The following publication Fan, J., Li, S., Zheng, P., & Lee, C. K. (2021, August). A High-Resolution Network-Based Approach for 6D Pose Estimation of Industrial Parts. In 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE) (pp. 1452-1457). IEEE is available at https://doi.org/10.1109/CASE49439.2021.9551495

A High-Resolution Network-Based Approach for 6D Pose Estimation of Industrial Parts

Junming Fan[1,2], Shufei Li[1], Pai Zheng[1,2*], Carman K.M. Lee[1,2]

***Abstract*— The estimation of 6D pose of industrial parts is a fundamental problem in smart manufacturing. Traditional approaches mainly focus on matching corresponding key point pairs between observed 2D images and 3D object models via hand-crafted feature descriptors. However, key points are hard to discover from images when the parts are piled up in disorder or occluded by other distractors, e.g., human hands. Although the emerging deep learning-based methods are capable of inferring the poses of occluded parts, the accuracy is not satisfactory largely due to the loss of spatial resolution from multiple downsampling operations inside convolutional neural networks. To overcome this challenge, this paper proposes a 6D pose estimation model consisting of a pose estimator and a pose refiner, by leveraging High-Resolution Networks as the backbone. Experiments are further conducted on a dataset of industrial parts to demonstrate its effectiveness.**

I. Introduction

The recent advances in smart manufacturing and human-robot collaboration [1], [2] raise the demand for precise part location and pose estimation to meet the efficiency and flexibility requirements. For instance, robots need to be able to consistently recognize objects of interest to cope with uncertainties introduced by human collaborators or flexible production.

Traditional applications of pose estimation of industrial parts mainly rely on tactile sensors. Saund et al. [3] proposed to localize parts by touching the part at different points with a probe on a robotic arm and introduced a particle filter-based algorithm to tackle the pose estimation problem. In [4] the authors introduced an optimization algorithm that uses tactile and force sensors to optimize the pose estimation provided by vision methods. Despite high precision in certain applications, these physical contact-based methods might fail in scenarios such as narrow spaces or complicated structures.

Vision-based methods, including hand-crafted feature-based and deep learning-based approaches, have also been widely adopted in industrial part localization for high-level flexibility and efficiency. The former ones can work efficiently in highly organized workshops while suffering from rigid requirement of light sources, such as the approaches of speeded up robust features (SURF) [5], background subtraction [6], and so on. Deep learning-based methods, especially convolutional neural networks (CNN), are endowed with better effectiveness and robustness. Nguyen et al. [7] utilized MobileNetV2 [8] to identify positive object patches from images for robot grasping. Object detection models such as region-based CNN (Faster R-CNN [9], Mask R-CNN [10]) were adopted for localizing industrial objects in [11]–[13]. Nevertheless, these methods only provide 2D information which is not suitable for complex tasks such as human-robot handover, where 3D information of objects is indispensable.

To make the leap from 2D to 3D, researchers formulated the problem as 6 degree-of-freedom (6D) pose estimation, referring to estimating the translation and rotation of objects in 3D coordinates given the 3D object models. Existing 6D pose estimation researches mainly have two perspectives: feature-based and template-based. Feature-based methods aim at matching corresponding key points based on the extracted features between the observed 2D image and the given 3D object model [14], [15], and the object pose parameters are further calculated from the correspondence. As for template-based methods, the object poses are obtained by calculating the similarities between the observed image and a set of pre-defined templates, and the most similar template represents the object pose [16]–[18]. However, these methods cannot properly handle heavy occlusion or textureless objects, in which case feature points are incomplete or missing.

Recent studies of CNN-based 6D pose estimation has shown promising results [19]–[22]. These methods manage to mitigate the suffering from occlusion and textureless objects by learning to predict pose parameters in a data-driven manner without explicit modelling of feature correspondence or template similarity. To further improve the pose estimation accuracy, some studies decided to borrow the idea from feature points-based methods and predict the correspondence of key points via CNN models [23], [24], while others added an extra refinement stage upon the CNN-based 6D pose estimation model [25], [26].

