2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

MFPN-6D : Real-time One-stage Pose Estimation of Objects on RGB Images

Penglei Liu[1,2,3] , Qieshi Zhang[1,2,3] , Jin Zhang[4], Fei Wang[1,2,3], Jun Cheng[1,2,3]*

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561878

***Abstract*— 6D pose estimation of objects is an important part of robot grasping. The latest research trend on 6D pose estimation is to train a deep neural network to directly predict the 2D projection position of the 3D key points from the image, establish the corresponding relationship, and finally use Pespective-n-Point (PnP) algorithm performs pose estimation. The current challenge of pose estimation is that when the object texture-less, occluded and scene clutter, the detection accuracy will be reduced, and most of the existing algorithm models are large and cannot take the real-time requirements. In this paper, we introduce a Multi-directional Feature Pyramid Network, MFPN, which can efficiently integrate and utilize features. We combined the Cross Stage Partial Network (CSPNet) with MFPN to design a new network for 6D pose estimation, MFPN-6D. At the same time, we propose a new confidence calculation method for object pose estimation, which can fully consider spatial information and plane information. At last, we tested our method on the LINEMOD and Occluded-LINEMOD datasets. The experimental results demonstrate that our algorithm is robust to textureless materials and occlusion, while running more efficiently compared to other methods.**

I. INTRODUCTION

Estimating the 6D pose of an object is an important task for robots as it enables robots to grasp and manipulate objects in the real word [1] [7] [8] [27] [30]. Pose estimation from images is also an important research topic in the field of computer vision. 6D object pose estimation is to calculate the transformation from the world coordinate system of the object to the camera coordinate system [2]. However, Objects have different 3D shapes and surface textures, at the same time there are also occlusions between objects, and clutter in the scene, the problem of 6D object pose estimation is challenging. The existing 6D pose estimation methods are mainly divided into two types: based on depth information or based on RGB information. Although the current methods for pose estimation using RGB-D cameras are robust, depth cameras are only suitable for indoor scenes and power hungry [3] [4] [5] [10]. On the contrary, RGB cameras are suitable for a wider range of scenes and power saving [11] [16] [17] [19] [26]. Therefore, in this work, we estimate the object pose from a single RGB image.

* Corresponding Author
[1]CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[2]Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Beijing, China
[3]The Chinese University of Hong Kong, Hong Kong, China
[4]The College of Computer Science and Software Engineering, Shenzhen University, China
{pl.liu, qs.zhang, fei.wang, jun.cheng}@siat.ac.cn, jin.zhang@szu.edu.cn

There are many Template matching [6], keypoint and edge-based methods [7] that are effective for textured objects. However, they rely on textures heavily and cannot handle the texture-less case [9] [12]. To solve this problem, methods based on deep learning have recently been used in pose estimation. Such as BB8 [16], PVNet [23] and Segmentation-driven [11], they trained deep neural networks to predict the 2D-3D correspondences and further solve the pose by a Pespective-n-Point (PnP) algorithm. Although they have achieved good performance, these methods either require a post-processing stage or need to use the RANdom SAmple Consensus (RANSAC) algorithm, and its require a lot of time. Some algorithms have achieved good results in terms of speed, such as YOLO-6D [17] which a single-shot deep CNN that takes image as input and directly estimate the object pose without any post-processing. As it is based on the YOLO-V2 network [13], although it inherits the advantages of YOLO-V2's speed, it is not effective in estimating small objects and objects with occlusion.

In this paper, we propose Multi-directional Feature Pyramid Network, MFPN, which can efficiently and fully integrate features. BiFPN [28] is one of the most advanced feature networks. It added a forward residual structure to PANet so that features can be better utilized in forward propagation, but it does not take into account the propagation utilization of features in the vertical direction. We added a vertical residual structure to BiFPN, so that features can be fully utilized not only in forward propagation, but also in vertical propagation. Our MFPN can effectively deal with the problem of insufficient features for textured-less and occluded objects.

