machines

Article

A Novel Deep Learning-Based Pose Estimation Method for Robotic Grasping of Axisymmetric Bodies in Industrial Stacked Scenarios

Yaowei Li [1], Fei Guo [1,*], Miaotian Zhang [1], Shuangfu Suo [1], Qi An [1], Jinlin Li [1] and Yang Wang [2]

[1] State Key Laboratory of Tribology, Tsinghua University, Beijing 100084, China
[2] Chinese Academy of Ordnance Sciences, Beijing 100089, China
* Correspondence: guof2014@mail.tsinghua.edu.cn

Abstract: A vision-based intelligent robotic grasping system is essential for realizing unmanned operations in industrial manufacturing, and pose estimation plays an import role in this system. In this study, deep learning was used to obtain the 6D pose of an axisymmetric body which was optimal for robotic grasping in industrial stacked scenarios. We propose a method to obtain the 6D pose of an axisymmetric body by detecting the pre-defined keypoints on the side surface. To realize this method and solve other challenges in industrial stacked scenarios, we propose a multitask real-time convolutional neural network (CNN), named Key-Yolact, which involves object detection, instance segmentation, and multiobject 2D keypoint detection. A small CNN as a decision-making subsystem was designed to score multiple predictions of Key-Yolact, and the body with the highest score is considered the best for grasping. Experiments on a self-built stacked dataset showed that Key-Yolact has a practical tradeoff between inference speed and precision. The inference speed of Key-Yolact is higher by 10 FPS, whereas its precision is decreased by only 7% when compared with the classical multitask Keypoint R-CNN. Robotic grasping experiments showed that the proposed design is effective and can be directly applied to industrial scenarios.

Keywords: 6D object pose estimation; multitask CNN; real-time CNN; robotic grasping; industrial stacked scenarios

check for updates

Citation: Li, Y.; Guo, F.; Zhang, M.; Suo, S.; An, Q.; Li, J.; Wang, Y. A Novel Deep Learning-Based Pose Estimation Method for Robotic Grasping of Axisymmetric Bodies in Industrial Stacked Scenarios. *Machines* **2022**, *10*, 1141. https://doi.org/10.3390/machines10121141

Academic Editor: Antonios Gasteratos

Received: 24 October 2022
Accepted: 24 November 2022
Published: 1 December 2022

1. Introduction

Axisymmetric bodies are widely used in various industrial settings, such as screws in the assembly line, bobbins in the textile industry, sealing plugs in the sealing area, etc. However, these objects are usually stacked randomly in a disordered manner before processing, that is, they are in a stacked scenario, which is defined as a class of scenarios in which a large number of industrial products are placed in random positions to achieve stacking.

The intelligent robotic grasping system based on computer vision (Figure 1) plays an important role in realizing intelligent, flexible, and unmanned operations in the industrial manufacturing field. It can be applied to manufacturing tasks such as assembly, loading and unloading, and handling. Industrial stacked scenarios also require the robotic grasping system to be upgraded intelligently. The prerequisite of the intelligent robotic grasping system in industrial stacked scenarios is obtaining the 6D pose of the body to be grasped. Presently, the commonly used method for stacked scenarios is to manually arrange the objects in specific locations so that the robotic arms can grasp them. Industrial stacked scenarios have unique characteristics. This paper first analyzes these characteristics in detail, based on which the overall scheme is designed.

The characteristics of industrial stacked scenarios can be summarized as follows:

- Changeability. As the axisymmetric bodies are in contact with and support each other, the movement of one body usually causes other bodies to move, resulting in a change in the entire stacking scenario.
- Model information. The dimensions, geometric shapes, and other parametric information of the axisymmetric bodies are known.
- Ungraspability. Some bodies cannot be grasped, because the part to be grasped is shielded by other bodies.
- The number of products is large. Different categories of industrial products will not be mixed and stacked; hence, there is only one product category. Products are generally in an environment with a single working background, such as an assembly line; hence, there is only one background.

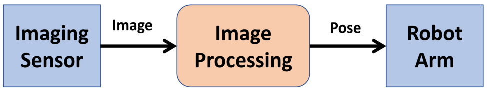


Figure 1. Intelligent robotic grasping system based on computer vision.

In this study, the advantages provided by the above characteristics were exploited to solve the difficulties in grasping using robotic hands in industrial settings. The specific solutions are as follows:

1. object detection based on deep learning is used to locate the component;
2. instance segmentation based on deep learning is used to select unshielded bodies that can be grasped;
3. conventional 6D object pose estimation methods have the characteristics of low efficiency or high cost of point cloud annotation, which are not suitable for stacked scenarios with a large number of objects. Therefore, we propose a novel method to obtain the 6D pose of the axisymmetric bodies to be grasped by detecting the pre-defined keypoints on the body surface; this approach requires multiobject 2D keypoint detection based on deep learning, and avoids the disadvantages of conventional methods.
4. the changeability of stacked scenarios requires (1), (2), and (3) to be carried out frequently to update the detection results. Therefore, to ensure efficient grasping, we integrated the three technologies involved in (1), (2), and (3) into one convolutional neural network (CNN) and set the real-time requirements. Fast detection can also extend our method to multiarm collaboration, camera in the arm, and other scenarios. More importantly, it is friendly to embedded devices with lower computing power due to fewer model parameters;
5. a large number of bodies usually results in more than one piece of graspable body being detected. Therefore, the grasping system requires a decision-making subsystem to create a grasping strategy to perform action ranking. In this study, a small CNN was used to score the quality of predictions of (2) and (3), which involves the intersection over union (*IoU*) of masks and object keypoint similarity (*OKS*) of keypoints, and these scores are used as the ranking criteria.