Although CNN-based approaches have made some significant improvements in object 6D pose estimation, there are still some problems in this field. One critical issue is that subtle pose differences can be easily neglected since the backbone networks are usually borrowed from renowned classification models such as VGGNet [27] and GoogLeNet [28], which leverages a gradually downsampled structure to obtain a more compact representation with higher semantic information that could facilitate object classification or detection, but lower spatial resolution which is not ideal for tasks

This research is funded by the Laboratory for Artificial Intelligence in Design (Project Code: RP2-1), Hong Kong Special Administrative Region.

[1]J. M. Fan, S. F. Li, P. Zheng and C.K.M Lee are with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong Special Administrative Region (e-mail: jun-ming.fan@connect.polyu.hk, shufei.li@connect.polyu.hk, ckm.lee@polyu.edu.hk. *Corresponding author phone: +852-2766-5633, email: pai.zheng@polyu.edu.hk).

[2]J. M. Fan, P. Zheng and C.K.M Lee are also with Laboratory for Artificial Intelligence in Design, Hong Kong Special Administrative Region.

that require more subtle spatial features such as semantic segmentation and 6D pose estimation.

This study suggests that the task of 6D pose estimation not only requires strong semantic representations to recognize the object, but also precise spatial features to distinguish small variations of object pose. Aiming at exploiting the advantage of high resolution features, in this study, High-Resolution Network (HRNet) [29] is leveraged as the backbone network, upon which a model is constructed that directly predicts 6D pose parameters from RGB image and its corresponding depth image (RGB-D). It is hard to achieve accurate pose estimation through a single shot, hence a refining stage is additionally adopted. Concretely, the predicted 6D pose parameters from the first stage would be utilized to render the 3D object model to its estimated pose, and the rendered RGB-D image is concatenated with the original image and sent to a refining network to predict the difference between the coarse estimation result and ground-truth pose.

The rest of the paper is organized as follows: Section II and III will explain the proposed method in detail and show its effectiveness with experimental results respectively. Section IV will summarize the main contributions of this work and highlight the future directions.

II. Methodology

In this section, the proposed high-resolution network-based 6D pose estimation method is explained in detail. Provided with the observed RGB-D image and 3D object model, the objective of 6D pose estimation is to infer the object pose parameters, which are normally presented as a SE(3) transformation (SE: Special Euclidean Group) consisting of a 3-DoF (Degree of Freedom) rotation $\mathbf{R}$ and a 3-DoF translation $\mathbf{t}$. With the estimated $\mathbf{R}$, $\mathbf{t}$ and the object's 3D model, the complete 3D information of the object can be well obtained.

A. Architecture Overview

The overall architecture of the proposed model for 6D pose estimation of industrial parts is illustrated in Fig. 1. The architecture mainly consists of three stages, i.e., industrial parts detection, coarse pose estimation, and pose refinement. The detection stage takes RGB image as input, and output parts bounding boxes and classification results. The detection stage can be regarded as a preprocessing step and the following pose estimation stages are actually agnostic to which specific detection model is used, so this study simply adopts Faster R-CNN [9] as the detector. Then in the coarse pose estimation stage, an HRNet-based pose estimation network is constructed to better distinguish small pose differences of industrial parts by taking advantage of high-resolution features. The coarse pose estimation network takes the cropped RGB-D patch of the detected part as input, and estimates the rotation $\mathbf{R}$ and translation $\mathbf{t}$ respectively. The final stage is designed for pose refinement, which first generates a rendered RGB-D image based on the estimated coarse pose parameters and the object 3D model, then concatenate the rendered image and the cropped image together as the input, and estimate the pose deviations $\mathbf{\Delta R}$ and $\mathbf{\Delta t}$. By applying the predicted $\mathbf{\Delta R}$ and $\mathbf{\Delta t}$ to the coarse $\mathbf{R}$ and $\mathbf{t}$, the final 6D pose estimation is obtained.

B. Industrial Parts Detection

The first step of this work is to extract the regions of interest for industrial parts. Concretely, the observed image might contain multiple industrial parts, for which the individual regions and categories need to be extracted first to facilitate subsequent 6D pose estimation. For a specific part in the image, the image patch will be cropped according to the detection results which are normally represented as bounding boxes. Then the following pose estimation processes only need to consider the cropped image area, which brings two benefits: 1) The removal of irrelevant image area could ease the model training process; 2) Better computational efficiency. Meanwhile, the part category given by the classification results will be used to decide which part model should be applied to the rendering process in the pose refinement stage. Therefore, a successful detector Faster R-CNN is employed to locate industrial parts in complex industrial scenarios, regardless of occlusion and textureless interference. Under this prerequisite, the work can pay more attention to the following two-stage pose estimation.