In the object detection area, researchers mostly employ the backbone network to extract features from raw images. CSP-Net [22] is current state-of-the-art network, which optimizes the process of gradient back propagation, while maintain network size and computation cost. Herein, we propose a novel object pose detection architecture, MFPN-6D, that combines MFPN mentioned above and the CSPNet to efficiently predict the object pose. As such the MFPN-6D is characterized by the lightweight model size and capabilities of detecting occluded or texture-less objects. Our MFPN-6D is suitable for real-time pose estimation.

Most of existing methods calculate the offset between projection points and the ground truth as a measure of confidence, but the problem is that the offset in the 2D plane may be small, but in the 3D space the offset is very large, resulting in inaccurate confidence. To solve this problem, we propose a new confidence algorithm that projects the

3D bounding box of the target object into the front view, end view and vertical view, and calculates Intersection over Union (IoU) respectively, then use the weighted average as the final confidence. We named the new confidence algorithm FIoU.

In short, the contributions of our paper are listed as follows:

- We propose MFPN which can efficiently and fully integrate features and effectively deal with occlusion and texture-less problems.
- We propose MFPN-6D which can estimate the pose of objects quickly.
- We propose a new confidence calculation method for object pose estimation, which can consider spatial information fully.

II. Related Work

In this section, we briefly summarize existing projects related to our topic, which only use RGB images to estimate the 6D pose of objects. Estimating the pose of an object by RGB images requires finding the 2D-3D correspondence between the 3D model and the image, then using the PnP algorithm for pose estimation. We roughly divide existing projects into two categories: traditional methods and CNN based methods. Here, we briefly review some representative methods in each category.

A. Classical methods

The method based on key points does not directly estimate the pose from the image, but first predicts the 2D key points of the object, and then associates it with the corresponding keypoints of the 3D model, and finally calculates the 6D pose with PnP algorithm. For objects with rich textures, traditional methods can detect the local texture of the object as key points, 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 have poor results when processing poorly textured objects and low-resolution images. Compared with sparse keypoint prediction, dense keypoint prediction means that each pixel or patch will predict the pose, and then get the final result by voting [11]. Some methods use the random forest algorithm to predict the 3D object coordinates corresponding to each pixel [15] [18], and use geometric constraints to generate 2D-3D correspondences, and finally estimate the pose of the object. Although the accuracy of dense key point prediction is higher than that of sparse keypoint prediction, the corresponding cost is that the running speed is very slow.

B. CNN-based methods

In order to solve the shortcomings of object pose estimation based on key points in the traditional field, many reseachers use CNN as a key point detector 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 not fast enough. BB8 [16] is a 6D pose estimation detector which divides the process of object pose estimation into two steps.The image is roughly segmented first, and then the keypoints are detected in the segmented object.The method is effective, but due to the multistage, the method cannot achieve the real-time requirements. BB8 [16] uses the CNN to predict the keypoints of the object. Segmentation-driven [11] and YOLO-6D [17] have the same method of predicting keypoints. For YOLO-6D, the advantage of this method is that it does not require a detailed 3D model, so it is not affected by the texture of the surface of the object. The advantage of this algorithm is that it is fast, but it also has the disadvantage of YOLO-V2 [12] itself that the accuracy will decrease when detecting small objects and occluded objects.

In order to effectively deal with weak texture and occlusion problems, while taking into account the running speed, we combined CSPNet and MFPN to form the MFPN-6D network for pose estimation. Proved by experiment, our method have achieved the requirements of speed and accuracy at the same time.

III. Approach

In this section, we first introduce the main ideas for our proposed MFPN: efficient multi-directional feature pyramid network, and then introduce the MFPN-6D network, at last, introduce the FIoU. The network we designed is an end-to-end trainable network, and it can predict 6D pose in real-time. Now, we describe our network architecture and explain various aspects of our approach in details.