The contributions of this paper can be summarized as follows:

- a novel method is proposed to obtain the 6D pose of axisymmetric bodies to be grasped;
- a real-time multitask CNN is designed, named Key-Yolact;
- a small scoring CNN is designed to score the quality of the Key-Yolact prediction.

This paper is organized as follows. Section 2 presents an overview of the related concepts, namely, 6D object pose estimation, instance segmentation, and multiobject 2D pose estimation through keypoint detection, and a review of the associated literature. Section 3 explains the three contributions mentioned above. Section 4 reports the experiment conducted to verify the validity of the three contributions. Finally, Section 5 presents the conclusions.

2. Related Work

2.1. 6D Object Pose Estimation

According to the input data, this task can be classified into three types: RGB image as input, point cloud as input, and RGB-D image as input. Tulsiani and Malik [1] used a CNN to directly estimate the pose of an object from the RGB image. Another widely adopted method is to first predict the 2D keypoints through an RGB image and then obtain the pose of the object through perspective-n-point computation [2,3]. Further, orientation estimation using RGB images has also been reported [4]. Owing to the lack of depth information, the result of pose estimation is generally unsatisfactory. DOPE [5] used an RGB image and PnP algorithm for real-time multiple objects' pose estimation. For PnP, errors that are small in projection can be large in real 3D space [6]. In other words, the PnP method is not strong against noise. Therefore, this method is not suitable for the stacked scenarios where keypoint errors are likely to occur. Regarding point cloud input data, some researchers [7,8] have used a 3D CNN to predict the 3D boxes to estimate the pose of an object, but each frame had a delay of approximately 20 s. Further, most studies [9,10] have used a CNN structure similar to PointNet [11] to predict the pose. PoseCNN [12] uses the RGB-D data to directly perform pose estimation, but the post-processing has considerable time delay. The traditional PPF algorithm [13] can use the point cloud for multiobject pose estimation, but its speed is 2 s/obj. Segmentation is needed when the ICP algorithm [14] is used for multiobject pose estimation. Information fusion of RGB-D data [15,16] has also been applied in different ways to improve the estimation speed and accuracy. Some methods, such as DenseFusion [16] and PoseRBPH [17], which directly predict rotation and translation of objects, have poor generalization due to the non-linearity of the rotation space [6]. PVN3D [6] is based on 3D keypoints, and requires eight keypoints. In the industrial stacked scenarios, the points to be predicted will be close together or even overlap. This is not good for training neural networks. Unlike the above methods, our method obtains two 3D coordinates of the keypoints from the depth image after 2D keypoint detection, and then obtains the 6D pose of the object through post-processing in combination with the known model information, which requires RGB-D data.

2.2. Instance Segmentation

According to the structure of a neural network, instance segmentation can be classified as two-stage and one-stage. In the first stage of two-stage instance segmentation [18–20], the candidate regions-of-interest (ROIs) are generated; in the second stage, the ROIs are classified and segmented. Mask R-CNN [18] is a representative network of this type. FGN [19] introduced different guidance mechanisms into the various key components in Mask R-CNN to improve performance. Liu et al. [20] revealed the way that information propagates in neural networks is of great importance, and thus PANet is proposed to shorten the information path between lower layers and topmost feature. Owing to its structure, it is difficult to achieve real-time performance with two-stage instance segmentation. One-stage instance segmentation [21–25] is usually applied in the post-processing stage, such as cropping based on semantic segmentation. A representative technique is Yolact [21], which is the first real-time instance segmentation network. PolarMask [22] introduced an anchor-box free and one-stage method, which formulates the instance segmentation problem as the predicting contour of the instance through instance center classification and dense distance regression in a polar coordinate. Some one-stage methods generate position sensitive maps that combine semantic segmentation logits and direction prediction logits [23] or are assembled into final masks with position-sensitive pooling [24,25]. In general, the accuracy of two-stage instance segmentation is better than that of one-stage, but the inference speed is inferior. In this study, to filter out the objects that are shielded and hence ungraspable in the industrial stacked scenario, we identify the graspable objects according to the integrity of the mask obtained through instance segmentation. However, as described above, our algorithm should focus on the inference speed because of the characteristic of changeability in industrial stacked scenarios; hence, we adopted one-stage instance segmentation in our study.

2.3. Multiobject 2D Keypoint Detection

In this study, 6D object pose estimation was based on multiobject 2D keypoint detection, which can be classified as top-down and bottom-up. In the top-down approach [26–28], the objects are detected first, and then the keypoints are detected for each detected object. In the bottom-up approach [29–32], all the keypoints are first detected and then grouped into different objects. The extraction of keypoints can be performed using fully connected direct regression or a Gaussian heat map. The direct regression method loses spatial information and seriously damages the spatial generalization ability, resulting in high possibility of overfitting. The Gaussian heat map method retains the spatial information and has strong spatial generalization ability; hence, it shows better performance. The top-down method is generally considered more accurate whereas the bottom-up method is faster. In this study, based on the need for rapid detection, we used the bottom-up method.

According to demand analysis in the industrial stacked scenario, the CNN should simultaneously complete the three tasks of object detection, instance segmentation, and multiobject 2D keypoint detection. Keypoint R-CNN (Mask R-CNN with a keypoint detection branch) [18] can simultaneously complete all three tasks, but it does not meet the real-time demand.

3. Approaches

3.1. 6D Object Pose Estimation

The optical motion capture system estimates the object pose by obtaining the 3D coordinates of multiple "markers" attached to its surface; this approach is often used for motion capture of the human body and UAVs. Pose estimation in this system requires at least three marker points. With this method as the basis, this paper proposes the replacement of the maker points with computer vision to obtain the 3D coordinates. Specifically, multiobject 2D keypoint detection is used to detect the 2D keypoints on RGB images, and then the depth image is matched to obtain the 3D coordinates. The method is further improved to allow practical application. After improvement, only two 3D points are required for pose estimation. Our approach is described as follows.