C. Coarse Pose Estimation

Instead of segmenting each object with extra branches [20], this study simply crop the part region from the observed image as input of the pose estimation model. Based on the cropped RGB-D image, the pose estimation model can directly regress the part pose parameters including the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ of the part to fully recover its pose in 3D space. The details are illustrated as follows. d image as input of the pose estimation model. Based on the cropped RGB-D image, a pose estimation model will directly regress the part pose parameters. This section illustrates the details of the pose estimation model as follows.

1) Depth Image: The introduction of depth image plays an essential role in our pose estimation model. Existing methods such as [20], [25] tend to tackle the 6D pose estimation problem only from RGB images, which could bring uncertainties for the translation estimation. Without depth information, a traditional pose estimation model has to memorize the size of a specific object for coordinate transformation, which can be further utilized to transform each object from 2D image pixels to 3D space. This is not only difficult for the model to learn, but also prone to error when facing similar objects with different actual sizes, which is often the case for industrial parts such as bolts. Although [20] has explored to tackle this problem with ICP refinement algorithm, it is infeasible for real industrial applications due to the time-consuming computation.

To avoid the above issue, this study simply attaches depth image as an extra channel to the RGB image to form an RGB-D image, which is further input to the pose estimation model for processing.

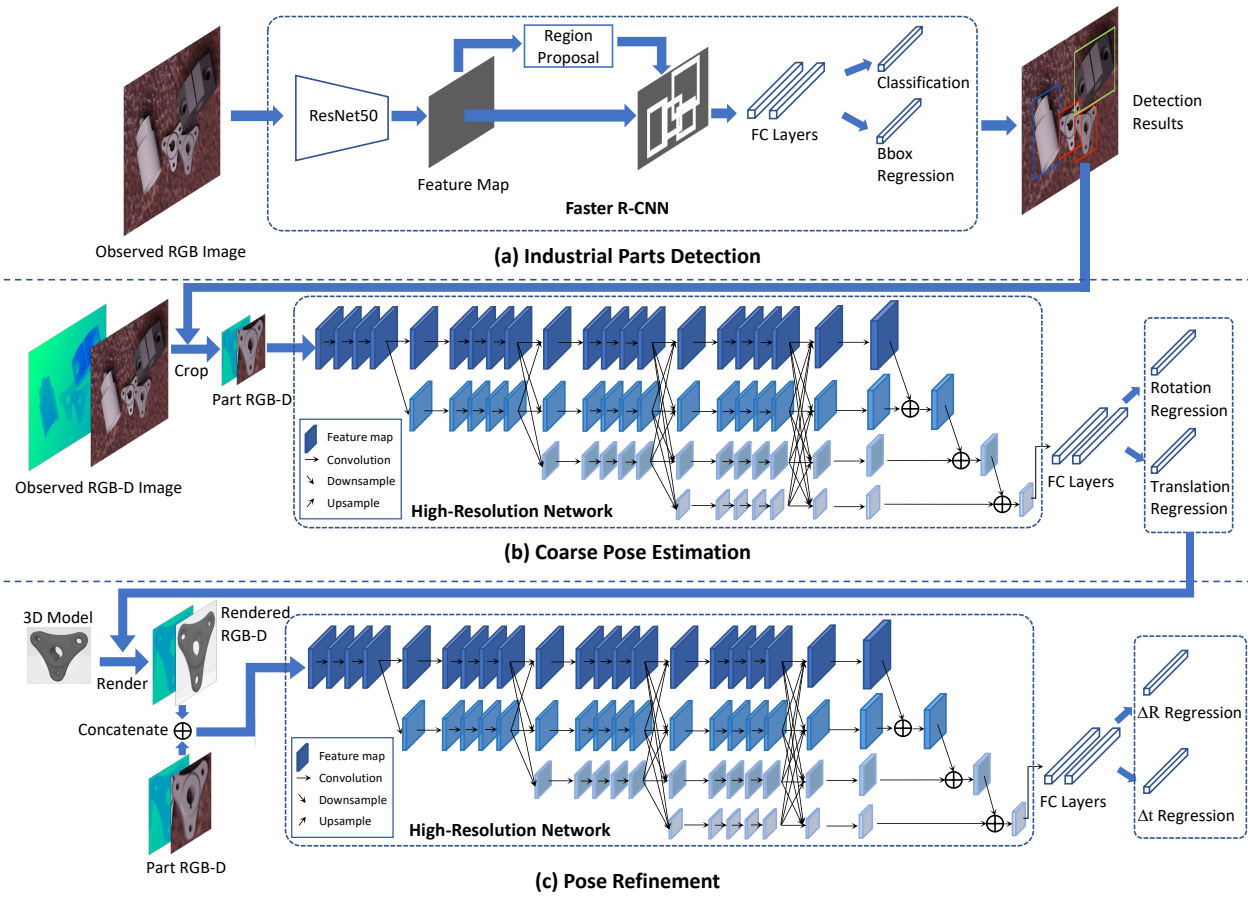

Fig. 1. Overall architecture.

2) High-Resolution Feature Extraction: A critical limitation that hinders current methods [19], [20] of 6D pose estimation is that deep neural networks are prone to lose spatially feature representation after the gradually downsampling operations, e.g., VGGNet [27] and GoogLeNet [28]. Inspired by [29], this study adopts the backbone network design of High-Resolution Networks, which can ensure both spatially precise and semantically strong feature representation for 6D pose estimation. The major differences between classification network design and high-resolution network design are shown as Fig. 2. While normal deep learning networks quickly decrease the feature map size, high-resolution network maintains the spatial resolution throughout the process.

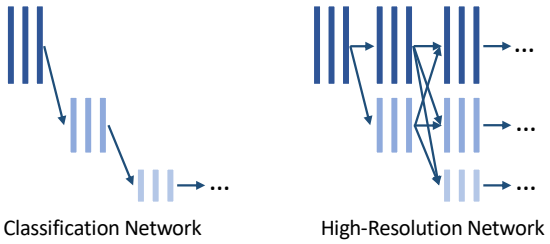

Fig. 2. Backbone comparison.

The input to the network has the shape $4 \times H \times W$, where 4 represents the 4-channel RGB-D image, H means image height and W means image width. The first two convolution layers have 3×3 kernels and the strides are 2, after which the feature map resolution is decreased to $\frac{H}{4} \times \frac{W}{4}$.