A. Network architecture

1) MFPN: One of the main difficulties in object detection is how to effectively represent and process multi-scale features. Earlier detectors often directly perform predictions based on the pyramidal feature hierarchy extracted from backbone networks. As shown in Figure 1 (a), [31] proposes a feature pyramid network (FPN) which is a top-down pathway to combine mutli-scale feature, but FPN only disseminate information from top to bottom, and cannot disseminate information from bottom to top. In order to solve this problem, literature [32] 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 Figure 1 (b), Although PANet shows good accuracy in the field of object detection, it requires more parameters and computational costs. In order to improve the operating efficiency of the model and reduce the parameters, Google researchers proposed a multi-scale feature fusion method [28], and then proposed the BiFPN (as shown in Figure 1(c)): efficient bidirectional cross-scale connections and weighted feature fusion. BiFPN achieves a better accuracy than PANet, and with the less cost than PANet. BiFPN [28] is one of the most advanced feature networks, but it only considers the problem of forward feature propagation, without considering the problem of vertical propagation of feature which will result in loss of features during vertical propagation and cannot effectively use all feature information.

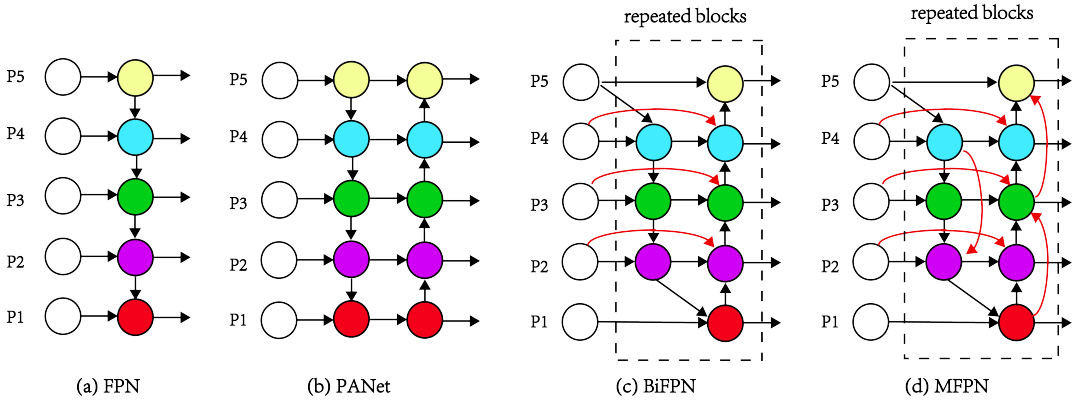

Fig. 1. Feature network design (a) FPN introduces a top-down pathway to fuse multi-scale features from level 1 to5 (P1 - P5); (b) PANet adds an additional bottom-up pathway on top of FPN; (c) BiFPN adds a residual structure to the lateral propagation direction of PANet; (d) MFPN is our network. It adds a residual structure to the vertical propagation direction of BiFPN to achieved a better accuracy and efficiency trade-offs.