In Figure 2, the points A, B, and C are three non-collinear points on a rigid body, which means

$$\overset{\rightarrow}{r_a^R} \times \overset{\rightarrow}{r_b^R} \neq 0 \tag{1}$$

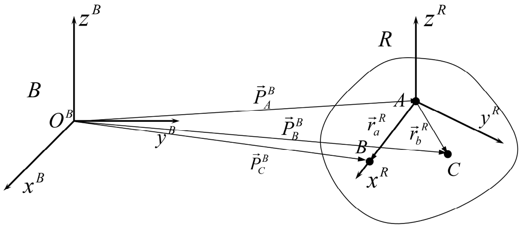


Figure 2. Pose of rigid body.

The coordinates of the three points in the base coordinate system B are given by the marker points or the RGB-D image. The coordinate system of a rigid body R can be established at any fixed position relative to the rigid body. To simplify the calculation, point A is selected as the origin of the coordinate system R, the unit vector of $\overset{\rightarrow}{AB}$ is the x-axis, the plane where the three points are located is the XOY plane of the coordinate system, and the remaining two axes are determined according to the right hand rule of the coordinate system. Then, the three base vectors of the coordinate system R—$\vec{i}$, $\vec{j}$, and $\vec{k}$—can be represented in the base coordinate system as the cross product between $\vec{r_a^R}$ and $\vec{r_b^R}$, as shown in Equation (2).

$$\vec{i} = \frac{\vec{r_a^R}}{\left|\vec{r_a^R}\right|} \quad \vec{k} = \frac{\vec{i} \times \vec{r_b^R}}{\left|\vec{i} \times \vec{r_b^R}\right|} \quad \vec{j} = \frac{\vec{k} \times \vec{r_a^R}}{\left|\vec{k} \times \vec{r_a^R}\right|} \tag{2}$$

Finally, according to R = TB, we can find the rotation matrix T, which is the orientation of the rigid body coordinate system. The position of the rigid body can be represented by the coordinates of the origin point A in the base coordinate system. Thus, the homogeneous transformation matrix representing the pose of the rigid body can be completely determined.

Theoretically, we can define multiple keypoints on a rigid body, and its pose can be obtained by detecting the coordinates of three keypoints that are non-collinear. However, in practical application, it is necessary not only to ensure non-collinearity, but also to maintain a certain distance between the two keypoints to ensure the accuracy of the vector direction. This is highly unsuitable for some products with a large aspect ratio, for example, the axisymmetric body discussed in this paper. Therefore, we employ the depth image and solve this application problem by detecting only two keypoints.

To obtain the pose of the rigid body coordinate system in the world coordinate system, we need to know the representation of the two basis vectors of the rigid body coordinate system in the world coordinate system. After the world coordinates of two 3D points are detected, only one of the basis vectors can be obtained. The method proposed in this paper uses the principal component analysis (PCA) method [33] to obtain the point cloud normal vector as another basis vector. The point cloud $(x_1, x_2, \cdots, x_i, \cdots, x_n)$ in Figure 3 consists of one of two 3D keypoints pre-defined on a rigid body and its n $-$ 1 nearest points. Point m is the centroid of the point cloud, and C is the fitted plane. The vector $\vec{n}$ is a normal vector to the plane C, which is used to obtain rigid body coordinates. The other arrows represent vectors $\vec{y_i}$. Point m and vectors $\vec{y_i}$ are used in the PCA method. The steps of the PCA method are detailed in Appendix A.

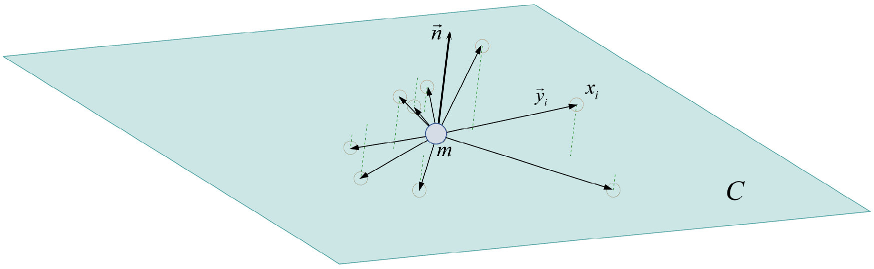


Figure 3. Fitting plane by PCA.

Next, the characteristics of axisymmetric bodies are analyzed. When an axisymmetric body rotates around its central axis, the RGB image and depth image taken in a fixed direction do not change as its pose changes. According to this characteristic, while defining the keypoints on its side surface, we can define dynamic keypoints on its generatrix; these keypoints do not rotate with the rotation of the rigid body around the central axis. Obviously, this leads to a non-unique pose of the rotating body, but only the angle of rotation of the body around its central axis is uncertain. Obviously, this uncertainty has no effect on the grasping and subsequent tasks of the manipulator.

The workflow of the novel pose estimation method proposed in this paper is shown in Figure 4. The first step is to obtain the 3D coordinates of the two keypoints. The 2D coordinates of keypoints are obtained from the RGB image by using 2D keypoint detection. Next, these two 2D keypoints obtain depth information by matching the corresponding depth image generated by the same RGB-D camera, i.e., two 3D keypoints are obtained. A normal vector that is non-collinear to the vector formed by two 3D key points is obtained by the PCA fitting plane method described above. Finally, through Equation (2) and coordinate transformation, the pose required for mechanical grasping can be obtained.

