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

A 3DoF Pose Estimation Method for Multi-Trolley from a Single RGB Image

Xiaomeng Liu[1], Wenbo Han[1], Shuang Song[1]*, Max Q.-H. Meng[2], *Fellow, IEEE*

***Abstract*—3DoF pose estimation for multi-trolley is pivotal in Automatic Collection Trolley Robot System. Because of the challenge of occlusion and low-resolution problem, the existing methods do not solve these problems well. In this paper we propose a 3DoF pose estimation method for multi-trolley based on the keypoints from a single RGB image. The proposed method is divided into three steps: Firstly, YOLOv3 is used to detect the trolley in a RGB image and crop the trolley individually. Secondly a new network including two parts is proposed for detecting the 2D keypoints of the trolley. One part called DetectNet is used for detecting easy keypoints, while the other part called HardNet is for detecting hard keypoints under occlusion followed by post-processing for finetuning. Finally, 3DoF pose estimation of a trolley can be obtained by the corresponding relationship between the 2D keypoints and the real 3D keypoints in trolley 3D model. Experiments have been carried out to validate the proposed model. The results show that our approach is accuracy and robust for 3DoF pose estimation of trolley.**

***Index Terms*—pose estimation, keypoints, post-processing, PNP.**

I. Introduction

Automatic Collection Trolley Robot System is aimed to collect trolleys in airport without human control. A complete pipline in this system includes trolley detection [1], trolley tracking, trolley pose estimation, navigation and visual control. Pose Estimation for trolley is the first step in this system, which makes very important. The reason why RGB-D-camera is not used is that the trolley is a hollow object. On one hand, it is hard to deal with point noise, which is very time consuming when matching by ICP. On the other hand, the result is unsatisfactory.

This paper is aimed to estimate the 3DoF pose for Multi-Trolley from a single RGB image, shown in Fig. 1. At present, domestic and foreign scholars have systematically studied this problem.

In the traditional method the template matching method is use to identify the pose of the object [2]–[4]. This method is limited to the feature selection of the object, and is not robust to background changes and object changes. the result is unsatisfactory. Another method uses the deep learning meth odto directly predict 6DoF pose of the target object [5]–[7]. The method assumes that the space is huge, while it is difficult to obtain the accurate solution from this hug space. This method is difficult to apply to Automatic Collection Trolley Robot System. In addition, the method based on keypoints detection is divided into two steps [8]. Firstly, the 2D keypoints of the target object are detected, and then the pose of the target object is estimated by the 2D-3D PNP algorithm according to the corresponding relationship between the 2D keypoints and the object 3D keypoints in model [9]. Accurate keypoints detection algorithm can obtain more accurate pose of the target object [10]. It may affect the detection of keypoints that the method uses the whole image as input with complex background.

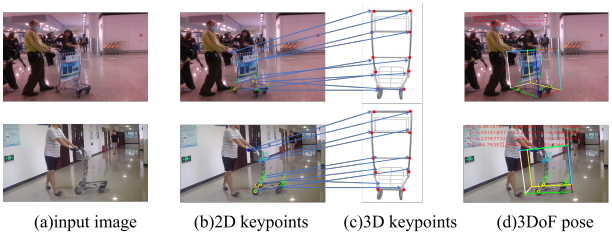

Fig. 1. pipline for pose estimation. (a)input a single RGB image. (b)predicted 2D keypoints for trolley. (c)The real 3D keypoints for Trolley. (d) Result: 3DoF pose for trolley.

However, the keypoints of trolley is under occlusion frequently in different pose. Mentioned above methods are sensitive to occlusion and the existing methods do not solve these problems well

In this paper we proposed a 3DoF pose estimation method for multi-trolley based on the keypoints. This paper adopts three steps to realize the 3DoF pose estimation for multi-trolley. The steps are as follows: Firstly, the trolley is detected by YOLOV3 [1]and is cut out individually. Secondly, the cropped image is taken as input to 2D keypoints predicting model and the model predicts the 2D keypoints of the trolley. Finally, we fine-tune the 2D keypoints obtained from second step by post-processing. Then we obtain the 3DoF pose for

[1]Xiaomeng Liu, Wenbo Han, Shuang Song are with School of Mechanical Engineering and Automation, Harbin Institute of Technology(Shenzhen), Shenzhen, China, 518055.