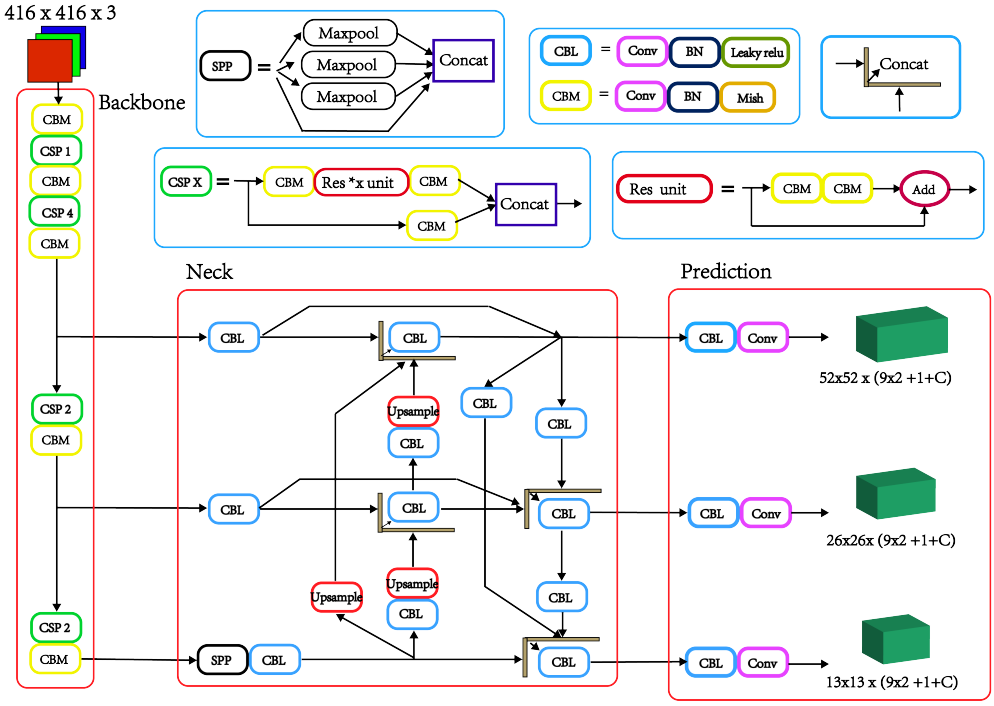

Fig. 2. We combined CSPNet with MFPN to propose a new network for 6D pose estimation, MFPN-6D

Therefore, in order to improve the utilization rate of input and extract effective features more fully, we apply the ideas of residual network and densely connected convolutional network to the feature network. The residual connection is added to the forward and vertical propagation of the feature network, and finally a new feature fusion extraction network is proposed. The new feature network is named MFPN, as shown in Figure 1(d). MFPN can improve the utilization of features in forward and vertical propagation.

2) MFPN-6D: CSPNet [22] as a backbone network used in YOLO-V4 [29] shows good performance, and the processing speed is improved a lot, so we use CSPNet as a backbone network for feature extraction, and then combine it with MFPN. Finally, the network MFPN-6D for pose estimation is designed, as shown in Figure 2.

We select the 8 vertices and center points of the bounding box of the 3D model as control points, similar to BB8 [16] and YOLO-6D [17]. This parameterization is conventional and can be used for any rigid 3D object with any shape and topology. Input a single full-color image into our model, processes it with a fully-convolution structure shown in Figure 2, and finally divide the image into a 2D regular grid containing $S \times S$ cells as same as [13]. Each grid location in the 3D tensor output by our model will be associated with a multidimensional vector, which includes the predicted 2D image position of 9 control points, the class probability and confidence of the object.

3) FIoU: As in YOLO-V2 [13], YOLO-V4 [29], 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 2D detection, the calculation method of confidence is mainly based on calculating the Intersection over Union (IoU) score. The higher value of IoU, the greater the possibility that the area contains objects. In case of 6D pose estimation, the objects are located in 3D space. If we want to calculate the equivalent IoU score between them, we need to calculate the overlap volume between the two 3D bounding box. This would be very time consuming and difficult.

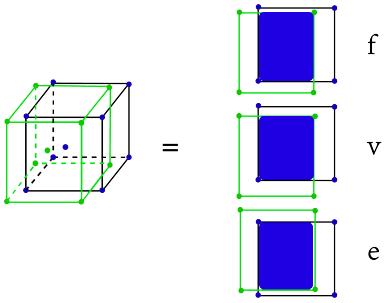

Fig. 3. A new confidence algorithm that projects the 3D bounding box of the target object into the front view, end view and vertical view, and calculates IoU respectively.

In YOLO-6D, a new confidence calculation method is used to project the predicted 3D bounding box vertex onto a 2D plane, and then compare it with the ground truth 2D projection point. By calculating the average offset of 9 projection points as a measure of confidence. But the problem is that although the offset in the 2D plane may be small, the offset will be large when mapped to the 3D space, resulting in inaccurate confidence. We propose a new confidence solution idea, FIoU, as shown in Figure 3. We predict the projection point of the target in the 2D space, and calculate the weighted value of the IoU between the three predicted faces and the ground truth as a measure of confidence, of course, Generalized Intersection over Union (GIoU) and Complete-Intersection over Union (CIoU) are ok. The advantage of this is that it takes full advantage of the relationship between the 3D information and each projection point. FIoU is defined as:

$$IoU = \frac{|A \bigcap B|}{|A \bigcup B|}, \tag{1}$$

$$FIoU = a \cdot f_{IoU} + b \cdot v_{IoU} + c \cdot e_{IoU}, \tag{2}$$

a, b, c are the weights of each projection surface, and $a+b+c=1$. Since the area of each projection surface is different, different weights are given.