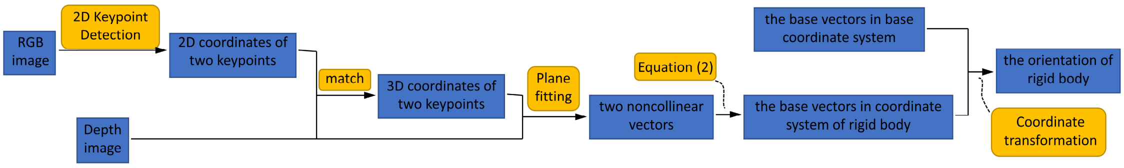

RGB image
2D Keypoint Detection
2D coordinates of two keypoints
Depth image
match
3D coordinates of two keypoints
Plane fitting
two noncollinear vectors
Equation (2)
the base vectors in base coordinate system
the base vectors in coordinate system of rigid body
the orientation of rigid body
Coordinate transformation

Figure 4. Novel pose estimation method.

3.2. Multitask CNN Key-Yolact

Mask R-CNN with a keypoint detection branch (Keypoint R-CNN) can simultaneously perform object detection, instance segmentation, and multiobject 2D keypoint detection with high accuracy, but it does not meet the real-time requirements of industrial stacked scenarios. Therefore, this paper proposes the real-time multitask CNN Key-Yolact.

3.2.1. Architecture

The common instance segmentation CNN can accomplish both object detection and instance segmentation. Our design approach is to add keypoint detection branches to the existing instance segmentation network. We selected Yolact, a real-time instance segmentation network, as the basic framework, and added the keypoint detection branch after modification. The network architecture is shown in Figure 5.

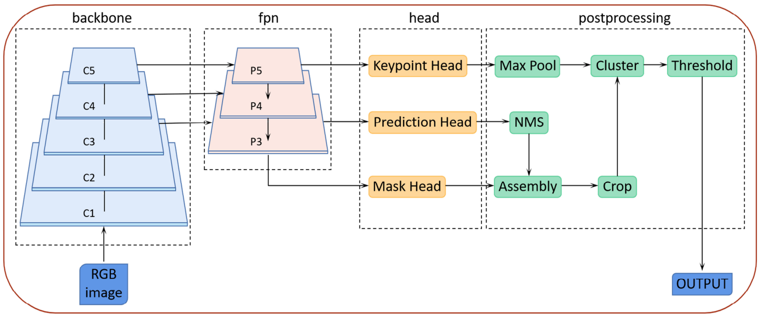


Figure 5. Key-Yolact architecture.

3.2.2. Object Detection and Instance Segmentation

After the features are extracted from the backbone network by using a feature pyramid network (FPN) [34], Yolact uses two parallel head networks, mask head and prediction head, to complete the task of instance segmentation. The first head network uses the FCN network [35] to generate the mask prototypes. The second head network completes the prediction of classes and detection boxes, and is similar to the head network of RetinaNet [36]. Yolact adds an additional branch on the prediction head to predict the mask coefficients. Finally, after non-maximum suppression (NMS), the linear combination of the mask coefficient and mask prototypes is cropped with the predicted bounding box to obtain the segmentation result of each instance.

Key-Yolact adopts ResNet-50 [37] as the feature backbone, and the C3, C4, and C5 layers are fused with the FPN method. As an industrial stacked scenario has a single product category, Key-Yolact does not need multiscale detection and uses only the P3 feature map as the input of the following two head networks to reduce the amount of calculation.

Object detection is performed using the prediction head in Figure 6. The prediction head architecture comprises the class, box, and mask branches. All convolution layers indicated by solid arrows have a 3 × 3 structure. In the figure, c is the number of classes, a is the number of anchors, and k is the number of mask prototypes, which was 32 in this study as in the case of Yolact. W and H are the width and height of the input image, respectively. For example, each pixel in the feature map, that represents classification and has a size of $W/8 \times H/8$, generates anchor boxes, and each anchor box generates c class confidences (including the background). Instance segmentation was performed using the mask head in Figure 7. The final convolution layer is 1 × 1, whereas the preceding layers are 3 × 3. All convolutional layers are followed by batch normalization and ReLU non-linear activations.

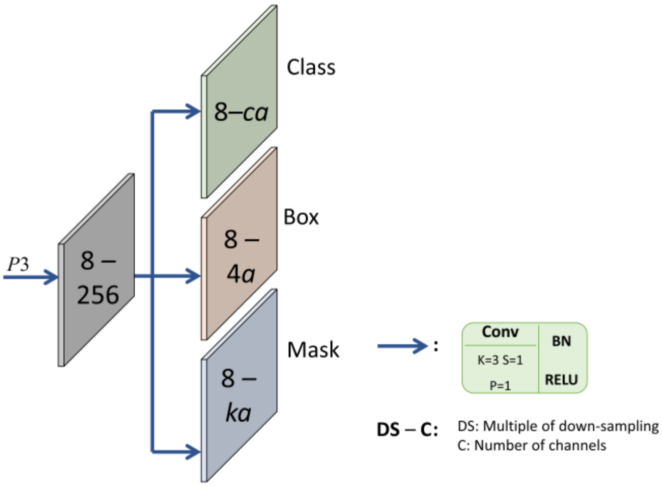


Figure 6. Prediction head in Key-Yolact.

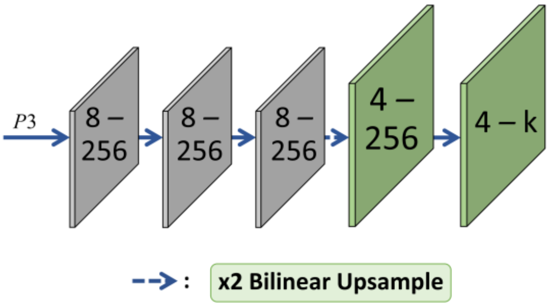


Figure 7. Mask head in Key-Yolact.

Key-Yolact adopts an anchor-based approach for object detection, with an aspect ratio of [1, 1/2, 2] and scale of 96. Regarding NMS, Key-Yolact adopts Fast NMS proposed by Yolact to improve the inference speed.