[2]Max Q.-H. Meng is with Department of Electronic Engineering, the Chinese University of Hong Kong, Hong Kong, China, and affiliated with the State Key Laboratory of Robotics and Systems (HIT), Harbin Institute of Technology, China.

*Corresponding author: Shuang Song, mail: songshuang@hit.edu.cn

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

the trolley by PNP, according to the correspondence between the 2D keypoints and the real 3D keypoints.

The first step is to detect trolley from a single RGB image and crop it individually. The single-stage object detection model called yolov3 is used to identify the trolley in the picture. According to the size of the box, the trolley is cropped individually with a ratio of 1:1.5.

The second step predicts the 2D keypoints of the trolley. We select 10 2D keypoints of the trolley and proposed a new network model to identify the 2D keypoints of the trolley. The network includes two parts (DetectNet and HardNet). The first stage called DetectNet is mainly used to identify easy keypoints, while the second stage called HardNet is mainly used to identify hard keypoints that includes the keypoints under occlusion.

The third step is aimed to fine-tune the predicted 2D keypoints according to Geometric constraint relationship of the trolley real 3D model and also estimate the 3DoF pose for the trolley based on the correspondence between the 2D keypoints and the 3D real model by the 2D-3D PNP algorithm.

Structure of this paper is organized as follows:

II. RELATED WORK

The 3DoF pose estimation from a single RGB image is much more challenging due to the lack of depth information. Scholars have also conducted in-depth research on this issue.

Traditional methods: the traditional method uses template matching to estimate the pose, [2] proposed an image invariance, rotation and translation invariance characteristics. Through the image gradient calculated in multiple direction planes and multiple scales, the image feature points of the local geometric deformation are created. Then the target feature points are matched with these feature points, which identify if the predicted object is the target object. In addition, [3] extract the sift and ferns features of the object with strong texture features, while estimating the pose of the object. This method is mainly for objects with strong texture features, and is not suitable for the hollow trolley. Different from the above two methods, [4] firstly used the gPb algorithm to obtain the edge information of the object, and then minimized distance between the target 3D model edge and the image contour projection by least squares method to obtain the final target pose. Traditional methods are limited to feature selection of objects and are not robust to background changes and object changes.

CNN based methods: Thanks to the power of the neural network, current work attempts to directly regress the 6DoF pose of the target object using a convolutional neural network. For example, [5] used a convolutional neural network to estimate the pose of the camera, and directly output the pose of the camera. Due to the lack of depth information and the huge search space, the result is unsatisfactory. For the problem that arises in [5], [6] improves it. Different from directly regressing pose, [6] first predicted the distance from the object to the camera and obtained the actual 3d coordinates of the target object and then regress the rotation matrix of the object. Because of the need to regress the rotation matrix and the huge search space, it is difficult to achieve convergence. In addition, [7] improves on the basis of the target detection algorithm SSD [8] for predicting the box, viewpoint and rotation matrix of the target object followed by optimizing the predicted pose. However, this method needs to optimize the regressed pose with time consuming growing. Due to the lack of depth information and the search space is huge, and it is very difficult to directly regress the pose of an object based on the neural network.

Keypoints based method: these methods are based on the keypoints, which can be divided into two steps. The first step is to detect the 2D keypoints from a single RGB image. The second step is to calculate the 6DoF pose of object by PNP algorithm, according to the corresponding relationship between the 2D keypoints and the real 3D keypoints in 3D model. [9] used convolutional neural network to directly predict the coordinates of the 2D keypoints on the image from the 3D model box corner projection, and then calculated the 6DoF pose of the object through pnp. Training this model is much more challenging due to the picture similarity of the objects in different poses. In order to solve these problems, [10] proposed to use the heatmap method to predict the keypoints of the object. Then it crops multiple picture blocks that are be as the input of the network, while directly output the probability the heatmap for predicting each keypoint. As result, it selects the coordinate point of the maximum probability as the target object keypoints. In addition, in order to solve the problem of occlusion, a pixel point voting strategy is proposed in [11], that is, each pixel points predicts a vector pointing to a keypoint. according to the idea of the RANSAC algorithm, the final keypoint is predicted. However, [11] needs to segment the area of the target object, which is not suitable for the hollow trolley.

