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, and the Director of the Laboratory for Human Machine Control, Shenzhen. His current research interests include computer vision, robotics, and machine intelligence and control.

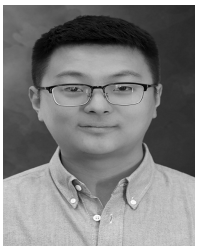
Penglei Liu received the M.E. degree in electrical engineering from Xiangtan University, Xiangtan, China, in 2018. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, Shenzhen Institutes of Advanced Technology, Shenzhen, China.

His current research interests include robots, neural network applications, and machine learning.

Qieshi Zhang (Member, IEEE) received the Ph.D. degree from Waseda University, Tokyo, Japan, in 2014.

From 2010 to 2012, he was a Research Fellow with Japan Society for the Promotion of Science (JSPS), Tokyo. From 2012 to 2019, he was a Research Assistant, a Research Associate, and an Adjunct Researcher with the Information, Production and Systems Research Center (IPSRC), Waseda University. He is currently an Associate Professor with Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Sciences, Shenzhen, China. He has authored or coauthored over 70 scientific articles in international journals and conferences. His current research interests are artificial intelligence, unmanned drive, robot, image processing, computer vision, and machine learning.

Dr. Zhang serves as the technical/program committee member for over 50 conferences and over 100 times.



Hui Ma received the M.E. degree in software engineering from the University of Science and Technology of China, Hefei, China, in 2019. He is currently pursuing the Ph.D. degree with Beijing Normal University-Hong Kong Baptist University United International College (BNU-HKBU), Zhuhai, China.

His current research interests include light-neural network design, embedded systems, 3-D reconstruction, and camera calibration.

Fei Wang is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, Shenzhen Institutes of Advanced Technology, Shenzhen, China.

His current research interests include computer vision, structure from motion, robotics, and deep learning.

Jin Zhang received the Ph.D. degree in computer science and engineering from the University of New South Wales, Sydney, NSW, Australia, in 2017.

He is an Associate Researcher with Shenzhen University, Shenzhen, China. He was a Post-Doctoral Researcher with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen. His research interests include Wi-Fi human sensing, deep learning, the Internet of Things, and computer networks.