

# Image-Based Pose Estimation of Sub-Centimeter Industrial Parts for Automated Assembly

Hameed Abdul-Rashid<sup>1</sup>, Yangfei Dai<sup>1</sup>, Holly Dinkel<sup>1</sup>, Minh Quang Ta<sup>2</sup>, Tan Chen<sup>3</sup>,  
Junyi Geng<sup>4</sup>, and Timothy Bretl<sup>1</sup>

<sup>1</sup>University of Illinois Urbana-Champaign

<sup>2</sup>University of West Florida

<sup>3</sup>Michigan Technological University

<sup>4</sup>Pennsylvania State University

<sup>1</sup>{hameeda2, yangfei4, hdinkel2, tbretl}@illinois.edu, <sup>2</sup>mta@uwf.edu,  
<sup>3</sup>tanchen@mtu.edu, <sup>4</sup>jgeng@psu.edu

**Abstract:** This work integrates existing machine vision techniques to estimate the 6DoF pose of sub-centimeter parts from images for high-mix, low-volume assembly lines. In this system, the 3D models of three parts are input to a Blender-Proc2 rendering engine to generate a physically- and photometrically-realistic synthetic image dataset. Synthetic images are used to train a Mask R-CNN model for segmenting individual part instances in a scene, with automatically-generated instance mask labels, eliminating the need for manual labeling. Instance segmentation enables part selection for assembly when multiple parts are present. Additionally, a PVNet model is trained on cropped images of each part instance from synthetic data to estimate their positions and orientations. An additional pose refinement step adjusts PVNet pose estimates by aligning the orientation to the nearest physically-stable configuration on a planar surface and refining the translation using calibrated object-to-camera distances from the workspace. To evaluate robustness, noise is injected into the keypoint detection stage of the PVNet model in an ablation study to assess the impact of sensor noise on pose estimation. Real robot pick-and-place experiments demonstrate the system performance.

**Keywords:** Synthetic Data, Instance Segmentation, Pose Estimation, Robotic Manipulation, Industrial Assembly, Sim2Real Transfer

## 1 Introduction

The objective of this work is to estimate the 6DoF pose of three sub-centimeter industrial parts—an inserted part, a main part, and a top part—which are assembled to form a USB Type-C connector. Automating assembly with customized machinery is not cost-effective in low-volume production. A reliable and cost-effective part pose estimation pipeline is essential to enable high-precision, sub-mm tolerance assembly by unsupervised industrial robots on high-mix, low-volume production lines [1]. Previous work investigates small-scale part assembly with robot arms in simulation, however it assumes known object poses prior to picking [2]. This work presents an image-based system that localizes and estimates the pose of industrial parts, shown in Figure 1, from a tabletop spread of components for picking and assembly. The system assumes that parts are arranged in an unstacked, planar configuration on a flat surface, without any overlap or occlusions. The system first identifies different categories of parts and their unique instances in a scene, then estimates the pose of each part. These pose estimates can then be used by a robot system for picking and assembly.

This work addresses two challenges in developing pose estimation systems for automated sub-cm robotic picking: the requirement of large amounts of accurate object pose annotations for training pose estimators, and the requirement of accurate model inference for robotic picking. To respond to these challenges, this work makes the following contributions:

1. This work introduces a data generation pipeline that produces a photorealistic, labeled synthetic image dataset for training both an object instance segmentation model and a pose estimation model. Using a physics-based rendering engine, the pipeline simulates object CAD models in physically stable configurations within scenes that closely replicate real-world picking environments.
2. A method of object instance segmentation and a method of object pose estimation are deployed in series to localize and predict the pose of three types of sub-cm industrial parts from a photometrically challenging scene containing many part instances with wide pose distributions.
3. The sensitivity of pose estimation to pixel noise and object distance is analyzed in an ablation study to characterize admissible pixel noise and object-to-camera distance limits.
4. The pose estimation system—learned entirely on synthetic data—is directly applied to picking real parts with real robot arms.
5. Source code is openly released at [github.com/Hammania689/sub-cm-part-pose](https://github.com/Hammania689/sub-cm-part-pose).

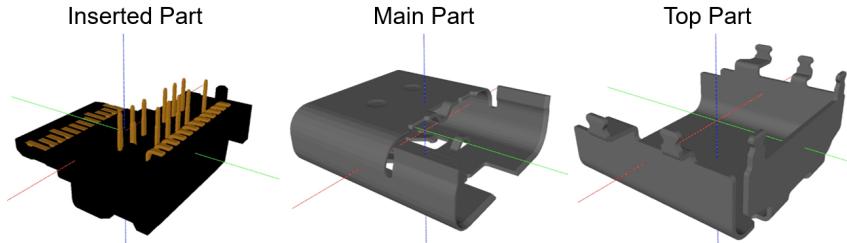


Figure 1: The CAD models of three sub-centimeter, symmetric parts are inputs to the learned pose estimation system.

