Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

A Multi-task Learning Convolutional Neural Network for Object Pose Estimation*

Yurui Wang [1,2,3], Shaokun Jin [1,2,3], Yongsheng Ou [1,3,†]

***Abstract*— Estimating 6D poses of objects from RGB images is very crucial for robots to interact with the surrounding environment and to cooperate with humans. It is a challenging problem due to the various shapes of objects, the occlusions among objects, as well as the complexity of the scene. In this study, we present a new multi-task convolutional neural network for 6D object pose estimation, which also learns the object region in the image to further improve the estimation result. A weighted loss computed from the training samples is adopted to guarantee that the components of the estimated pose are equally accurate. Additionally, we sythesize new virtual data based on the existing training data so as to overcome the possible adverse situation caused by insufficiency of training data. Experiments on the YCB-video dataset are undertaken to validate the effectiveness of the proposed method.**

***Index Terms*— Pose Estimation, Multitasking network, Deep learning, Semantic segmentation**

I. Introduction

Object pose estimation is a very important technique and has a wide range of applications in robotics, autopilot, virtual reality, etc. For example, in robotic grabbing and operation, the pose estimation of the object is first required. The robot first needs to recognize the object and estimate the robot-object range to complete the next grabbing action. However, the pose estimation of objects is a challenging problem, and it is difficult to find an efficient and robust solution. Because the shape of the estimated object is always varied, and objects have different properties, such as no texture, smooth surface, transparent object, etc. These characteristics will bring great generalization requirements for object recognition and pose estimation. Also in the case where the non-structural, the target portion of the occlusion, but also brought great uncertainty to the estimation of the object pose.

The traditional classical method mainly uses the relationship between the image and the target template to infer the pose of the object. For better results, a large amount of information is required in advance to collect information about the target object. This type of method is usually intuitive and effective but usually do not handle occlusion, light changes, etc. very well. In recent years, the emergence of low-cost depth camera, such as kinect, has spawned many new pose estimation method based on depth data[6], [7], [8], [9]. Such methods by combining the depth data of traditional RGB cameras can not provide, can usually get a better result of object pose estimation. However, depth cameras have certain limitations in use, such as being sensitive to the environment, and not well applied to raster types or smooth surfaces, and inaccessible depth of these objects that are too close to the camera. Another type of method is to use the deep learning network to perform pose estimation only through RGB. This type of method is more robust and has made some progress, but there is still room for improvement in accuracy.

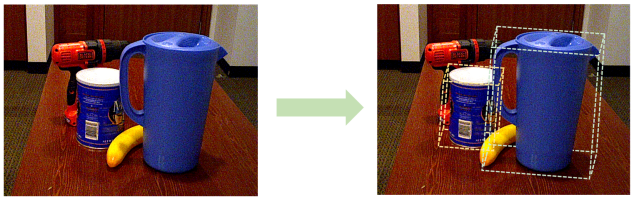

Fig. 1. Our target is to estimate the rigid transformation from the object coordinate system O to the camera coordinate system C given RGB image.

In our work, we present a deep learning multitasking end-to-end network for estimating the object pose based on RGB images. Our network consists of two parts: the first part realizes the initial recognition and segmentation of the object, and the second part uses the segmented Mask combined with the RGB image to estimate the 6D pose of the target object. We learn the position and target object pose, and the mask corresponding to the object region in the RGB image, where in the mask learning is similar to the attention mechanism. By learning the mask, the weight in the CNN can effectively to focus on the region where the object locates on the image.

We validate our approach on the YCB-video dataset. To compensate for the lack of data and improve scalability, we try to use only trained pose estimation network through three-dimensional model synthesis picture dataset, and also validate in true dataset.

[1]CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology. [2]University of Chinese Academy of Sciences, Beijing, China. [3]Guangdong Provincial Key Lab of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences.

[†]Yongsheng Ou is the corresponding author (ys.ou@siat.ac.cn).