3.2.3. Multiobject Keypoint Detection

Multiobject keypoint detection mainly consists of two steps: (1) extracting keypoints from the entire feature map; (2) clustering the keypoints and dividing them into their respective instances.

Regarding keypoint extraction, the Gaussian heat map has better spatial generalization ability. The method first completes the conversion from 2D coordinates to heat maps. The pose estimation method proposed in this paper defines n keypoints ($n > 1$) for each axisymmetric body, and there are m objects ($m > 0$) in one RGB image. The coordinate of the keypoint i of object j in RGB is $(x_{i,j}, y_{i,j})$, $(0 < i \leq n, 0 < j \leq m)$, and its expression in the heat map i is shown in Equation (3).

$$H_i = exp\left(-\frac{\left(x - x_{i,j}\right)^2 + \left(y - y_{i,j}\right)^2}{2\sigma_{i,j}^2}\right) \tag{3}$$

σ represents the standard deviation of the Gaussian kernel, which controls the size of the Gaussian kernel.

As shown in Figure 8, the prediction point may appear at any location in the Gaussian kernel. If the size of the Gaussian kernel is not reasonable, the quality of keypoint extraction will be reduced. Therefore, it is necessary to control the size of the Gaussian kernel. In this study, *OKS* [38], which is the evaluation criterion for keypoint detection, was used to control the size of the Gaussian kernel. The *OKS* of each point in the Gaussian kernel is required to be greater than or equal to a certain threshold, which was 0.8 in this study.

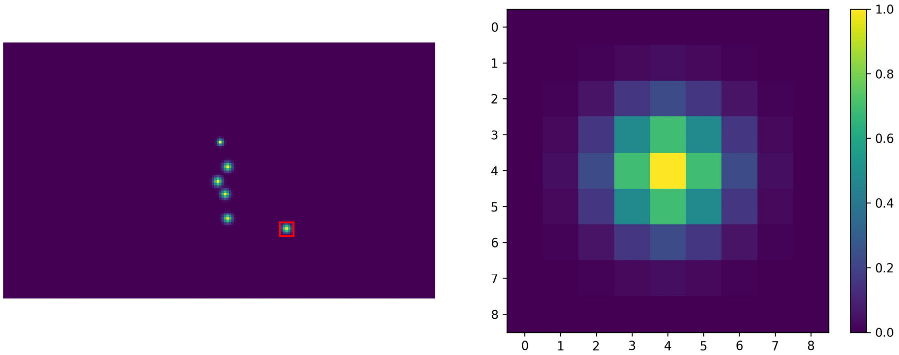


Figure 8. Example of Gaussian kernel. The image on the right is a partial enlargement of the red area in the image on the left.

OKS is defined as follows:

$$OKS = \frac{\sum_i \left[exp\left(-d_i^2/2s^2\kappa_i^2\right)\delta(v_i > 0)\right]}{\sum_i [\delta(v_i \ > 0)]} \tag{4}$$

The keypoint head architecture is shown in Figure 9. The feature map P5 is up-sampled by three transposed convolution layers and then goes through two convolution layers to obtain the predicted heat map. n is the number of keypoints defined on each body.

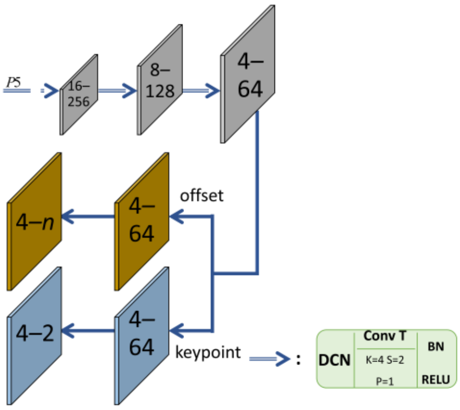


Figure 9. Keypoint head in Key-Yolact.

After the keypoints are extracted, they need to be correctly divided into their respective instances. Key-Yolact uses the mask predicted by the instance segmentation to crop the heat map of the keypoints, and the keypoints in the mask belong to the instance represented by the mask. Obviously, this approach is not suitable for dividing the keypoints with visibility flag of 1, but in an industrial stacked scenario, the dataset is not marked with instances of $v = 1$ because such objects are shielded and cannot be grasped.

3.2.4. Improvements

Based on three head networks, some improvements were made to enhance the accuracy and inference speed of the network [39].

The accuracy of keypoint detection directly determines the accuracy of body pose estimation. Therefore, the accuracy was improved by using deformable convolution [40] and adding the regression branch of the offset to the keypoint head [41]. Deformable convolution adds learnable offsets to the convolution kernel to augment the spatial sampling locations, thus improving the generalization ability of the model. Key-Yolact replaces the convolution layer in the keypoint detection head network with deformable convolution to improve the accuracy of keypoint detection.

In the keypoint detection head network, the heat map predicting the keypoints is down-sampled four times relative to the input image, which means that one pixel in the feature map corresponds to four pixels in the original image, which will undoubtedly reduce the prediction accuracy of the keypoints. To solve this problem, Key-Yolact adds a regression branch of the offset to the keypoint head.

The offset branch is similar to the keypoint branch, which is shown in Figure 9. However, the pixel of the output feature map does not represent the probability of the keypoints but the offset of this location. The number of channels was fixed as 2, representing the X and Y directions. The ground truth of the offset is

$$offset = \frac{p}{R} - \widetilde{p} \tag{5}$$

where p is the original coordinate, R is the down-sampling multiple, and $\widetilde{p}$ is the approximate integer coordinate after down-sampling.

TensorRT is an inference optimizer developed by Nvidia, which improves the inference speed of a neural network by reducing the accuracy and fusing the tensor and layer. TensorRT was used to accelerate the backbone network of Key-Yolact in this study.