## 2 Related Work

Object instance information is a prerequisite for many pose estimation methods [3, 4, 5]. Object instance detection localizes regions of an image containing object instances, while instance segmentation classifies image pixels as belonging to specific object instances. Instance detection predicts bounding box regions and class labels within the image [6] while instance segmentation may or may not use the context of the object class to predict masks [7]. It is common to simultaneously predict object classes, bounding boxes, and instance masks from RGB images. One method which does this is Mask R-CNN, a model trained to predict instance segmentation mask and class labels for each object in an image [8]. More recent convolutional neural network-based methods [9] and transformer-based methods [10, 11] also offer promising performance improvements over the Mask R-CNN baseline.

A drawback of using Mask R-CNN for instance segmentation is the requirement of large quantities of high-quality data with corresponding instance class and mask labels. Labeling can be challenging

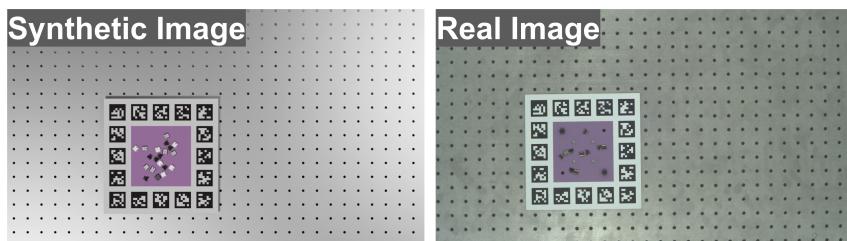


Figure 2: BlenderProc2 renders photorealistic, physics-based scene images (left) that closely resemble real-world scenes captured with a Basler acA5472-17uc camera with a 25 mm lens (right). While the synthetic image achieves similar detail, minor discrepancies remain, such as subtle differences in lighting conditions, slightly reduced noise levels, and slight variations in the reflective properties of certain parts.

due to partial occlusion, scene clutter, extreme lighting variations, a large variety of object poses, and a large number of instances per image [12, 13, 14, 15]. Furthermore, texture-less, symmetric, and reflective objects present additional photometric challenges during labeling and inference [16, 17, 18, 19, 20, 21, 22]. Photorealistic synthetic image generation with physics-based CAD model rendering is one promising alternative to collecting a large number of real images and manually annotating them [23].

Pose estimation finds the 6DoF position and orientation of an object in a scene. Several methods address the pose estimation problem using various types of prior information about the object as input. One approach estimates object pose directly from RGB images for deployment in systems using low-cost camera sensors [24]. Another method for image-only 6DoF pose estimation is PoseCNN [25]. The PoseCNN method estimates the translation of an object by finding its pixel center in the image, estimating its distance from the camera, and regressing object features within a bounding box representation to a quaternion representation. The RCVPose method also estimates object 6DoF pose from images [26]. The RCVPose method learns to estimate the distance between a 3D keypoint and the 3D scene point corresponding to each image pixel. For each pixel, a sphere with a radius equal to this regressed distance is centered at each corresponding 3D scene point. The 3D keypoint locations are estimated, and execution for at least 3 keypoints allows the recovery of each unique 6DoF object pose.

A second approach to pose estimation incorporates depth information. One existing method estimates pose directly from 3D depth maps and uses object CAD model alignment for training, bypassing incorporating image information entirely [27]. The OVE6D model uses a depth image, an object mask, and a viewpoint codebook built off of varying viewpoints of object CAD models to predict object poses [28]. These depth-based methods offer promising results, but may struggle to estimate the pose of sub-cm parts. The depth accuracy for off-the-shelf depth sensors positioned 1 m away from an object—a typical distance in robot work cells—may be 1% of the distance to the object (i.e., 1 cm) [29]. This depth accuracy may not meet requirements for precise pose estimation of small-scale parts.

Other pose estimation approaches use data beyond the single-frame image and depth information provided by cameras. One method performs in-hand manipulation to estimate object pose through combination of image and tactile information to reduce pose estimation uncertainty [30]. Other methods perform 6DoF pose tracking, which is especially useful when the object intermittently leaves the view of the camera or is occluded [31, 32].

This paper uses PVNet, an image-based method, for pose estimation [33]. The PVNet method performs two tasks: semantic segmentation and vector-field prediction. For each pixel, it outputs an instance mask linking it to an object and a unit vector indicating the direction from the pixel to a keypoint on the object. Based on the instance masks and unit vectors, keypoint hypotheses are generated using a RANSAC-based voting scheme. A custom PnP solver is also proposed that takes a subset of the predicted 2D keypoints and known set of 3D keypoints from training to estimate the 6D pose of the object.

### 3 Methodology

This work uses BlenderProc2 to generate a photorealistic scene dataset with BlenderProc2 to train a Mask R-CNN instance segmentation model. Part instance crops are input to PVNet for part pose estimation. The pose estimation pipeline is shown in Figure 3.