The main body of the network consists of several parallel branches with different spatial resolutions. As Fig. 1 (b) shows, the uppermost branch maintains the resolution $\frac{H}{4} \times \frac{W}{4}$ until the final fusion, while gradually lower-resolution branches are added to the network one-by-one with $\frac{1}{2}$ of resolution of the previous branch until there are four branches with different resolutions.

In the middle of the model, there are interleaved connections between the parallel branches every few convolutions. It is commonly acknowledged that low-resolution feature maps represent semantic information better while high-resolution feature maps contain more precise spatial information. The interleaved design is leveraged to better fuse and exchange information between multiresolution branches.

At the final part of the network, feature maps from higher-resolution branches are first downsampled by $\frac{1}{2}$ and concatenated with the ones from lower-resolution branches. This process repeats until all the features are squeezed into the final feature maps, which are further processed by 2 fully connected (FC) layers sequentially. Finally, the translation

parameters are estimated by an FC layer with 3 neurons and the rotation parameters are estimated by another FC layer with 4 neurons.

3) 6D Pose Estimation: With the definition of input format and network structure in previous parts, the model is ready to do forward inference. To be able to train the model, a loss function is required to represent the prediction error. In this part, the pose parameterization is first introduced and then the loss function is defined based on the estimated pose parameters.

The 6D pose is represented by a rotation $\mathbf{R}$ and a translation $\mathbf{t}$. Let $\mathbf{t} = (t_x, t_y, t_z)^T$ be the translation vector of the object, where t_x and t_y represent the object center in the image coordinates and t_z the average distance from the object to the camera. Here t_x and t_y are actually the pixel deviations from the left-top corner of the cropped image patch to the center of the object for the convenience of implementation. The actual position could be easily obtained by combining this representation and the bounding box coordinates of the object from the detection stage. And the loss function for translation regression is defined as:

$$L_t(\hat{t}, t) = \begin{cases} 0.5(\hat{t} - t)^2 & \text{if } \left|\hat{t} - t\right| < 1 \\ \left|\hat{t} - t\right| - 0.5 & otherwise \end{cases}, \tag{1}$$

where $\hat{t}$ denotes the ground truth translation and t denotes the estimated translation. Notice that this is actually the smooth-L1 loss function [9], which is differentiable at 0.