*This work was jointly supported by National Natural Science Foundation of China (Grant No. U1613210), Guangdong Special Support Program (2017TX04X265), Science and Technology Planning Project of Guangdong Province (2019B090915002), and Shenzhen Fundamental Research Program (JCYJ20170413165528221).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

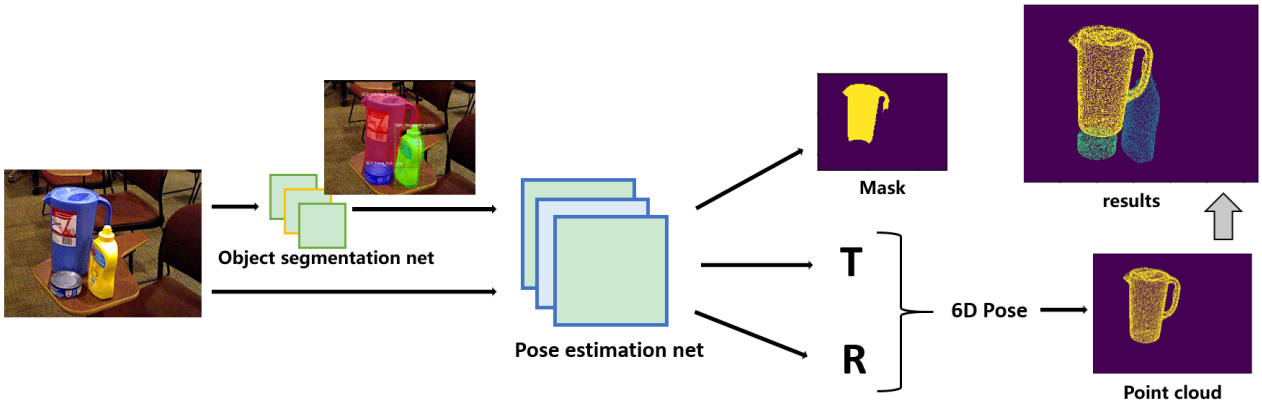

Fig. 2. Overview of our 6D pose estimation model. This framework is divided into two main networks. The first part is object segmentation net to identify and segment the Mask of each object from the RGB image. The second part is pose estimation net, we feed the corresponding Mask and RGB image into this network for the target object to be estimated.

II. Related Work

In recent years, most problems in the field of computer vision use deep learning. For simplicity the 6D pose method can be divided into an RGB-based method and an RGB-D-based method, which uses a depth camera.

The emergence of low-cost depth cameras has spawned many new pose estimation methods. Deep cameras can obtain depth data and point cloud. Stoisser et al proposes a template matching method based on color and depth data[10]. Brachmann et al.[3] proposed the regression of forest-predicted object coordinates, segmenting the target, and recovering the pose from the dense correspondence. Chen Wang et al. [11] proposed an iterative network based on RGB-D, which separated objects from RGB and connected point cloud data to the network through PointNet [12].Combine the two features to obtain the pose estimation of the target object through the network, and further iterate through the difference between the estimated pose and the true pose to improve the accuracy.

Many depth networks have been used to estimate poses from RGB images[13], [14], [15]. Using the techniques of viewpoint, key point[16] and CNN, object classification and pose estimation become classification problems by discretizing pose estimation in three-dimensional space. Markus et al.[18] proposed using a network to extract feature heat maps and recovering object poses based on PnP correspondence.PoseNet[17] directly uses CNN to estimate from RGB images to poses. PoseCNN[15] has been improved on PoseNet, decoupling translation and rotation prediction, predicting 6D object poses from a single RGB image in multiple stages. In [19], CNNs infer 6D pose by outputting two-dimensional coordinates and two-dimensional mask.

The key features of the proposed method can be listed as follows:

1. We used a new weighted loss function. We propose the training data such that the error on each component of the network output is reasonable.

2. In order to overcome the negative effects caused by insufficient training data, we divided the whole system into two networks: object recognition and segmentation and pose estimation. The object recognition network uses real data for training, and the pose estimation network uses synthetic data for training.

3. Inspired from the attention mechanism of LSTM network, we let the proposed CNN to learn the mask such that the entire network focuses on the object in the specified mask range and estimates the pose of the object.