In the BlenderProc2 rendering engine, each object is initialized in random poses within a simulated environment and allowed to fall onto a planar surface. Once stationary, the final pose of the  $i$ -th part is labeled as a transformation matrix in the camera frame as

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ 0 & 1 \end{bmatrix}, \quad (1)$$

where  $\mathbf{R}_i \in \mathbb{R}^{3 \times 3}$  is the true  $i^{th}$  rotation matrix and  $\mathbf{t}_i \in \mathbb{R}^{3 \times 1}$  is the true  $i^{th}$  translation vector. Each part in each scene image is also labeled with an instance segmentation mask, 2D and 3D keypoint

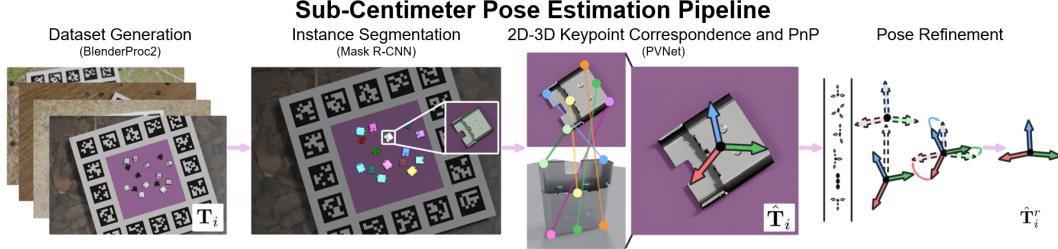


Figure 3: The image-based pose estimation system generates a photorealistic scene dataset with BlenderProc2 to train a Mask R-CNN instance segmentation model. Part instance crops are input to PVNet for 2D-to-3D keypoint correspondence and Perspective-n-Point estimation of part pose. The estimated poses are refined by aligning the orientation to the nearest stable pose and adjusting the translation based on calibrated object-to-camera distances from the workspace.

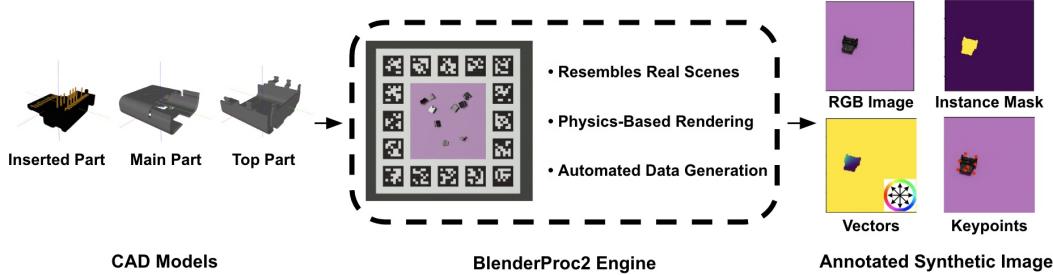


Figure 4: The BlenderProc2 physics-based engine renders part CAD models in replicated real-world scenes. This enables generation of synthetic image datasets with known part instance segmentation masks, 2D and 3D keypoint coordinates, and 2D vectors between the center and keypoints.

coordinates selected with the Farthest Point Sampling (FPS) algorithm, and 2D vectors between the center and keypoints as shown in Figure 4. Rendering is repeated until the dataset includes  $N$  instances of each part across all rendered images. Each rendered part has a set of possible stable pose candidates when the part is in contact with a planar surface. Orientations with greater contact area between the part and the planar surface appear with higher probability. The distribution of stable pose candidates for each part is shown in Figure 5.

A Mask R-CNN model is trained on the synthetic images and instance mask labels to predict the class, mask, and bounding box for each part instance [34]. The center pixel of the part instance bounding box is used to crop an image patch input to PVNet. A PVNet model is trained for each part class to predict pixel-wise unit vectors from the object center pointing to the keypoints used for keypoint localization. Keypoints are used in 2D-to-3D correspondence matching and solving for pose using the Perspective-n-Point (PnP) algorithm. The predicted pose for each part instance,  $\hat{\mathbf{T}}_i$ , is a transformation matrix in the camera frame,

$$\hat{\mathbf{T}}_i = \begin{bmatrix} \hat{\mathbf{R}}_i & \hat{\mathbf{t}}_i \\ 0 & 1 \end{bmatrix}, \quad (2)$$

where  $\hat{\mathbf{R}}_i \in \mathbb{R}^{3 \times 3}$  is the PVNet-predicted  $i^{\text{th}}$  rotation matrix and  $\hat{\mathbf{t}}_i \in \mathbb{R}^{3 \times 1}$  is the PVNet-predicted  $i^{\text{th}}$  translation vector. The PVNet model provides an initial pose estimate based on the cropped image patch. The orientation estimates of PVNet are refined by assuming the part instance lies on a table in a stable planar orientation. The possible stable orientations used for pose refinement are the same as those discovered by BlenderProc2 and shown in Figure 5. The  $z$ -component of the position for each part instance is also replaced with the part center-to-camera distance known obtained from extrinsic calibration. The refined part pose,  $\hat{\mathbf{T}}_i^r$ , is

$$\hat{\mathbf{T}}_i^r = \begin{bmatrix} \hat{\mathbf{R}}_i^r & \hat{\mathbf{t}}_i^r \\ 0 & 1 \end{bmatrix}, \quad (3)$$

where  $\hat{\mathbf{R}}_i^r$  is the refined orientation estimate and  $\hat{\mathbf{t}}_i^r$  is the refined position estimate.