Following existing work [20], the rotation $\mathbf{R}$ is represented using a quaternion $\mathbf{q} = q_r + q_i\mathbf{i} + q_j\mathbf{j} + q_k\mathbf{k}$ as follows:

$$\mathbf{R} = \begin{bmatrix} 1 - 2\left(q_j^2 + q_k^2\right) & 2\left(q_i q_j - q_k q_r\right) & 2\left(q_i q_k + q_j q_r\right) \\ 2\left(q_i q_j + q_k q_r\right) & 1 - 2\left(q_i^2 + q_k^2\right) & 2\left(q_j q_k - q_i q_r\right) \\ 2\left(q_i q_k - q_j q_r\right) & 2\left(q_j q_k + q_i q_r\right) & 1 - 2\left(q_i^2 + q_j^2\right) \end{bmatrix}, \tag{2}$$

which is easier for the model to learn than naive rotation angles. And the loss function for rotation regression is defined as:

$$L_R(\hat{R}, R) = \frac{1}{N} \sum_{i \in N} \min_{j \in N} \left\| \hat{R}x_i - Rx_j \right\|^2, \tag{3}$$

where x_i denotes the i^{th} point of N points of the 3D object model, x_j the j^{th} point, $\hat{R}$ the ground truth rotation, and R the estimated rotation. The basic idea is to apply the ground-truth rotation and estimated rotation to a point of object model and calculate the L2 distance. But for symmetrical objects, different rotation angles might result in the same appearance, which cannot be represented well by simply taking the L2 distance of applying rotation matrices to a same point. So this loss function instead measures the distance between a point with the estimated rotation and the closest point with the ground-truth rotation.

The overall loss function is simply defined as the sum of the previous two loss functions:

$$L_{overall} = L_R + L_t \tag{4}$$

D. Pose Refinement

To improve the pose estimation accuracy, a pose refinement stage is introduced in this study. The goal is to predict the pose estimation error of the coarse pose estimation stage. To achieve this goal, the estimated pose from the previous stage is first applied to the 3D object model to obtain a rendered RGB-D image, which is then concatenated with the cropped RGB-D image same as the input of the coarse pose estimation stage to form an 8-channel tensor as the input of pose refinement model.

The backbone network of pose refinement takes the same design as the coarse pose estimation stage. Although this is not compulsory, this paper utilizes the same backbone model for the convenience of implementation and also exploiting the advantage of high-resolution feature representation.

The output format is also highly similar to that of the coarse pose estimation stage. The only difference is that the estimated target is the relative pose error $\mathbf{\Delta R}$ and $\mathbf{\Delta t}$ rather than absolute pose parameters. Applying the estimated pose error to the coarse pose, the final pose is obtained as follows:

$$\mathbf{R_{final}} = \mathbf{\Delta R R}, \tag{5}$$

$$\mathbf{t_{final}} = \mathbf{t} + \mathbf{\Delta t}, \tag{6}$$

where $\mathbf{R_{final}}$ and $\mathbf{t_{final}}$ represent the final rotation and translation. The loss functions for pose refinement also takes the same form as in coarse pose estimation but replacing $\mathbf{R}$ and $\mathbf{t}$ with $\mathbf{R_{final}}$ and $\mathbf{t_{final}}$ respectively.

III. Experiments

A. Dataset

To demonstrate the effectiveness of the proposed model, experiments are conducted on an industrial 3D object dataset—MVTec ITODD [30], which is specifically designed for industrial parts aiming to facilitate industrial applications. The dataset contains 28 types of industrial parts with over 50,000 labeled samples that are synthesized via a physics-based renderer. Considering that the synthetic data is only randomly generated by sampling from a particular set of parameters and the computational resource constraints during the experiment process, we decided to randomly choose 5,000 samples from the original dataset as the new training set and 1,000 as the new test set. Although 5,000 samples seem to be insufficient for training large CNN models, in the experiment we found it is acceptable since the whole distribution space is covered via random sampling. The following experiments and results are all based on this new division.