4. A virtual dataset of RGB images is synthesized based on the exsiting dataset of point clouds, by adding distinct backgrounds to them. This self-made dataset has been verified to better train the network parameters, therefore further improving the network performance.

III. Model

Our target is to estimate the transformation from the object coordinate system O to the camera coordinate system C when given an RGB image. The transformation consists of a translation T and a rotation R, the translation matrix consists of the Euclidean space distance between the camera origin and the origin of the target object. The rotation matrix has multiple representation methods: rotation matrix, Euler angle, quaternion, etc. In our method we use quaternion as a representation of the rotation. A pose p is $p = [R|T]$. We propose a convolutional neural network to estimate R and T, respectively.

A. Overview of the Network

The framework of our entire network is shown in Figure 2. The entire framework is divided into two main parts. The main task of the first part is to identify and segment the Mask

of each object to be estimated from the RGB image. In the second part, we feed the corresponding Mask and color image into the second network for each object to be estimated.

B. Semantic Segmentation

The first step is to segment the objects of interest in the input image. Our semantic segmentation network is an encoder-decoder architecture that takes an image as input and generates an N +1-channelled semantic segmentation map. Each channel is a binary mask where active pixels depict objects of each of the N possible known classes. At present, some networks have achieved very good results in object recognition and segmentation, thus we use an existing segmentation architecture proposed by Mask r-cnn[21], this network classifies and segments objects on RGB images.

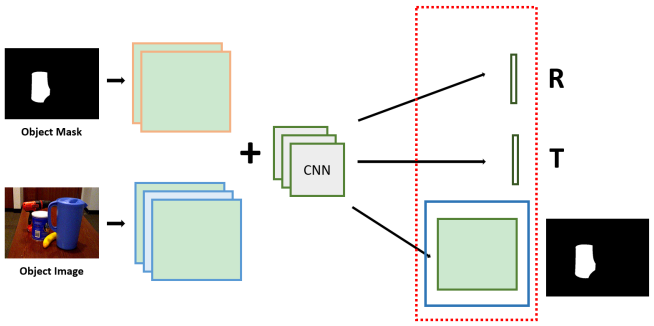

Fig. 3. Multi-task pose estimation network. The input of the network is a single RGB image, with a well trained object segmentation network to acquire a mask from the input image. The red dotted block represents the output of the neural network, wherein the blue solid block is an output branch of the dimension same with that of the input image. The output branch of this block (blue solid) is to learn a mask that segments the region of the object. In this way, the network parameters hopefully implicate the function of paying attention to the object region, which further facilitates the pose estimation of the target object.

C. 6D Object Estimation

This part of the network accepts the RGB data and the first part of the semantic segmentation Mask as a network input. RGB images are first extracted by VGG16 network which initialized with the weights trained on ImageNet to obtain the feature map after 5 times of downsampling. In order to improve the robustness of the whole network to the segmentation error, we do not directly reduce the mask dimensions to the same dimension as the feature map, but after a certain convolution and downsampling. The convolution kernels are (11,11), (7,7), (3,3) separately. The resolutions of the mask feature maps are 1/32 of the original Mask size, merged after the same dimensions as the image features extracted by VGG and Mask by Mask r-cnn[21], the merged features are further fed into the convolutional network. For reducing the network parameters, the convolved featrue maps are pooled before fed into two Fully-Connected(FC) layers to predicting the translation matrix and the rotation matrix separately. On the other hand, the feature map is upsampling and convolved, and outputs the same dimension as the image. In this way, the network parameters hopefully implicate the function of paying attention to the object region, which further facilitates the pose estimation of the target object.

The network prediction translation prediction is $T = (x_1, x_2, x_3)$, representing the three outputs of the network prediction translation matrix respectively. The corresponding true translation matrix is $\tilde{T} = (\tilde{x_1}, \tilde{x_2}, \tilde{x_3})$. Similarly, we use quaternion to represent rotation R, $\tilde{R}$.