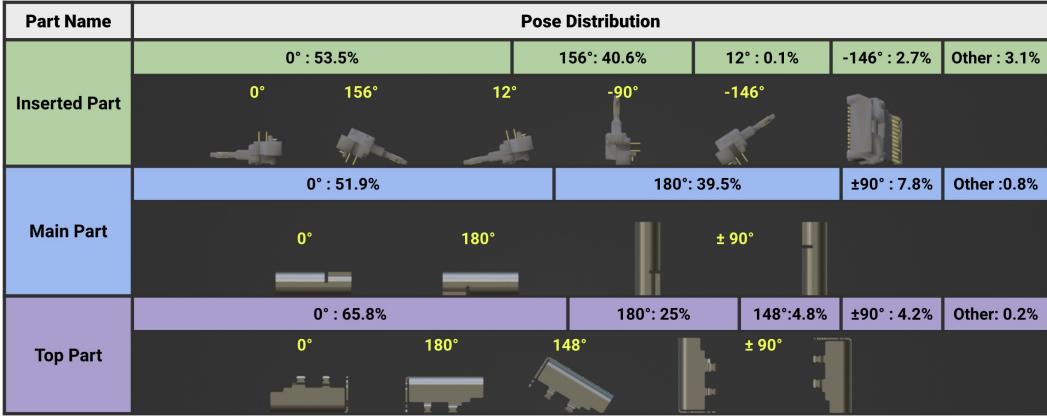


Figure 5: Each rendered part has a set of possible physically-stable pose candidates given contact between the part and a planar surface. Orientations with greater contact area between the part and the planar surface appear with higher probability. Since the parts are symmetric about the  $y$ -axis, the possible stable orientations are shown here as the roll angles (rotation about the  $x$ -axis).

## 4 Experiments

In experiments analyzing the accuracy of part instance segmentation in the pipeline, instance classes were predicted with high precision and recall and high instance mask intersection as compared to ground truth labels (Section 4.1). In experiments analyzing part pose estimation, part positions were estimated within 1mm of planar translation error and 3mm vertical translation error using position refinement and within  $< 5^\circ$  of angular error (Section 4.2). An ablation study was performed to highlight the relationship between pixel noise and object-to-camera distance on pose estimation performance (Section 4.3), and the learned pose estimation system was deployed in a real robot picking workcell as an enabling step for robotic assembly (Section 4.4).

The BlenderProc2 engine was used to generate 10,000 images in 14 hours at  $2208 \times 1242$  resolution for training Mask R-CNN. The BlenderProc2 engine was used to generate three part-specific datasets, each containing 20,000 images at  $256 \times 256$  resolution, in 17 hours for training PVNet. The Mask R-CNN model was trained for four hours with 5 epochs and a batch size of 1. Three part-specific PVNet models were trained for a total of 18 hours, and each model was trained for 250 epochs with a batch size of 32. Dataset generation and training was completed on a workstation with two NVIDIA RTX 3090 GPUs and a 24-core AMD Ryzen Threadripper 3960x CPU.

### 4.1 Instance Segmentation Evaluation

Mask R-CNN was evaluated on a test set of 200 synthetic images generated with BlenderProc2. The lighting, camera distance, object poses, scene backgrounds of these images are different from the training data. This was done to get a proxy on how robust the model is to variations that may occur in real environments. The quantitative results on this synthetic test set are summarized in Table 1, while qualitative results on real images are shown in Figure 7.

The mean Intersection over Union, or mIoU, metric is a common choice for evaluating the inference performance of semantic segmentation models. While this metric does not distinguish between instances, it is still useful for quantifying average pixel-wise overlap between predicted and ground truth segmentation masks. The mIoU is

$$\text{mIoU} = \frac{1}{N} \sum_{i=1}^N \frac{\text{area of intersection}_i}{\text{area of union}_i}, \quad (4)$$

where  $N$  is the total number of classes. A detection is classified as a True Positive,  $TP$ , if the class label is correctly predicted and the instance mask exceeds a specified IoU threshold. A detection is classified as a False Positive,  $FP$ , if its instance mask does not correspond to any ground truth

instance. A detection is classified as a False Negative,  $FN$ , if the model does not detect any instance where one exists in the ground truth label.

For instance-level evaluation, Average Precision, AP, and Average Recall, AR, are each defined as

$$AP = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i} \quad (5)$$

$$AR = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FN_i}. \quad (6)$$

Table 1: Mask R-CNN Instance Segmentation Evaluation

Metric	Inserted Part	Main Part	Top Part	Overall
mIoU	89.7 ( $\pm 20.7$ )	80.4 ( $\pm 29.8$ )	91.6 ( $\pm 12.8$ )	87.5 ( $\pm 22.2$ )
AP <sub>90</sub>	95.2	73.0	89.8	86.0
AR <sub>90</sub>	96.4	83.0	92.9	90.7
Number of Instances	995	980	998	3158

## 4.2 Pose Estimation Evaluation

Table 2 presents the raw output and refined output of PVNet evaluated on a synthetic testing dataset consisting of 2,000 images, where translation and angular error are used to evaluate the accuracy of the 6DoF pose estimation. Sample predictions on real images are shown in Figure 8, demonstrating the model’s ability to transfer from synthetic training data to real-world scenarios, though some failure cases can occur as illustrated in Figure 9. Since the object pose is expressed as an orientation and a position, two error metrics are used to quantitatively evaluate the estimate. Assembly of sub-centimeter components with tight insertion tolerances requires accurate pose estimates for picking and assembly. The translation error quantifies the difference between the estimated and true positions in Cartesian space. The translation error is calculated as the L1 norm, which is the sum of the absolute differences between the predicted translation,  $\hat{\mathbf{t}}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$ , and the true translation,  $\mathbf{t}_i = (x_i, y_i, z_i)$ . The translation errors for each dimension are

$$\begin{aligned} e_{x,i} &= |\hat{x}_i - x_i| \\ e_{y,i} &= |\hat{y}_i - y_i| \\ e_{z,i} &= |\hat{z}_i - z_i| \end{aligned} \quad (7)$$