3.2.5. Loss Function

In a multitask CNN, each task branch needs a corresponding loss function. The total loss function adopted in Key-Yolact is shown in Equation (6). Each loss function and weight are shown in Table 1. The weight values of different loss functions are used to express each loss function value in the same order of magnitude.

$$L_{total} = \lambda_{class} L_{class} + \lambda_{box} L_{box} + \lambda_{mask} L_{mask} + \lambda_{kp} L_{kp} + \lambda_{offset} L_{offset} \tag{6}$$

Table 1. Loss function.

Branch	**Loss Function/L**	**Weight/λ**
Class	Cross Entropy Loss	1
Box	Smooth L1 Loss	1.5
Mask	BCE Loss	6.125
Keypoint	Focal Loss	1
Offset	L1 Loss	1

All loss functions except for the keypoint loss function are in common use in their respective tasks. As shown in Figure 8, the proportion of positive and negative samples in the heat map of the keypoints is considerably unbalanced. To solve this problem, this study draws on the principle of focal loss and uses the loss function shown in Equation (7).

$$L_{kp} = \frac{-1}{N}\sum_{xyc}\begin{cases} \left(1-\hat{Y}_{xyc}\right)^{\alpha} log\left(\hat{Y}_{xyc}\right) & Y_{xyc}=1 \\ \left(1-Y_{xyc}\right)^{\beta}\left(\hat{Y}_{xyc}\right)^{\alpha} log\left(1-\hat{Y}_{xyc}\right) & otherwise \end{cases} \tag{7}$$

3.3. Decision-Making Subsystem for Multiobject Grasp

In industrial stacked scenarios, multiple graspable bodies can usually be detected. The grasping system needs a strategy to sort the grasping operations for the graspable bodies. The strategy adopted in this study is to grasp the object with high prediction quality first.

A small CNN was designed to score the prediction results of Key-Yolact. The object with the highest score was grasped first. Its architecture is shown in Figure 10. The network is composed of five 3 × 3 convolution layers and two fully connected layers, where the stride of the convolution layer is 2.

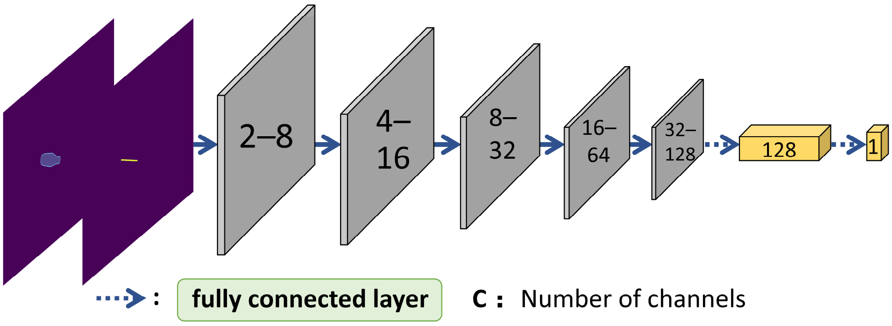


Figure 10. Scoring network architecture.

The input of this network is a simple fusion of the prediction results of instance segmentation and keypoint detection. It has two channels: the first channel is the mask of the object that can be grasped, and the second channel comprises the keypoints and their connections, as shown in Figure 10.

The ground truth of the score consists of two parts. One part is the *IoU* between the predicted mask and its ground truth, and the other part is the *OKS* of the predicted keypoints and its ground truth. The loss function used in training is the mean squared error. The ground truth is expressed as shown below.

$$GT = IoU + 2 \times OKS \tag{8}$$

4. Experiments

4.1. Self-Built Dataset

In this study, a stacked scenario dataset of bobbins in the textile industry was built. Inspired by Image Net, the dataset contained 3000 images as the training set and 600 images as the validation set, and every image in the dataset had a size of 396 × 704. Image Net contains 14,197,122 images and 21,841 synsets indexed. On average, each synset contains 650 images. Thus, the self-built dataset has a sufficient number of images. A single object category and background in an industrial stacked scenario are the main reason for the low cost of the dataset and small number of images required. The bobbin is a typical axisymmetric body; hence, two dynamic keypoints are defined on its side surface.

4.2. Experiments on Key-Yolact

4.2.1. Training and Loss

Key-Yolact was trained with a batch size of 4 on an Nvidia GeForce RTX 1080Ti using CUDA and cuDNN for about 17 h. Thus, the hardware required for this network is affordable. The training was conducted with stochastic gradient descent (SGD) for 100 epochs, starting at an initial learning of 1×10^{-2} and dividing by 10 at epochs 30, 60, and 90 using a weight decay of 5×10^{-4} and momentum of 0.9. Horizontal and vertical flip augmentation was used.

The change in total loss and loss at each branch in the training process are shown in Figure 11. It is obvious that Key-Yolact can completely converge.

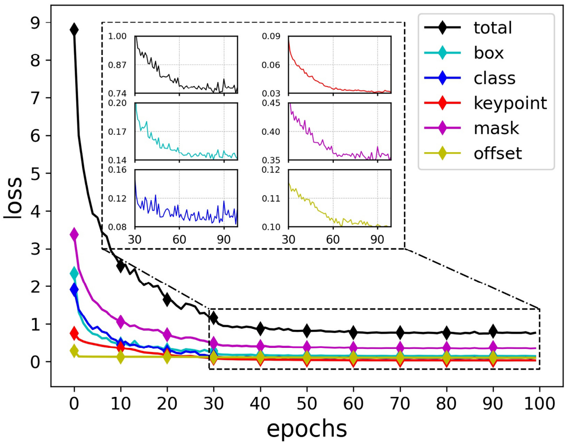


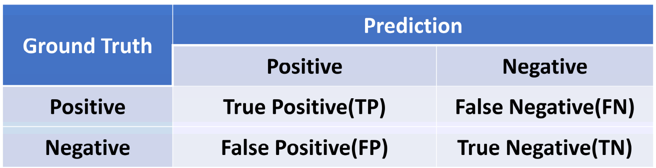
Figure 11. Loss of Key-Yolact.