1) Translation and Rotation Regression: The two FC layers have dimensions 2048, and the last FC layers has dimensions 3, 4 for translation and rotation respectively. Traditional regression predictions often use mse as a loss. The movement of the object in the three-dimensional space is inconsistent with the change in the two-dimensional image, in order to estimate the pose of the object directly from the RGB map, we propose an improved mse loss

$$loss = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{P}\|\tilde{\omega_j}(x_{ji} - \tilde{x}_{ji})\|^2 \tag{1}$$

where x_{ji} represents the j^{th} component of the i^{th} label corresponding to the i^{th} training sample. $\tilde{x}_{ji}$ denotes the similar counterpart of the network prediction. P represents the dimension of label, M denotes the number of samples, $\tilde{\omega}_j$ as the weight of each axis. The weight is calculated as follows

$$\omega_j = (\frac{1}{M}\sum_{k=1}^{M}\|x_k\|_1)^{-1} \tag{2}$$

$$\tilde{\omega}_j = \frac{\omega_j}{\sum_{i=1}^{N}\omega_i} \tag{3}$$

Where N represents the number of samples, and k^{th} represents the kth value of the samples.

2) Mask Learning: In order for the network to estimate the pose of a particular object, we have joined the learning of Mask inspired by attention mechanism. We let the neural network focus on the image area of the estimated object, and reduce the attention of the network to other areas by regularization, thereby improving the accuracy of the pose estimation of the network to a specific object. We choose dice loss as the loss function

$$diceloss = 1 - \frac{2\,|X \bigcap Y|}{|X| + |Y|} \tag{4}$$

X represents the output of Mask, and Y represents Mask in ground truth. By adding regularization the network focus on object areas of image, we achieved better results when training with synthetic image dataset.

IV. Experiments

A. Datasets

The YCB-Video dataset[15] consists of 92 video sequences, where 12 sequences are used for testing and the remaining 80 sequences for training. There are 21 objects in the dataset, which are high-quality 3D models. The dataset contained 80k synthetically rendered images used for training our model to test the model expansibility without real data. In addition those synthetically rendered images used with random backgrounds sampled from VOC2007[20].

B. Metric

We use the average distance (ADD) metric to evaluate our model. The average distance computes the mean distance between the cloud points transformed according to the ground truth pose and the estimated pose:

$$ADD = \frac{1}{b}\sum_{X\in B}\left\|(Rx+T)-(\tilde{R}x+\tilde{T})\right\|_2 \quad (5)$$

where the ground truth translation T and rotation R and the estimated translation $\tilde{T}$ and rotation $\tilde{R}$, B denotes the points of 3D model, b is the number of the model points.

The closest point distance is computed as average distance used for some experiments:

$$ADD\text{-}S = \frac{1}{b}\sum_{X\in B}\min_{X_2\in B}\left\|(Rx_1+T)-(\tilde{R}x_2+\tilde{T})\right\|_2 \quad (6)$$

although the symmetric object can be handled relatively well, this metrics does not well represent the influence of the rotation R when these objects are asymmetrical.

C. Implementation Details

Our network is implemented using the Keras library, the backend of keras is tensorflow. The network's minibatch size is 32, a learning rate is 0.001 and 10 iterations. RMSPROP is used as optimizer for training. The parameters of the first 13 convolutional layers in the pose estimation network extraction stage are initialized with the VGG16 network weights [2] trained on ImageNet[1].

Fig. 4. Synthetically rendered images with background added.

D. Result on YCB-Video Dataset

Regressing image pixels to 3D object coordinates [3], [4] is used by the state-of-the-art 6D pose estimation methods, we use a variant of the 6D object coordinate network for comparison. In addition, we also conducted experimental comparisons with PoseCNN[16].

Table I lists a detailed evaluation of the YCB video data set. We use the ADD metric to display the area under the accuracy-threshold curve, where we use a maximum threshold of 10 cm and then calculate the pose accuracy.