The angular error,  $e_{\theta,i}$ , measures the deviation in orientation between the predicted rotation,  $\hat{\mathbf{R}}_i$ , and true rotation,  $\mathbf{R}_i$ , as

$$e_{\theta,i} = \cos^{-1} \left( \frac{\text{tr}(\mathbf{R}_i^\top \hat{\mathbf{R}}_i) - 1}{2} \right). \quad (8)$$

One challenge with evaluation is the difficulty of estimating depth using RGB images which lack direct depth information. While the refinement step improves both depth and orientation accuracy, errors can arise from the camera-to-table surface extrinsic calibration used to replace the  $z$ -component with the part center-to-camera distance.

## 4.3 Sensitivity of PnP Solver

In PVNet, a set of 2D keypoints are estimated from RGB images. The object pose is determined by establishing 2D-3D correspondences between the localized keypoints in an image and a CAD model and minimizing keypoint reprojection errors—typically solved using the PnP algorithm. Optimization with PnP is highly sensitive to keypoint detection accuracy. Small pixel shifts or errors in keypoint coordinates can lead to significant pose estimation errors. The degree of this sensitivity is also influenced by the focal length of the sensor, the distance between the object and the sensor, and the size of the object. An ablation study was conducted to examine the relationship between pose

Table 2: PVNet Evaluation

Metric	Error Type	Inserted Part	Main Part	Top Part	Average
Translation error (mm)	$e_x$	0.57 ( $\pm 0.44$ )	0.62 ( $\pm 0.48$ )	0.99 ( $\pm 0.75$ )	0.73 ( $\pm 0.56$ )
	$e_y$	0.56 ( $\pm 0.44$ )	0.59 ( $\pm 0.46$ )	0.98 ( $\pm 0.75$ )	0.71 ( $\pm 0.55$ )
	$e_z$	62.01 ( $\pm 22.19$ )	64.24 ( $\pm 20.15$ )	105.41 ( $\pm 28.53$ )	77.22 ( $\pm 23.62$ )
	$e_z$ (refined)	1.02 ( $\pm 2.75$ )	0.97 ( $\pm 0.73$ )	1.29 ( $\pm 0.90$ )	1.09 ( $\pm 1.46$ )
Angular error ( $^{\circ}$ )	$e_{\theta}$	3.39 ( $\pm 1.65$ )	3.85 ( $\pm 1.79$ )	3.09 ( $\pm 1.88$ )	3.44 ( $\pm 1.77$ )
	$e_{\theta}$ (refined)	2.52 ( $\pm 0.27$ )	2.60 ( $\pm 0.50$ )	2.55 ( $\pm 0.33$ )	2.56 ( $\pm 0.37$ )

Note: All values are in the format: mean ( $\mu$ )  $\pm$  standard deviation ( $\sigma$ ).

prediction errors and keypoint noise,  $\epsilon$ , for varying average object-to-camera distances,  $\mu_d$ , where  $\mu_d$  is

$$\mu_d = \frac{1}{M} \sum_{i=1}^m d_i. \quad (9)$$

Different levels of noise were introduced to the pixel coordinates of eight pre-defined keypoints in the dataset. The perturbed keypoints and the camera intrinsics were input to an Efficient Perspective-n-Point (EPnP) solver to estimate the object pose relative to the camera [35]. The results of the ablation study, shown in Figure 6, indicate  $e_x$  and  $e_y$  are relatively insensitive to  $\epsilon$ . However,  $e_z$  increases significantly with increased  $\epsilon$ , especially for large  $\mu_d$ . Similarly,  $e_{\theta}$  increases exponentially with  $\epsilon$  and is further amplified at large  $\mu_d$ .

This finding underscores the importance of precise keypoint localization in achieving accurate depth and orientation estimates when using 2D-3D correspondence-based methods that rely on PnP solvers. In practical applications, it suggests the potential benefit of incorporating depth information alongside high-resolution imaging, robust keypoint detection, or effective noise-reduction strategies to enhance 3D pose accuracy, especially in setups with greater object-to-camera distances.