4.2.2. Evaluation Metric

Generally, the mean average precision (mAP) is used as the evaluation metric for object detection, instance segmentation, and keypoint detection, and is jointly determined using the precision and recall. The precision (P) and recall (R) are defined below.

$$\begin{cases} \mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \\ \mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \end{cases} \tag{9}$$

TP, FP, and FN in Equation (9) are defined in Figure 12. The positive and negative in object detection and instance segmentation are divided by the *IoU*, and those in keypoint detection are divided by the *OKS*. The threshold used in this study was 0.5.



Ground Truth	Prediction	
	Positive	Negative
Positive	True Positive(TP)	False Negative(FN)
Negative	False Positive(FP)	True Negative(TN)

Figure 12. Confusion matrix.

Precision and recall are a pair of contradictory metrics. When one is high, the other is generally low. In the absence of specific application scenarios, mAP is used to comprehensively measure the network model. However, industrial stacked scenarios should focus on whether the objects detected are positive, and not whether all positive objects can be detected, because only one body can be grasped at a time. Therefore, precision can be used as the evaluation metric of Key-Yolact. On the contrary, it is more reasonable to choose recall as the evaluation metric in a defect detection scenario.

4.2.3. Analysis of Results

To evaluate Key-Yolact without using TensorRT, we compared it with Keypoint R-CNN in terms of precision, inference time, and number of parameters. The results are shown in Table 2, and Figure 13 shows a prediction example.

Table 2. Contrast and ablation experiment results.

	Precision	Run Time/ms	FPS	Parameters/*M*
Keypoint R-CNN	79.68%	88.59	11.3	58.8
Key-Yolact	72.61%	47.26	21.29	38.8
Key-Yolact with TensorRT	72.61%	35.39	28.1	/
Key-Yolact without DCN	70.36%	44.89	22.3	35.9
Key-Yolact without offset branch	51.54%	44.92	22.3	35.9

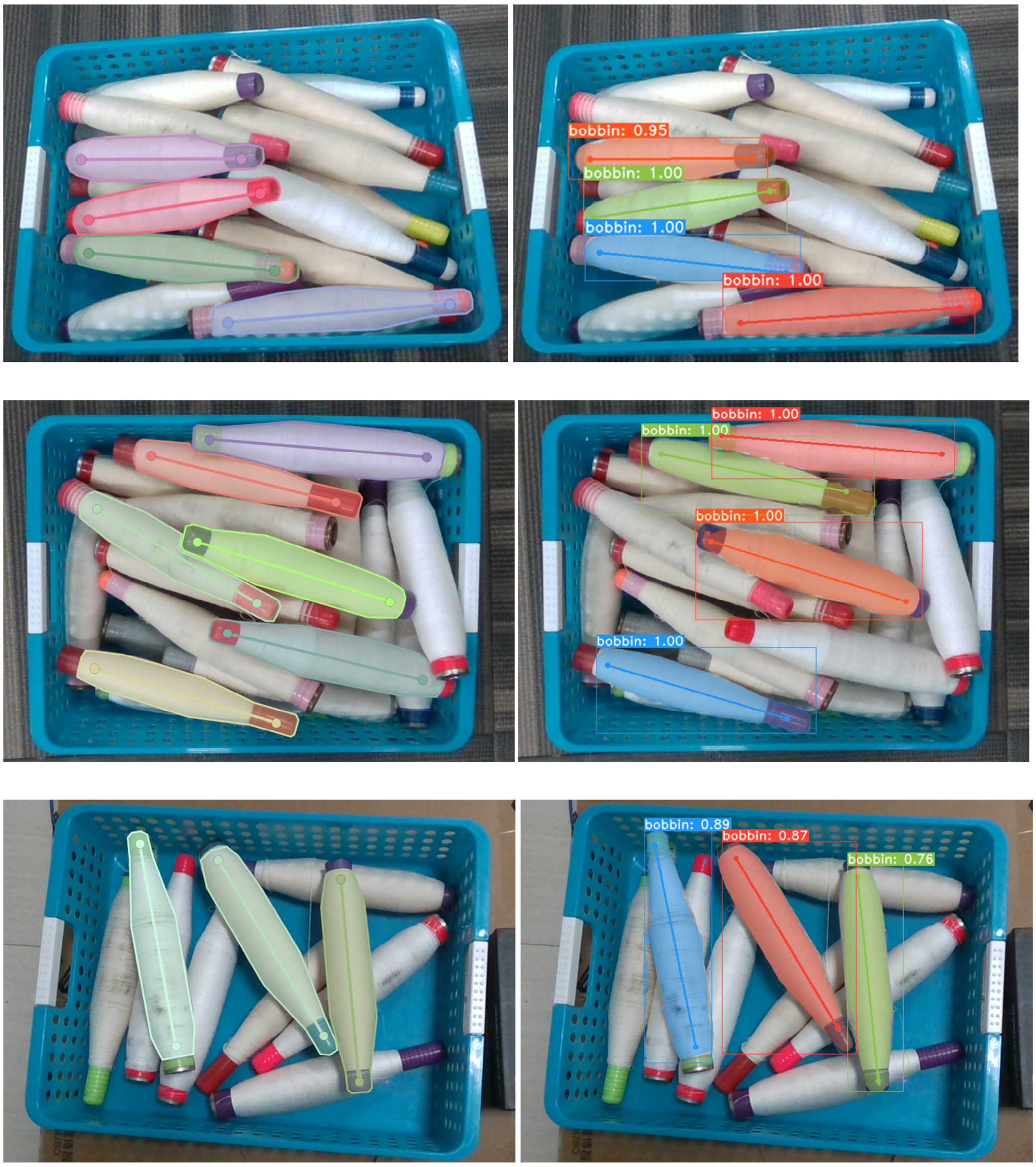


Figure 13. Results for validation set in self-built dataset. The images on the left show the ground truth, and those on the right show the prediction. The prediction in the first row was perfect. Two objects were not detected due to occlusion in the second row. The class confidences of the prediction were slightly lower, because there were slightly fewer images like this in the dataset, where the bobbins were sparsely distributed.