We also predict the classification of each grid. In general, our output 3D tensor depicted in Figure 2 has dimension $S \times S \times D$, where the 2D spatial grid corresponding to the image dimensions has $S \times S$ cells 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.

B. Training Procedure

During training, MFPN-6D only need to know the center points and 3D bounding box corners of the object, not a detailed 3D model or texture map of object. We train the network to predict the projection position of the corner points of the 3D bounding box, as well as the classification and confidence of the target object. The confidence value is computed by the function defined in Eq. (2) to measure the $FIoU$. The network predict the offset of the 2D 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 limit the offset to between 0 and 1. The predicted control point (b_x, b_y) is defined as:

$$b_x = \sigma\,(t_x) + c_x, \tag{3}$$

$$b_y = \sigma\,(t_y) + c_y, \tag{4}$$

where $\sigma\,(\cdot)$ is chosen to be a 1D sigmoid function in case of the centroid and the identity function in case of the eight corner points. This forces the network to first find the location of the cell containing the center point of the object, and then refine its 8 corner loactions. The loss function during network training is as follows:

$$Loss = \lambda_{pt} \cdot Loss_{pt} + \lambda_{conf} \cdot Loss_{conf} + \lambda_{id} \cdot Loss_{id}, \tag{5}$$

where $Loss_{pt}$, $Loss_{conf}$, $Loss_{id}$ denote the coordinate, confidence and the classification loss.

Mean square error is used for coordinate and confidence losses, and cross entropy is used for classification loss. In order to improve the stability of the model, we refer to the method in the paper [17] and set λ_{conf} to 0.1 to reduce the confidence loss of cell that don't contain the object. For the cell containing the object, λ_{conf} is set to 5.0, λ_{pt} and λ_{id} are set to 1.

When multiple objects appear in the 3D scene, some cells may contain multiple objects. In order to be able to predict the poses of multiple objects in the same cell, we allow up to 3 candidates pre cell. For an input image of 416×416×3 , set a priori boxes in each grid of the feature map of each scale, a total of 13×13×3+26×26×3+52×52×3=10647 prediction candidates. Each predicted candidate is (2×9+1+13)=32 dimensional. This 32 dimensional vector contains coordinates (18 values), confidence (1 value), and object category probability (for LINEMOD datasets, there are 13 kinds of objects). Compared with YOLO-6D using 13×13×5=845 prediction candidates, our method attempt to predict the number of candidates has increased by more than 10 times, and it is performed at different resolutions, so the detection effect of small objects has a certain improvement. As in [17], we calculated the size of 3 anchor boxes by using the K-means 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 2D coordinates of the object.

C. 6D Object Pose Estimation

The MFPN-6D network is much efficient, only need to call the network once to estimate the pose of the 6D object. At test time, we estimate the class-specific confidence scores

TABLE I

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

	w/o refinement					w/ refinement	
Method	Brachmann [2]	BB8 [16]	YOLO-6D [17]	BiFPN-6D	MFPN-6D	BB8 [16]	Brachmann [2]
Ape	-	95.3	92.10	97.85	98.39	96.6	85.2
Benchvise	-	80.0	95.06	96.72	98.36	90.1	67.9
Cam	-	80.9	93.14	96.13	97.79	86.0	58.7
Can	-	84.1	97.44	95.89	97.78	91.2	70.8
Cat	-	97.0	97.41	92.98	94.75	98.8	84.2
Driller	-	74.1	79.41	91.20	94.92	80.9	73.9
Duck	-	81.2	94.65	96.71	97.95	92.2	73.1
Eggbox	-	97.9	90.33	96.28	98.40	91.0	83.1
Glue	-	89.0	96.53	96.22	96.28	92.3	74.2
Holepuncher	-	90.5	92.86	93.17	94.72	95.3	78.9
Iorn	-	78.9	82.94	94.93	96.57	84.8	83.6
Lamp	-	74.4	76.87	95.57	97.89	75.8	64.0
Phone	-	77.6	86.07	95.87	97.85	85.3	60.6
Average	69.5	83.9	90.37	**95.19**	**97.05**	89.3	73.7

for each object by multiplying the class probabilities and the score returned by the confidence function, and use the confidence threshold to prune units with low confidence predictions.