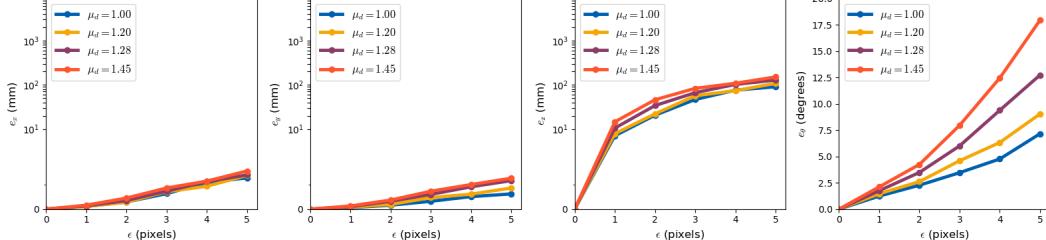


Figure 6: These plots illustrate the relationship between the translation prediction errors  $e_x$  (left),  $e_y$  (center left),  $e_z$  (center right), and angular prediction error  $e_{\theta}$  (right) as a function of  $\epsilon$  for varying  $\mu_d$ . Planar translation errors ( $e_x$  and  $e_y$ ) are relatively insensitive to  $\epsilon$ , but the non-planar translation error ( $e_z$ ) grows with  $\epsilon$ , with larger  $\mu_d$  exacerbating the error. Rotation error ( $e_{\theta}$ ) grows exponentially with increasing  $\epsilon$ , amplified at greater  $\mu_d$ .

#### 4.4 Robot Demo

This pose estimation system is deployed in a real robot workspace equipped with a Universal Robots UR5e robot arm, Robotiq Hand-E gripper, and Basler acA5472-17uc camera with a 25 mm lens. The task is to pick and place an inserted part, a main part, and a top part from random initial configurations on a tagboard surface. With a scene configuration similar to the synthetic scene configured in BlenderProc2 for image data generation, the camera was mounted on a stand in the workspace and captured the workspace from a top-down perspective. Figure 7 shows Mask R-CNN inference performance for instance segmentation of each part category in real scene images. Figure 8 shows PVNet inference performance for pose estimation of each part category in real scene image crops. Figure 9 shows how incorrect instance segmentation can lead to part pose estimation failures within the system. Figure 10 demonstrates the full instance segmentation, pose estimation, and robot picking system for four randomly configured inserted parts<sup>1</sup>.

<sup>1</sup>A demo video is available at [github.com/Hammania689/sub-cm-part-pose](https://github.com/Hammania689/sub-cm-part-pose)

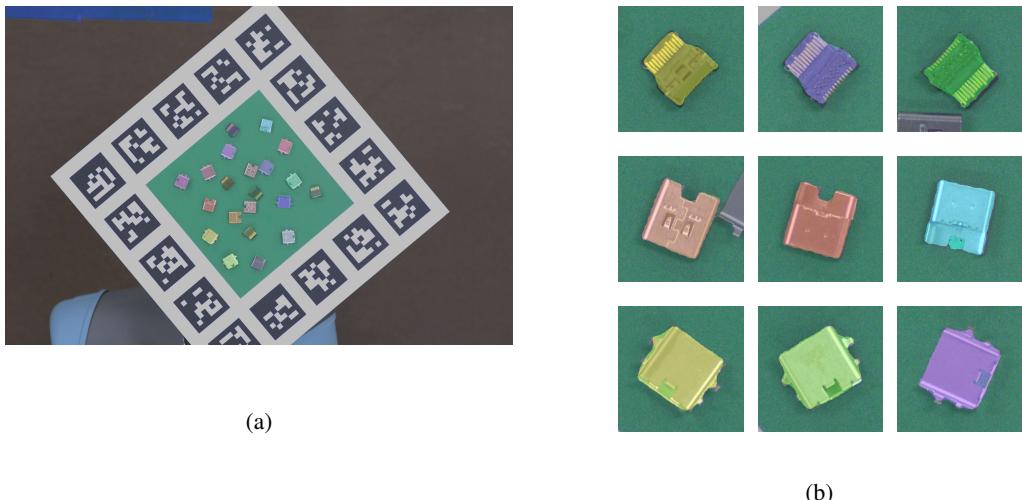


Figure 7: This figures shows Mask R-CNN instance segmentation results on real images. (a) Detection is performed on full-scene images with multiple detected instances for each part category. (b) Cropped part instances from instance segmentation are inputs to PVNet. Here, each column shows different instances of each part class.

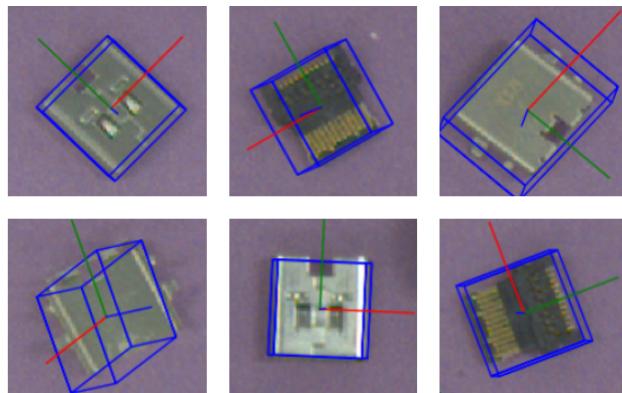


Figure 8: Visualization of pose estimation results obtained from PVNet on the inserted part, main part, and top part. The detected 3D bounding boxes are shown in blue, while the estimated coordinate axes are indicated by colored lines (red:  $x$ -axis, green:  $y$ -axis, blue:  $z$ -axis).