III. PROPOSED APPROACH

The 6DoF pose of a 3D object can be represented by $[x, y, z, \theta, \varphi, \phi]$ where $[x, y, z]$ are the location of the object and $[\theta, \varphi, \phi]$ are the deflection angle called pitch, yaw, roll. Since the robot and the trolley are on the same plane, we only need to estimate three parameters $[x, y, \varphi]$ that can make robot reach right location and orientation. Therefore, it is our task to obtain 3DoF pose of the trolley.

In this paper, we propose a 3DoF pose estimation method for multi-trolley based on the keypoints. It can be divided to three steps, shown in Fig. 2. As shown in the Fig. 2, it includes trolley detection, keypoints detection, post-processing and PNP [12].

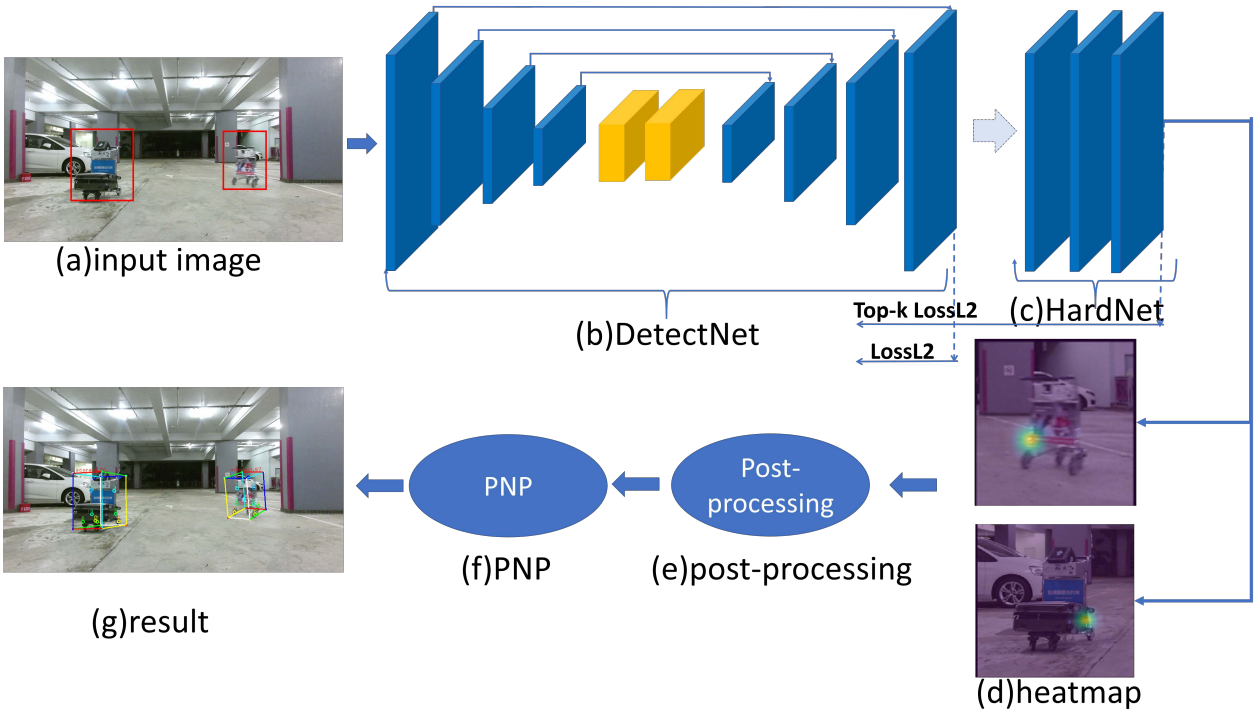

Fig. 2. overview for pose estimation. (a)input image. (b)the structure of DetectNet for detecting easy keypoints. (c) the structure of HardNet for detecting keypoints under occlusion. (d) the heatmap [21] for predicted keypoints. (e) post-processing for fine tuning the predicted keypoints. (f) PNP for calculating the 3DoF pose. (g)the result.