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

IV. EXPERIMENTAL EVALUATION

In this section we evaluate our approach on the LINEMOD [9] and Occluded-LINEMOD [2] datasets. At the same time, we compare our model to other state-of-the-art models for 6D pose estimation.

A. Datasets

The Linemod dataset is a popular benchmark dataset for 6D pose estimation of objects. Most algorithms are evaluated on this dataset, so it is persuasive to use this dataset to verify our algorithm. Linemod dataset consists of 13 different objects which are placed in different cluttered scenes. There are about 1200 images for each object. Although each image contains multiple objects, only one object has 6D pose annotation information. Occlusion dataset is a subset of Linemod. 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 data set can evaluate the performance of the algorithm in the case of multi-object stacking and occlusion.

B. Evaluation Metrics

We use two kinds of standard metrics to evaluate the accuracy of 6D pose estimation, one is 2D reprojection error, and the other is *ADD* score [2] [14]. When we use the 2D reprojection error as a measure of 6D pose estimation, we consider the 6D pose predicted by the model is correct if the average distance between the 2D projection point of the 3D vertex of the object and the ground truth is less than 5 pixels. When comparing 6D poses using the *ADD* metric. *ADD* is defined as an average Euclidean distance between model vertices transformed with the predicted and the ground truth pose. 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:

$$m = \underset{x \in M}{avg} \left\| (Rx+t) - \left(R^{'}x+t^{'}\right) \right\|_2, \tag{6}$$

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. The *ADD* metric can be extended to solve symmetric objects, as shown in [9]:

$$m = \underset{x_2 \in M}{avg} \underset{x_1 \in M}{min} \left\| (Rx_1+t) - \left(R^{'}x_2+t^{'}\right) \right\|_2. \tag{7}$$

C. Pose Estimation on LINEMOD Dataset

In order to compare the effects of MFPN and BiFPN, we replaced the feature network in MFPN-6D with BiFPN, and thus proposed BiFPN-6D.In order to prove the effectiveness of our proposed 6D pose estimation algorithm, we compare it with the most advanced 6D pose estimation algorithm. At the same time, we compare with BiFPN-6D to prove that our MFPN network is better. It can be seen from Table I and Table II that our algorithm is comparable to the best result of this dataset. It can be seen from Table I that MFPN-6D is 1.86% higher than BiFPN-6D. From Table II, it can be seen that MFPN-6D is 3.27% higher than BiFPN-6D, which fully shows that MFPN is more effective than BiFPN. In Table III, we report the computational efficiency of our approach for single object pose estimation in comparison to the state-of-the-art approaches [2] [14] [16] [17] [24]. Compared with existing methods, our method is the fastest. And when detecting multiple targets, our algorithm don't takes extra time, but [24] needs to spend an extra 5 ms on each object. In terms of model size, our model is the smallest, only 29 M. It is 1/20 of SSD-6D , and about 1/5 of DPOD. It is very suitable for embedded and mobile terminals.

Fig. 4. The pose estimation results of our method on the datasets LINEMOD and Occluded LINEMOD, and our method is also effective for occluded objects