TABLE I
RESULTS ON YCB-DATASET(ADD)

Object	3D Coordinate	PoseCNN	Our	Our(syn)
master chef can	12.3	50.9	35.8	**62.7**
cracker box	16.8	51.7	34.5	**80.8**
sugar box	28.7	68.6	42.1	**83.8**
tomato soup can	27.3	66.0	**70.2**	60.4
mustard bottle	25.9	79.9	69.4	**85.1**
tuna fish can	5.4	70.4	75.3	**75.4**
pudding box	14.9	62.9	**71.9**	17.7
gelatin box	25.4	75.2	**88.3**	79.9
potted meat can	18.7	59.6	**88.1**	55.0
banana	3.2	**72.3**	55.9	59.6
pitcher base	27.3	52.5	53.2	**96.1**
bleach cleanser	25.2	50.5	61.03	**89.4**
bowl	2.7	6.5	**58.8**	49.5
mug	9.0	57.7	31.5	**87.7**
power drill	18.0	55.1	41.0	**96.4**
wood block	1.2	31.8	26.0	**43.8**
scissors	1.0	35.8	45.3	**60.2**
large marker	0.2	58.0	79.6	**87.5**
large clamp	6.9	25.0	49.7	**90.7**
extra large clamp	2.7	15.8	26.4	**88.1**
foam brick	0.6	40.4	**77.8**	26.3
MEAN	15.1	53.7	55.9	**70.3**

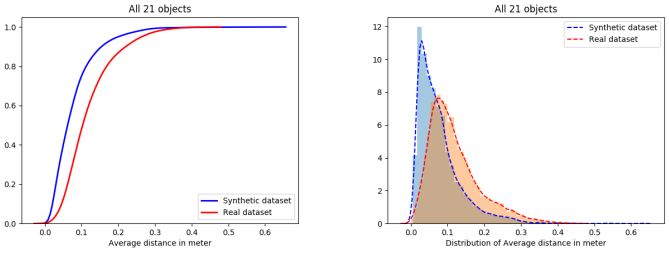

Fig. 5. The result between the synthetic dataset and the real dataset in our model.

The fourth column is the result of training the network in the real dataset. We can see that: Our network significantly outperforms the 3D Coordinate (the second column) by only using color images. In addition, we are significantly better than PoseCNN (third column) in some data. Because