Fig. 3. pipline for pose estimation.

A. *Trolley Detection*

The object detection algorithm is mainly divided into two methods including single stage and two stages. The single-stage algorithm (e.g. [1], [13]–[16]) directly outputs the detection result and it is fast, but the precision is relatively low. In contrast, the two-stage algorithm (e.g. [17]–[20]) needs to select the candidate region firstly and then classify and regress the candidate region, etc., So it is slow, but the precision is relatively high. Since trolley detection is only an initial forecasting process, we directly use the yolov3 algorithm with fast speed and accuracy for meeting the requirements to detect the trolley.

The trolley is detected by the yolov3 algorithm followed by cropping out from the original image individually according to the ratio of the box size of 1:1.5, shown in Fig. 4. The prediction of keypoints on the cropped trolley can not only eliminate the influence of complex background and improve the accuracy of recognition, but also handle with multi-trolley individually.

Fig. 4. trolley detection.

B. *Keypoints Detection and Post-processing*

Selection of keypoints: The model of the trolley is shown in Fig. 5. The keypoints of the trolley are selected according to two principles:

1. Ensure that the selected keypoints are as far as possible and remain in different planes.

2. The selected key points must have strong features.

As shown in Fig. 6. Based on the above two points, the following 10 points are selected as the keypoints of the trolley. In addition, each keypoint has its own semantic information and marks unique serial numbers.

Keypoint model structure: The network structure of this paper consists of two parts, DetectNet and HardNet, shown in Fig. 3. The difficulty of identifying different key points is different. For example, the keypoint under occlusion is

Fig. 5. trolley 3D model.

Fig. 6. the keypoints of trolley.

harder predicted than others without occlusion. DetectNet is responsible for the identification of simple key points, while HardNet is responsible for the identification of difficult key points, that is, only the loss of top-k is selected for back propagation, making the model more concerned Identification of hard keypoints.

DetectNet is a structure of an encoder-decoder. The encoder uses ResNet-18 as the backbone, with a total step size of 8. In order to ensure that the resolution of the feature map is no longer reduced and the large receptive field is maintained when the network is deepened, the dilation convolution is used instead of the standard convolution.

$$W_i = \left(W_{i-1} + 2 \times p - d \times (k-1) - 1\right) / \text{stride} + 1 \quad (1)$$

$$R_i = (R_{i-1} - 1) \times \text{ stride } + d \times (k-1) + 1 \quad (2)$$

where W_i represents the size of the i-th feature map, p represents padding, and d is the dilation ratio. In addition, k represents the convolution kernel size and stride represents the step size and R_i represents the receptive field of the i-th layer.

For example, the receptive field is the same for dilation=1, k=3, stride=2 and dilation=2, k=3, stride=1. The size of the former feature map does not change by the dilation convolution. In addition, the Decoder adopts the quadratic interpolation method for up-sampling, and finally outputs the feature map of w*h*10. Each feature map represents the prediction result of a keypoint. Then the maximum value is selected as the final prediction result of the keypoint, while L2-Loss is used as a loss function of DetectNet to measure the difference between the predicted map and the real map.

$$\text{Loss } = \sum_{j=1}^{10} \sum_{P} \left\| S_j(p) - S_j^*(p) \right\|^2 \quad (3)$$

Where p represents the coordinates of the pixel, $S_j(p)$ is the predicted value of the j-th keypoint at p, and $S_j^*(p)$ represents the true value of the j-th keypoint at p.

$$S_j^*(p) = \exp\left(-\left\| p - x_j \right\|^2 / \sigma^2\right) \quad (4)$$

Where x_j represents the position (x, y) of the j-th keypoint of the annotation. And σ controls the slope of the Gaussian peak. The true map value is normally distributed centered on the position of the keypoint. And the closer the distance is to the position of the keypoint, the larger the value, and the smaller the distance is, the smaller the value.