TABLE II

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

	w/o refinement						w/ refinement		
Method	SSD-6D [14]	BB8 [16]	YOLO-6D [17]	DPOD [24]	BiFPN-6D	MFPN-6D	SSD-6D [14]	BB8 [16]	DPOD [24]
Ape	0	27.9	21.62	53.28	39.89	42.65	65	40.4	87.73
Benchvise	0.18	62.0	81.80	95.34	85.70	87.43	80	91.8	98.45
Cam	0.41	40.1	36.57	90.36	78.84	81.48	78	55.7	96.07
Can	1.35	48.1	68.8	94.10	92.17	93.33	86	64.1	99.71
Cat	0.51	45.2	41.82	60.38	53.56	59.29	70	62.6	94.71
Driller	2.58	58.6	63.51	97.72	82.98	86.48	73	74.4	98.80
Duck	0	32.8	27.23	66.01	53.77	56.24	66	44.3	86.29
Eggbox	8.9	40.0	69.58	99.72	81.31	83.40	100	57.8	99.91
Glue	0	27.0	80.02	93.83	76.92	80.85	100	41.2	96.82
Holepuncher	0.30	42.4	42.63	65.83	65.87	67.74	49	67.2	86.87
Iorn	8.86	67.0	74.97	99.80	82.13	85.87	78	84.7	100
Lamp	8.20	39.9	71.11	88.11	75.94	82.91	73	76.5	96.84
Phone	0.18	35.2	47.74	74.24	64.48	68.41	79	54.0	94.69
Average	2.42	43.6	55.95	82.98	**71.81**	**75.08**	79	62.7	95.15

TABLE III

COMPARISON OF THE OVERALL COMPUTATIONAL RUNTIME OF OUR APPROACH IN COMPARISON TO STATE-OF-THE-ART METHOD ON THE OCCLUDED-LINEMOD DATASETS

Method	Overall Speed	Refinement	Model size
Brachmann [2]	2 fps	100 ms/object	-
BB8 [16]	3 fps	21 ms/object	-
SSD-6D [14]	10 fps	24 ms/object	590 M
DPOD [24]	33 fps	5 ms/object	147 M
YOLO-6D [17]	50 fps	-	192 M
BiFPN-6D	56 fps	-	29 M
MFPN-6D	56 fps	-	29 M

D. Pose Estimation on Occluded-LINEMOD Dataset

We evaluate the model which we designed on the Occluded dataset. The accuracy of object detection on the OCCLUSION dataset is usually reported in terms of mean average accuracy (mAP). Our algorithm has a final score of 0.53, which is the best result of this data set (see Table IV). No need refinement, and the proposed detector shows very efficient results compared with other detectors. The experimental results demonstrated that our algorithm can significantly robust against occlusion, and run much efficiently.

TABLE IV

DETECTION PERFORMANCE FOR MULTIPLE OBJECTS: COMPARISON OF THE STATE-OF-THE-ART MEAN AVERAGE PRECISION (MAP) SCORES ON THE OCCLUDED-LINEMOD DATASETS]

Method	YOLO-6D [17]	Brach [2]	DPOD [24]	BiFPN-6D	MFPN-6D
mAP	0.48	0.51	0.48	**0.51**	**0.53**

V. CONCLUSION

We propose MFPN, which efficiently performs multi-scale feature fusion, effectively detects small target objects. MFPN is significantly robust against occlusions, texture-less and scene confusion. In order to improve the efficiency of detection, we combine MFPN and CSPNet to form MFPN-6D network for 6D pose estimation. MFPN-6D can predict the projection position of the 3D bounding box corners of the target object in the 2D space. When the correspondence between 3D points and 2D points is given, we can calculate 6D poses through PnP algorithm. Many existing methods add the refinement stage in order to improve the accuracy, which leads to the long detection process and it is difficult to realize real-time detection. In contrast, our single-shot prediction is very accurate and does not require the refine stage, thereby reducing the detection time. The experimental results demonstrated that MFPN-6D can significantly robust against occlusion, and run much efficiently.

REFERENCES

[1] Mercier, Jean-Philippe and Mitash, Chaitanya and Giguere, Philippe and Boularias, Abdeslam. Learning object localization and 6D pose estimation from simulation and weakly labeled real images. IEEE International Conference on Robotics and Automation (ICRA), 2019, (pp.3500-3506).

[2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. In European Conference on Computer Vision (ECCV), 2014, (pp.536-551).

[3] Tommaso Cavallari, Stuart Golodetz, Nicholas Lord, Julien Valentin, Victor Prisacariu, Luigi Di Stefano, and Philip HS Torr. Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2019.

[4] Changhyun Choi and Henrik I Christensen. RGB-D object pose estimation in unstructured environments. Robotics and Autonomous Systems, 2016, (pp.595-613).

[5] Mur-Artal, Raul and Tard, Juan D. Orb-slam2: An open-source slam system for Monocular, Stereo, and RGB-D cameras. IEEE Transactions on Robotics, 2017, (pp.1255-1262).