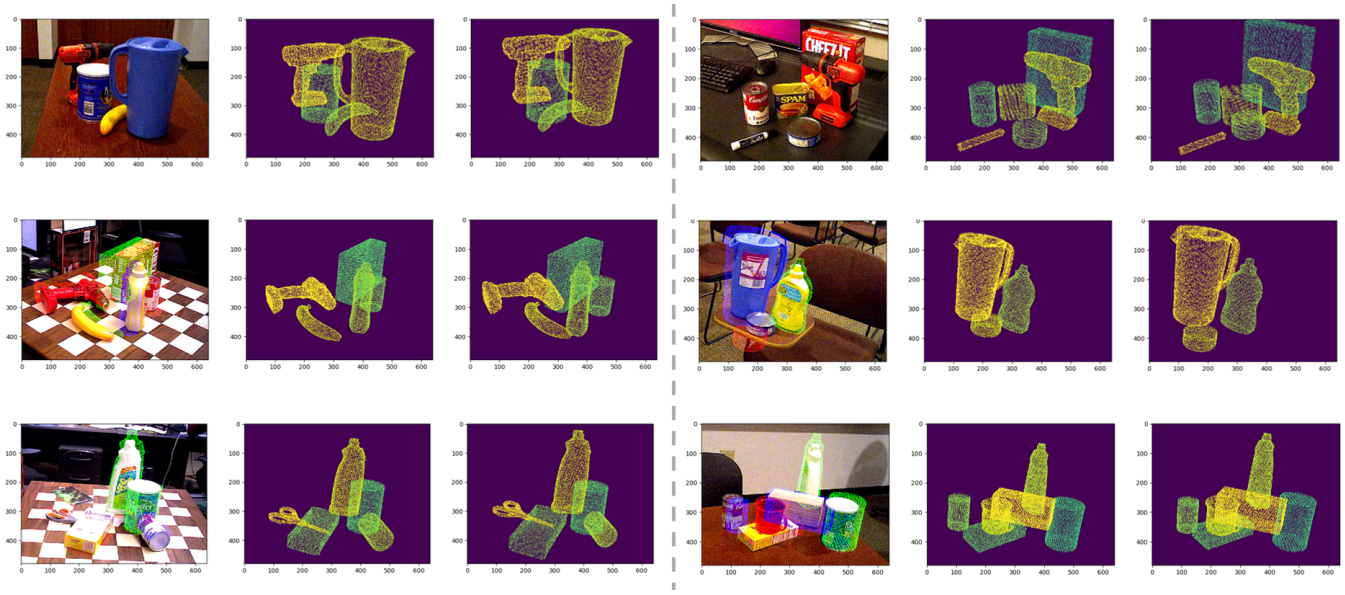

Fig. 6. The figure shows the visualization of the results predicted in the translation matrix. The left column is the input of the RGB image, the middle column is the point cloud image generated according to ground truth, and the right column is the point cloud image generated according to the network prediction. The above two lines are the result of the network trained real data, and the last line is the result of the network trained only synthetic data.

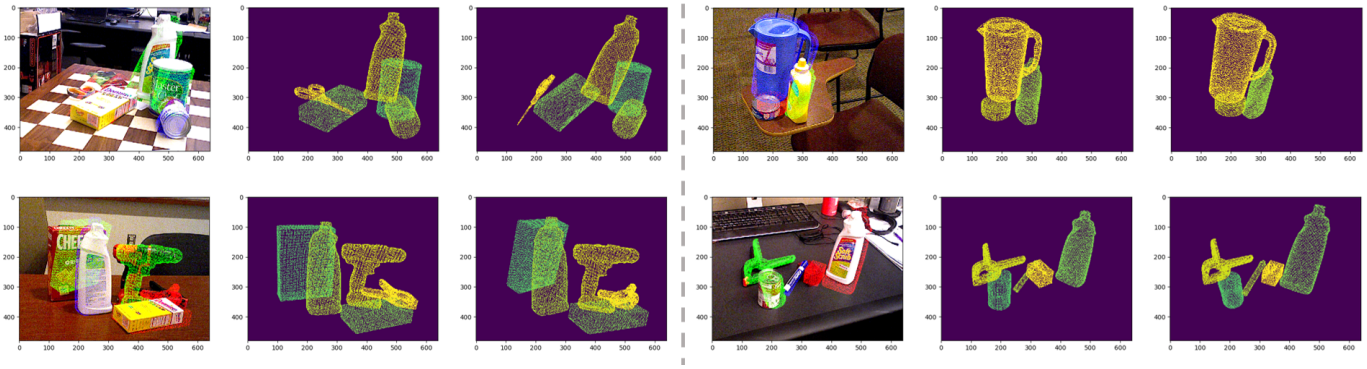

Fig. 7. The figure shows the prediction results of training the network using synthetic dataset. The left column is the input of the RGB image, the middle column is the ground truth, and the right column is our prediction result.

PoseCNN first uses the estimated center point and depth values of the object in the image, and then calculates the specific value of T according to the camera model, use hough voting to select the object center point for each pixel in the estimation of the object center. However, as shown in the table I, this method does not work well on objects with less texture. Our method, by using the new loss to estimate the pose of the object from the global information of the image, can achieve better results on less textured objects.

E. Result on Synthetic Dataset

In practical applications, obtaining data for a pose estimation network is costly in terms of both time and hardware. We tried to use the 3D model synthesized image for our pose estimation network training, and the Synthetic image randomly selected a picture from the VOC2007[20] dataset as the background image. We use 80,000 synthetic virtual images provided in the YCB-video dataset as training data for the pose estimation network, while using real data to train our object recognition and segmentation networks.Figure 4 shows the simulation training data with background added. The object estimation network exhibits better scalability in the simulated data by combining the Mask of the object recognition network trained by the real dataset.