B. Experimental Setup

The whole model is implemented using PyTorch [31]. And the experiments are conducted using a single Nvidia RTX2060s GPU as the acceleration device.

The Faster R-CNN detector is first trained separately using the official implementation of torchvision as a part of the PyTorch library. During the training process of the

Faster R-CNN detector, the pretrained weights provided by PyTorch are leveraged to initialize the model, which enables the detector to adapt to the relatively small training set more easily and mitigate the overfitting problem. The backbone network of Faster R-CNN is ResNet50 [32], which is renowned for its ability to deal with vanishing gradient via residual connections. Other parameters related to the detector remained their default values. The training process of the detector took about 19 hours. After the detector is trained, the weights are fixed for the rest of the whole experiment process.

As for the coarse pose estimation and pose refinement models, the layers in the backbone are initialized with the weights pretrained on ImageNet [33], and other layers are randomly initialized. Adam optimizer with initial learning rate 0.001 is utilized in model training. Training batch size is set to 8 for the two pose estimation stages. The training code is developed based on the code published by Labbe et al. [34]. For more detailed parameter settings, please refer to the original code base. The training process lasts for 500 epochs, which requires about 24 hours for each stage.

C. Results

The performance of the proposed model is evaluated on the new test set. Following [20], the average distance with symmetrical objects (ADD-S) is leveraged as the evaluation metric:

$$ADD-S = \frac{1}{N}\sum_{i\in N}\min_{j\in N}\left\|(\hat{R}x_i+\hat{t})-(Rx_j+t)\right\|^2, \quad (7)$$

which basically takes the same idea as the loss function for rotation regression. While previous work normally choose a predefined distance threshold to calculate the percent accuracy, this study directly reports the average distance calculated by equation (7) for simplicity.

Table I presents the evaluation results comparison between baseline model and the proposed model. The baseline model was proposed in [34], which had the best performance on the utilized dataset, and only the single-view model is adopted in the experiments of this work because the dataset only contains single-view data. The main differences between the baseline model and the proposed model lie in the backbone design and depth image usage. Concretely, the baseline model uses EfficientNet-B3 [35] as the backbone network, which is the state-of-the-art classification model on ImageNet, and the baseline model only takes RGB image as input without depth channel.

TABLE I
EVALUATION RESULTS COMPARISON

Method	Backbone	Depth	Refinement	ADD-S
Baseline [34]	EfficientNet-B3	w/o	w/o	0.0636
		w/o	w/	0.0252
Proposed	HRNetV2-W32	w/	w/o	0.0442
		w/	w/	**0.0163**

As Table I depicts, the proposed model with refinement has the smallest ADD-S distance 0.0163 on the test set. Comparing with the baseline model, the proposed model performs better with or without the refinement stage, suggesting that the introduction of high-resolution model design and depth image significantly improves the performance. Note that the reported results of the baseline model are different from the original paper because the experimental setup and the evaluation metric are different. To make a fair comparison, both models were trained under the same setting as section III-B shows. In terms of evaluation time for a single sample, the baseline model requires 0.69s on average for the whole process but varies from 0.4s to 0.9s depending on the number of objects presented, while the proposed model is about 0.2s slower on average.

Fig. 3 demonstrates some examples of 6D pose estimation results. For each pair, the left picture is the input image and the right one represents the rendered image according to the estimated 6D pose parameters. The left two columns show some good examples, while the right two columns are several failure cases, which shows the model still has trouble estimating the rotation of symmetrical objects.

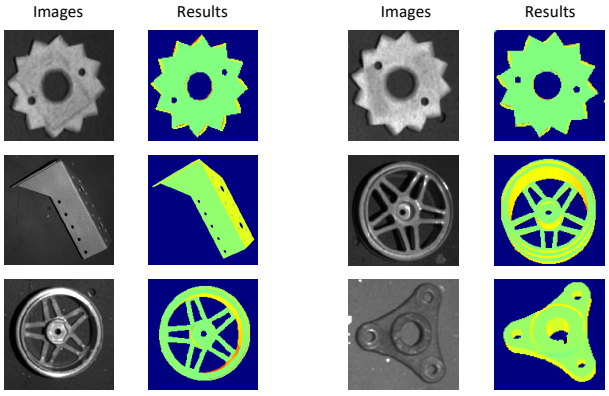

Fig. 3. Examples of 6D pose estimation results.