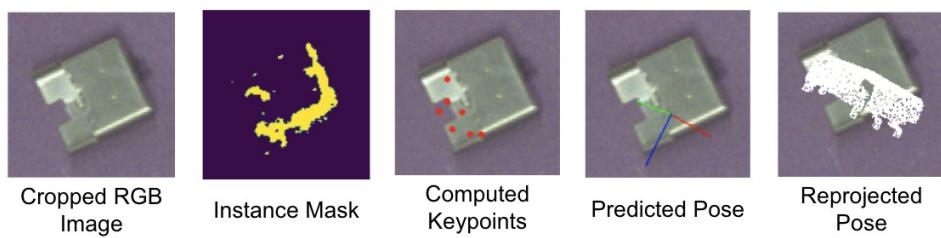


Figure 9: This image shows a failure case in our system. In this instance, Mask R-CNN incorrectly classified the main part as a top part, feeding the crop to the PVNet model trained specifically for the top part. As a result, the PVNet model produced an inaccurate instance mask and incorrect keypoints, leading to a severely erroneous pose estimation. This misclassification and subsequent processing by an inappropriate model resulted in significant errors throughout the pipeline, as illustrated by the poor alignment in the predicted and reprojected poses.

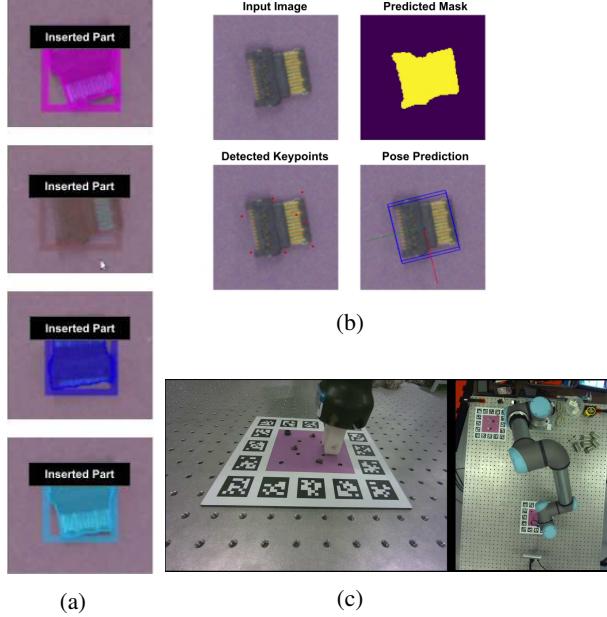


Figure 10: This figure illustrates a robot grasping detected parts. (a) Mask R-CNN detects and segments four instances of the inserted part. (b) One instance of a segmented inserted part is processed by PVNet, which performs semantic segmentation and predicts keypoints and uses this information to compute the 6DoF pose. (c) The robot grasps an inserted part based on its predicted pose.

## 5 Conclusion

This work presented a vision-based system for pose estimation of sub-centimeter industrial parts. This implementation of PVNet requires prior information from object instance detection and segmentation which is out of scope of the original PVNet implementation. An additional refinement strategy was also proposed to improve pose estimation accuracy. This approach can be applied not only to USB Type-C components, but also to other small-scale, tight-contact-tolerance industrial objects such as screws, bolts, and other connectors. Future work will focus on integrating this sub-centimeter pose estimation pipeline into a complete assembly line of components.

## Acknowledgments

The authors thank the members of the UIUC-FIT CoBot Factory Project and the teams developing the open-source software used in this project [36, 37, 38, 39, 40, 41, 34, 23]. All authors were supported by the Foxconn Interconnect Technology (FIT) and the Center for Networked Intelligent Components and Environments (C-NICE) at the University of Illinois Urbana-Champaign. Holly Dinkel was also supported by the Graduate Assistance in Areas of National Need award P200A180050-19 and the NASA Space Technology Graduate Research Opportunity awards 80NSSC21K1292.

## References

- [1] T. Chen, Z. Huang, J. Motes, J. Geng, Q. M. Ta, H. Dinkel, H. Abdul-Rashid, J. Myers, Y.-J. Mun, W.-C. Lin, Y.-Y. Yang, S. Liu, M. Morales, N. M. Amato, K. Driggs-Campbell, and T. Bretl. Insights from an Industrial Collaborative Assembly Project: Lessons in Research and Collaboration. In *IEEE Int. Conf. Robot. Autom. (ICRA) Workshop on Collaborative Robots and Work of the Future*, 2022.
- [2] M. Q. Ta, H. Dinkel, H. Abdul-Rashid, Y. Dai, J. Myers, T. Chen, J. Geng, and T. Bretl. The Impact of Time Step Frequency on the Realism of Robotic Manipulation Simulation for Objects of Different Scales. In *IEEE Int. Conf. Intell. Robot. Sys. (IROS) Workshop on Robotics and AI in Future Factory*, 2023.
- [3] G. Wang, F. Manhardt, F. Tombari, and X. Ji. GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 16611–16621, 2021.
- [4] V. N. Nguyen, Y. Hu, Y. Xiao, M. Salzmann, and V. Lepetit. Templates for 3D Object Pose Estimation Revisited: Generalization to New objects and Robustness to Occlusions. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.