In order to use Mask in the pose estimation network, we simulated the object recognition & segmentation network by performing a random and different size kernel expansion and erosion on the Mask provided by ground truth. Through the synthesized image, we can obtain a near-infinite data set for the training of our pose estimation network, and only need

to acquire the 3D model of the target object to generate new synthesized image and pose data for feeding the pose estimation network, improves the scalability of the entire system and avoids the need to reacquire the pose data of a large number of target objects in real world. By decoupling the object recognition from the pose estimation network, the influence of synthesized image data is limited to the pose estimation network, which reduces the influence of the difference between the synthesized image and the real image on the whole system.

The last column is the result of training the network in the synthetic dataset. Thanks to a large amount of data, the pose estimation network trained using synthesized image is better than the result of using the dataset in real world. As can be seen from Figure 5, our network can predict the location of the target in case of severe occlusion or complex background.

V. Conclusion

In this work, we introduce a multi-task convolutional network model for 6D object pose estimation, which is divided into two parts: object recognition & segmentation network and pose estimation network. Combining object segmentation results and mask learning in the pose estimation network, we have also achieved good results in the case of training using only the synthetic data for the pose estimation network. On the pose estimation network, the object 6D pose is predicted directly from the image by optimizing the loss function. The results show that our network can estimate the effective pose of occluded objects in complex scenes. Besides, the proposed method can be applied in the conditions with insufficient training data and without using depth camera. In the future, some improvements can be made to increase estimation accuracy.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database", *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248255, 2009.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv preprint arXiv:1409.1556, 2014.

[3] 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)*, pp. 536551, 2014.

[4] Eric Brachmann, Frank Michel, Alexander Krull,Michael Ying Yang, Stefan Gumhold, and Carsten Rother, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image", *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33643372, 2016.

[5] M. Young, The Technical Writer's Handbook, Mill Valley, CA: University Science, 1989.

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

[7] Kehl, Wadim, et al, "Deep learning of local RGB-D patches for 3D object detection and 6D pose estimation.", *European Conference on Computer Vision*, Springer, Cham, 2016.

[8] Lai, Kevin, et al, "A scalable tree-based approach for joint object and pose recognition.", *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[9] Zhang, Haoruo, and Qixin Cao. "Combined holistic and local patches for recovering 6D object pose." *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 2219-2227. 2017.

[10] Hinterstoisser, Stefan, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes." *In 2011 international conference on computer vision*, pp. 858-865. IEEE, 2011.

[11] Wang, Chen, Danfei Xu, Yuke Zhu, Roberto Martn-Martn, Cewu Lu, Li Fei-Fei, and Silvio Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion." *In IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3343-3352. 2019.

[12] Qi, Charles R., Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data.", *In IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918-927. 2018.

[13] Crivellaro, Alberto, Mahdi Rad, Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit, "A novel representation of parts for accurate 3D object detection and tracking in monocular images.", *In Proceedings of the IEEE international conference on computer vision*, pp. 4391-4399. 2015.

[14] Rad, Mahdi, and Vincent Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth.", *In IEEE International Conference on Computer Vision*, pp. 3828-3836. 2017.

[15] Xiang, Yu, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes." *arXiv preprint arXiv*:1711.00199 (2017).

[16] Tulsiani, Shubham, and Jitendra Malik, "Viewpoints and keypoints.", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[17] Kendall, Alex, Matthew Grimes, and Roberto Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization.", *In IEEE international conference on computer vision*, pp. 2938-2946. 2015.

[18] Oberweger, Markus, Mahdi Rad, and Vincent Lepetit, "Making deep heatmaps robust to partial occlusions for 3d object pose estimation.", *In European Conference on Computer Vision (ECCV)*, pp. 119-134. 2018.

[19] Mahendran, Siddharth, Haider Ali, and Ren Vidal, "3D pose regression using convolutional neural networks.", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2174-2182. 2017.

[20] Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The PASCAL visual object classes challenge 2007 (VOC2007) results." (2007).

[21] He, Kaiming, Georgia Gkioxari, Piotr Dollr, and Ross Girshick, "Mask r-cnn.", *In Proceedings of the IEEE international conference on computer vision*, pp. 2961-2969. 2017.