HardNet is two standard convolutional layers used to fine tune the previous network structure. DetectNet is used to identify simple keypoints, and HardNet focuses on the identification of difficult key points. Inspired by the hard example online mining and FocalLoss method, HardNet only selects the top-k loss for back propagation during the training process and make the network biased to learn hard keypoints. for example, the keypoints under occlusion.

Post-processing: The trolley is a symmetrical rigid body, shown in Fig. 7. We add a post-processing module to complement the key points not detected by the previous model based on the geometry of the trolley. It's easy to see that the lines between the keypoints in the image are always in a parallel relationship and the scale is the same.

According to the above characteristics, Firstly, we determine the length d_i^* of the real parallel line. Next, we calculate the length d_i and the off-angle $_i$ of the parallel lines existing in the prediction result and Simultaneously record the score s_i (mean value of predicted scores of two keypoints) of each predicted parallel line segment. So, it is our task to complete the undetected keypoints based on the detected keypoints.

$$d_i = \left\| x_{i0} - x_{i1} \right\| \quad (5)$$

$$\theta_i = \text{acos}\left(unit \cdot (x_{i1} - x_{i0}) / d_i\right) \quad (6)$$

$$\theta = \sum \left(s_i / \sum_i^n s_i \right) \theta_i \quad (7)$$

Fig. 7. parallel lines in trolley.

$$k = \sum \left(s_i / \sum_i^n s_i \right) d_i / d_i^* \tag{8}$$

$$x_{j1} = x_{j0} + k d_j^* \cos\theta \tag{9}$$

Where here d_i is the length of the predicted i-th parallel line, and x_{i0},x_{j1} represent the coordinates of the two endpoints that make up the parallel line. θ_i represents the angle between the i-th line and unit (0,1). In addition, θ represents the direction of the line with the keypoint, and d_j is the length of the j-th line where the keypoint is to be completed. Finally, $x_j 1$ that was not predicted by model is will to be completed.

C. PNP Matching

According to the 3D model of the trolley, the 3D coordinates of 10 keypoints are determined based on the 3D trolley model origin. Finally, we obtain the 6DoF pose for the trolley based on the correspondence between the 2D keypoints and the 3D real model by the 2D-3D PNP algorithm.

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{10}$$

where u and v are 2D keypoint coordinates, and $[x,y,z]$ are 3D keypoints coordinates in the trolley model frame, $[fx,fy,u0,v0]$ are camera intrinsic parameters that can be obtained via camera calibration. In practice, the 3DoF pose including R and T parameters can be obtained by PNP algorithm. Once R and T are obtained, we can calculate the 3DoF pose $[x,y,]$ of the trolley by simple transformation.

IV. EXPERIMENTS

A. Dataset

The data set has a total of 15430 and the image resolution is 640*480. The data set includes raw data (9260) and enhanced data (6170), and the data enhancement mainly uses operations such as inversion, cropping, and scaling. The data set is divided into a training set and a test set, a training set 14130, and a test set 1300.

In addition, the collection and labeling of data sets is a very time-consuming task, shown in Fig. 8. Therefore, this paper proposes a simple and effective method to automatically expand the data set. First, we extract the trolley template with marking the size of each template (w, h), and extract the mask corresponding to the template. Next, we select the appropriate image as background, and insert the template in picture position (x, y). Finally, we obtain a new artificial data with the trolley, marked as (x, y, w, h).