- [5] V. N. Nguyen, T. Groueix, M. Salzmann, and V. Lepetit. GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.
- [6] D. Dwibedi, I. Misra, and M. Hebert. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 1301–1310, 2017.
- [7] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment Anything. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 3992–4003, 2023.
- [8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.
- [9] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, et al. Internimage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 14408–14419, 2023.
- [10] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum. Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3041–3050, 2023.
- [11] C. Xia, X. Wang, F. Lv, X. Hao, and Y. Shi. ViT-CoMer: Vision Transformer with Convolutional Multi-scale Feature Interaction for Dense Predictions. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5493–5502, 2024.
- [12] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel. BigBIRD: A Large-Scale 3D Database of Object Instances. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 509–516, 2014.
- [13] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research. In *IEEE Int. Conf. Adv. Robot. (ICAR)*, pages 510–517, 2015.
- [14] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield. 6-DoF Pose Estimation of Household Objects for Robotic Manipulation: An Accessible Dataset and Benchmark. In *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, pages 13081–13088, 2022.
- [15] A. Guo, B. Wen, J. Yuan, J. Tremblay, S. Tyree, J. Smith, and S. Birchfield. HANDAL: A Dataset of Real-World Manipulable Object Categories with Pose Annotations, Affordances, and Reconstructions. In *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, pages 11428–11435, 2023.
- [16] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal Templates for Real-Time Detection of Texture-Less Objects in Heavily Cluttered Scenes. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 858–865, 2011.
- [17] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis. T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-Less Objects. In *IEEE Winter Conf. Comput. Vis. (WACV)*, pages 880–888, 2017.
- [18] B. Drost, M. Ulrich, P. Bergmann, P. Härtinger, and C. Steger. Introducing MVtec ITODD - A Dataset for 3D Object Recognition in Industry. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, pages 2200–2208, 2017.
- [19] K. Kleeberger, C. Landgraf, and M. F. Huber. Large-Scale 6D Object Pose Estimation Dataset for Industrial Bin-Picking. In *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, pages 2573–2578, 2019.
- [20] J. Yang, Y. Gao, D. Li, and S. L. Waslander. ROBI: A Multi-View Dataset for Reflective Objects in Robotic Bin-Picking. In *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, pages 9788–9795, 2021.
- [21] P. Wang, H. Jung, Y. Li, S. Shen, R. P. Srikanth, L. Garattoni, S. Meier, N. Navab, and B. Busam. PhoCaL: A Multi-Modal Dataset for Category-Level Object Pose Estimation With Photometrically Challenging Objects. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.
- [22] L. Chen, H. Yang, C. Wu, and S. Wu. MP6D: An RGB-D Dataset for Metal Parts' 6D Pose Estimation. *IEEE Robot. Autom. Lett.*, 7(3):5912–5919, 2022.
- [23] M. Denninger, D. Winkelbauer, M. Sundermeyer, W. Boerdijk, M. W. Knauer, K. H. Strobl, M. Humt, and R. Triebel. BlenderProc2: A Procedural Pipeline for Photorealistic Rendering. *J. Open Source Softw.*, 2023.
- [24] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and c. Rother. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes From a Single RGB Image. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016.
- [25] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robot. Sci. Syst. (RSS)*, Pittsburgh, Pennsylvania, June 2018.
- [26] Y. Wu, M. Zand, A. Etemad, and M. Greenspan. Vote from the Center: 6 DoF Pose Estimation in RGB-D Images by Radial Keypoint Voting. In *Eur. Conf. Comput. Vis. (ECCV)*, 2022.
- [27] J. Yang, D. Li, and S. L. Waslander. Probabilistic Multi-View Fusion of Active Stereo Depth Maps for Robotic Bin-Picking. *IEEE Robot. Autom. Lett.*, 6(3):4472–4479, 2021.
- [28] D. Cai, J. Heikkilä, and E. Rahtu. OVE6D: Object Viewpoint Encoding for Depth-based 6D Object Pose Estimation. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 6793–6803, 2022.
- [29] Intel. Compare depth cameras. <https://www.intelrealsense.com/compare-depth-cameras/>, 2024.

- [30] F. von Drigalski, K. Hayashi, Y. Huang, R. Yonetani, M. Hamaya, K. Tanaka, and Y. Ijiri. Precise Multi-Modal In-Hand Pose Estimation using Low-Precision Sensors for Robotic Assembly. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021.
- [31] B. Wen, C. Mitash, B. Ren, and K. E. Bekris. SE(3)-TrackNet: Data-driven 6D Pose Tracking by Calibrating Image Residuals in Synthetic Domains. In *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, pages 10367–10373, 2020.
- [32] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox. PoseRBPF: A Rao–Blackwellized Particle Filter for 6-D Object Pose Tracking. *IEEE Trans. Robot.*, 37(5):1328–1342, 2021.
- [33] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation. In *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.
- [34] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [35] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *Int. J. Comput. Vis.*, 81:155–166, 2009.
- [36] Stanford Artificial Intelligence Laboratory. Robotic Operating System: Noetic Ninjemys, 2018. URL <https://www.ros.org>.
- [37] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [38] C. R. Harris, J. Millman, S. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array Programming with NumPy. *Nature*, 585:357–362, 2020.
- [39] J. D. Hunter. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.*, 9(3):90–95, 2007.
- [40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Adv. Neur. Inf. Proc. (NeurIPS)*, 32, 2019.
- [41] P. Virtan, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods*, 17:261—272, 2020.