IV. CONCLUSIONS

In this work, a two-stage 6D pose estimation model was proposed to facilitate the recognition of small pose variations of industrial parts, consisting of a coarse pose estimation stage and a pose refinement stage. High-Resolution Network was leveraged as backbone network in both stages to maintain high-resolution feature representations. Meanwhile, depth information was incorporated to the model as an extra channel of the input. Then the rotation and translation parameters are regressed separately in the first stage and refined in the refinement stage. The experimental result shows that the proposed model outperforms the baseline model on an industrial parts dataset. Nevertheless, it still has its own limitations when dealing with the rotation of symmetrical objects. Hence, future work can be done to 1) explore better approaches to handle symmetrical objects, 2) reduce the dependency on large amounts of training data,

and 3) implement it into more promising applications (e.g. human-robot collaborative assembly).

REFERENCES

[1] B. Sadrfaridpour, H. Saeidi, and Y. Wang, "An integrated framework for human-robot collaborative assembly in hybrid manufacturing cells," in *2016 IEEE International Conference on Automation Science and Engineering*. IEEE, 2016, pp. 462–467.

[2] L. Wang, R. Gao, J. Váncza, J. Krüger, X. V. Wang, S. Makris, and G. Chryssolouris, "Symbiotic human-robot collaborative assembly," *CIRP annals*, vol. 68, no. 2, pp. 701–726, 2019.

[3] B. Saund, S. Chen, and R. Simmons, "Touch based localization of parts for high precision manufacturing," in *2017 IEEE International Conference on Robotics and Automation*, 2017, pp. 378–385.

[4] J. Bimbo, P. Kormushev, K. Althoefer, and H. Liu, "Global estimation of an object's pose using tactile sensing," *Advanced Robotics*, vol. 29, no. 5, pp. 363–374, 2015.

[5] P. Tsarouchi, S.-A. Matthaiakis, G. Michalos, S. Makris, and G. Chryssolouris, "A method for detection of randomly placed objects for robotic handling," *CIRP Journal of Manufacturing Science and Technology*, vol. 14, pp. 20–27, 2016.

[6] S. Astanin, D. Antonelli, P. Chiabert, and C. Alletto, "Reflective workpiece detection and localization for flexible robotic cells," *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 190–198, 2017.

[7] V.-T. Nguyen, C. Lin, C.-H. G. Li, S.-M. Guo, and J.-J. J. Lien, "Visual-guided robot arm using self-supervised deep convolutional neural networks," in *2019 IEEE 15th International Conference on Automation Science and Engineering*. IEEE, 2019, pp. 1415–1420.

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[11] K.-J. Wang, D. A. Rizqi, and H.-P. Nguyen, "Skill transfer support model based on deep learning," *Journal of Intelligent Manufacturing*, pp. 1–18, 2020.

[12] H. Nguyen, N. Adrian, J. L. X. Yan, J. M. Salfity, W. Allen, and Q.-C. Pham, "Development of a robotic system for automated decaking of 3d-printed parts," in *2020 IEEE International Conference on Robotics and Automation*. IEEE, 2020, pp. 8202–8208.

[13] S. Back, J. Kim, R. Kang, S. Choi, and K. Lee, "Segmenting unseen industrial components in a heavy clutter using rgb-d fusion and synthetic data," in *2020 IEEE International Conference on Image Processing*. IEEE, 2020, pp. 828–832.

[14] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 998–1005.

[15] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision*, vol. 66, no. 3, pp. 231–259, 2006.

[16] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–888, 2011.

[17] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 548–562.

[18] Z. He, Z. Jiang, X. Zhao, S. Zhang, and C. Wu, "Sparse template-based 6-d pose estimation of metal parts using a monocular camera," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 390–401, 2019.

[19] 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.

[20] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[21] 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.

[22] 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.

[23] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[24] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 632–11 641.

[25] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 683–698.

[26] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3343–3352.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[29] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[30] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger, "Introducing mvtec itodd-a dataset for 3d object recognition in industry," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2200–2208.

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8026–8037, 2019.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[34] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *European Conference on Computer Vision*. Springer, 2020, pp. 574–591.

[35] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.