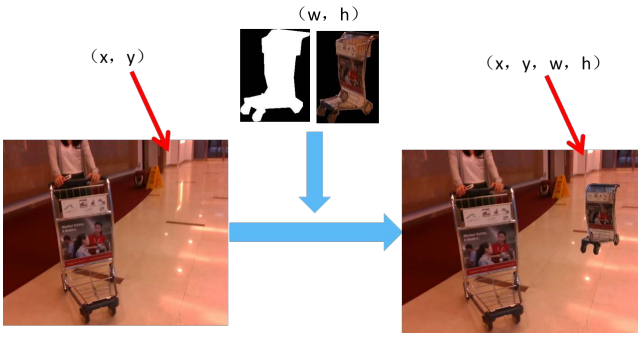

Fig. 8. automatically expanding the data.

B. Evaluation

In this paper, we use the OKS [21], [22] indicator to evaluate our model. OKS was proposed in the AI CHALLENGE human pose detection competition to measure the similarity between the predicted keypoints and the real keypoints. OKS is similar to the IOU in object detection task. In addition, the OKS is limited to 0-1. when the OKS is larger, it represents the more accurate. In contrast, the smaller the OKS, the larger the prediction error.

$$OKS_i = \sum_j \exp\left(-d_{ij}^2 / 2S_i^2 \sigma_j^2\right) \delta\left(v_{ij} = 1\right) / \sum_j \delta\left(v_{ij} = 1\right) \tag{11}$$

Where OKS_i represents the oks value of the i-th trolley, d_{ij}^2 is the Euclidean distance between the predicted keypoint and the real keypoint, In addition S_i^2 represents the size of the i-th trolley, and σ_j^2 is error of the label data and its value is the standard deviation of the j-th keypoint, v_{ij} represents the visibility of the marked key point. if v_{ij} is 0, we will not calculate the OKS for that keypoint.

$$mAP = mean\{AP@(0.50:0.05:0.95)\} \tag{12}$$

Finally, the average accuracy mAP is used as the evaluation index of the final model. The above is a com-

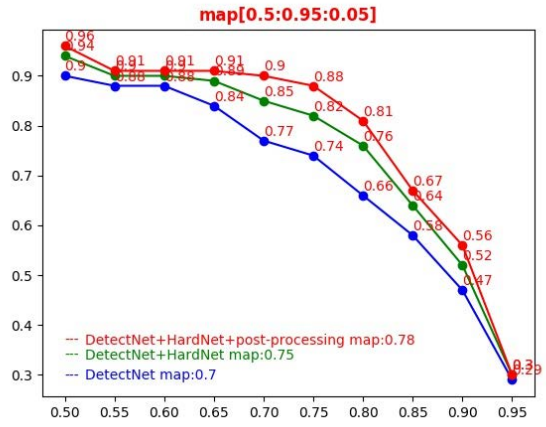

Fig. 9. comparison of the results.

parison of the results of DetectNet, DetecNet+HardNet, DetecNet+HardNet+post-processing, shown in Fig. 9. The DetectNet model is 5 points lower than the DetectNet+HardNet model, indicating that adding the HardNet layer can increase the identification of difficult key points under occusion. In addition, adding post-processing is helpful to obtained the keypoints and the result is satisfactory.

V. CONCLUSION

In summary, in this paper we proposed a 3DoF pose estimation method for multi-trolley based on the keypoints from a single RGB image. Our method is accuracy and robust for estimating the pose of the trolley under the problem of occlusion and low-resolution. The result shows that DetectNet+HardNet model followed by post-processing are effective for predicting the keypoints of trolley. In the future, we will simplify our model and apply it to Automatic Collection Trolley Robot System.

REFERENCES

[1] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[2] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features." in *iccv*, vol. 99, no. 2, 1999, pp. 1150–1157.

[3] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proceedings of the 7th IEEE/ACM international symposium on mixed and augmented reality*. IEEE Computer Society, 2008, pp. 125–134.

[4] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis, "Single image 3d object detection and pose estimation for grasping," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3936–3943.

[5] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[6] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[7] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1521–1529.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[9] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3828–3836.

[10] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3d object pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–134.

[11] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[12] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[14] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast yolo: a fast you only look once system for real-time embedded object detection in video," *arXiv preprint arXiv:1709.05943*, 2017.

[15] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[18] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[19] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[21] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[22] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 483–499.