[6] Park, Kiru and Patten, Timothy and Prankl, Johann and Vincze, Markus. Multi-task template matching for object detection, segmentation and pose estimation using depth images. IEEE International Conference on Robotics and Automation (ICRA), 2019, (pp.7207-7213).

[7] Haoruo Zhang and Qixin Cao. Detect in RGB, Optimize in Edge: Accurate 6D Pose Estimation for Texture-less Industrial Parts. IEEE International Conference on Robotics and Automation (ICRA), 2019, (pp.3486-3492).

[8] Zhigang Li, Xiangyang Ji. Pose-guided Auto-Encoder and Feature-Based Refinement for 6-DoF Object Pose Regression. IEEE International Conference on Robotics and Automation (ICRA), 2020, (pp.8397–8403).

[9] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stean Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In Asian Conference on Computer Vision (ICCV), 2012, (pp.548-562).

[10] Porzi, Lorenzo and Penate-Sanchez, Adrian and Ricci, Elisa and Moreno-Noguer, Francesc. Depth-aware convolutional neural networks for accurate 3D pose estimation in RGB-D images. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, (pp.5777-5783).

[11] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6D Object Pose Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, (pp.3385-3394).

[12] Tomas Hodan, Pavel Haluza, Stepan Obdrzalek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: an RGB-D Dataset for 6D pose estimation of texture-less objects. In IEEE Winter Conference on Applications of Computer Vision, 2017, (pp.880-888).

[13] Joseph Redmon, Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, (pp.7263-7271).

[14] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-based 3D detection and 6d pose estimation great again. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, (pp.1530-1538).

[15] Muoz, Enrique and Konishi, Yoshinori and Beltran, C and Murino, Vittorio and Del Bue, Alessio. Fast 6D pose from a single RGB image using Cascaded Forests Templates. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, (pp.4062-4069).

[16] Mahdi Rad and Vincent Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, (pp.880-888).

[17] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6D object pose prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, (pp.292-301).

[18] Tejani, Alykhan and Kouskouridas, Rigas and Doumanoglou, Andreas and Tang, Danhang and Kim, Tae-Kyun. Latent-class hough forests for 6 DoF object pose estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2017, (pp.119-132).

[19] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, (pp.2938-2946).

[20] Wadim Kehl, Faysto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep Learning of Local RGB-D Patches for 3D Object Detection and 6D Pose Estimation. In Proceedings of the European Conference on Computer Vision (ECCV), 2016, (pp.205-220).

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), 2016, (pp.21-37).

[22] Chien-Yao Wang, Mark Liao, Hong-Yuan,Wu, Yueh-Hua, Ping-Yang Chen, Hsieh, Jun-Wei and Yeh I-Hau. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[23] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6 DOF pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, (pp.4561-4570).

[24] Sergey Zakharov, Ivan Shugurov, Slobodan Ilic. DPOD: 6D pose object detector and refiner. In Proceedings of the IEEE International Conference on Computer Vision (CVPR), 2019, (pp.1941-1950).

[25] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards RealTime Object Detection with Region Proposal Networks. In Conference and Workshop on Neural Information Processing Systems (NIPS), 2015, (pp.91–99).

[26] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D orientation learning for 6D object detection from RGB images. In Proceedings of the European Conference on Computer Vision (ECCV), 2018, (pp.699-715).

[27] Guoguang Du, Kai Wang, Shiguo Lian, Kaiyong Zhao. Vision-based Robotic Grasping from Object Localization, Pose Estimation, Grasp Detection to Motion Planning: A Review, 2019.

[28] Mingxing Tan Ruoming Pang Quoc V. Le. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2020, (pp.10781-10790).

[29] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[30] Bo Cheng, Wanyin Wu, Dapeng Tao, Shibo Mei, Ting Mao, Jun Cheng, Random Cropping Ensemble Neural Network for Image Classification in a Robotic Arm Grasping System, IEEE Transactions on Instrumentation and Measurement, 69(9): 6795-6806, 2020.

[31] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[32] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.