We found that Key-Yolact had slightly lower accuracy (~7% lower) than Keypoint R-CNN, but the inference speed was almost doubled, approximately 10 FPS or higher, and the number of model parameters was significantly reduced by 20 *M*. Despite the slightly lower accuracy, the high inference speed makes Key-Yolact more valuable than Keypoint R-CNN for industrial applications.

Next, to verify the effectiveness of the improvement described in Section 3.2.4, we conducted an ablation experiment on Key-Yolact, and the results are shown in Table 2.

By using TensorRT to accelerate the backbone network, Key-Yolact is fast enough for real-time application (28 FPS, on an Nvidia GeForce RTX 1080Ti). Moreover, the drop in accuracy was almost negligible. Deformable convolution and offset branching require less time, approximately 2.5 ms, but they improve the precision by 2.25% and 21.07%, respectively. Obviously, an offset branch is essential, which greatly improves the precision.

In conclusion, Key-Yolact has a significant advantage in terms of the inference speed, whereas it is slightly less accurate than the classical network. High inference speed is critical for industrial stacked scenarios with changeability. Therefore, Key-Yolact can be applied to industrial stacked scenarios.

4.3. Scoring Network-Related Experiments

The predictions of Key-Yolact on the self-built dataset were divided into the training and validation sets in the ratio of 5:1. We trained with SGD for 80 epochs, starting at an initial learning of 5×10^{-2} and dividing by 10 at epoch 50 using a weight decay of 5×10^{-4} and momentum of 0.9.

The validation set was sorted according to the ground truth, and the curve of the ground truth and predicted value was drawn, as shown in Figure 14. It was found that the predicted value could fit the ground truth well, and the mean square error was 0.0826. This proves that the scoring network can accurately perform the regression and is effective. Figure 15 shows the scoring results for the example in Figure 13.

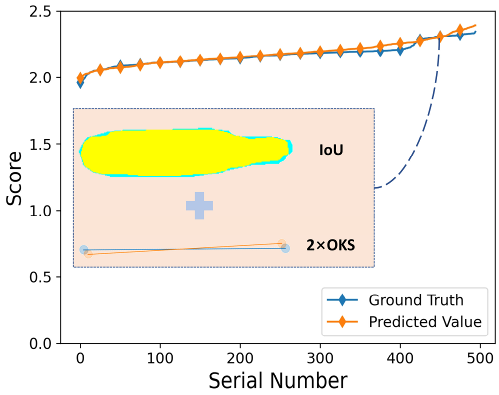


Figure 14. Prediction of scoring network on the validation set.

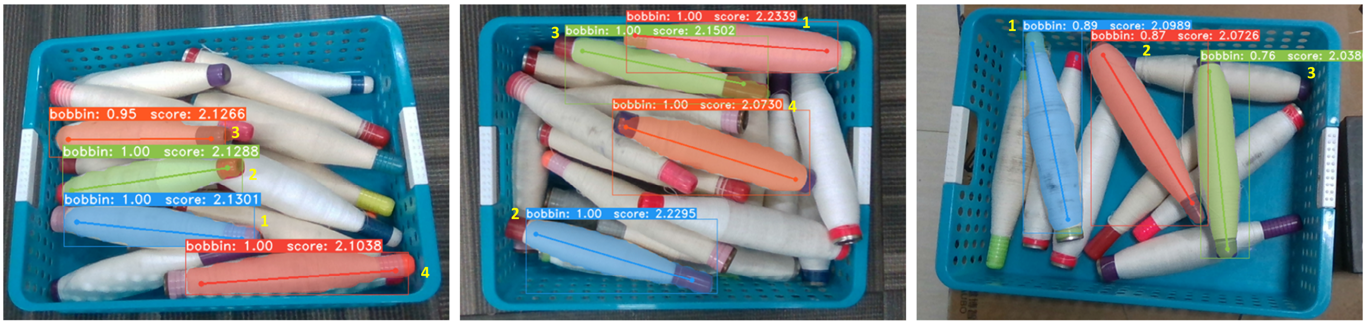

bobbin: 0.95 score: 2.1266
3
bobbin: 1.00 score: 2.1288
2
bobbin: 1.00 score: 2.1301
1
bobbin: 1.00 score: 2.1038
4
bobbin: 1.00 score: 2.2339
1
3
bobbin: 1.00 score: 2.1502
bobbin: 1.00 score: 2.0730
4
2
bobbin: 1.00 score: 2.2295
1
bobbin: 0.89 score: 2.0989
bobbin: 0.87 score: 2.0726
2
bobbin: 0.76 score: 2.038
3

Figure 15. Scoring results for the images in Figure 13.

4.4. Robotic Grasping Experiment

To prove the rationality of the pose estimation method proposed in this paper, we used the AUBO-i3 robot to conduct a grasping experiment. The robot grasped two boxes containing 25 randomly placed bobbins, which fell freely into the container from a height of 0.8 m. The bobbin that the robot failed to grasp was removed manually. The grasping success rates for the first and second boxes were 88% and 92%, respectively, and the average success rate was 90%. This shows that the design approach of this paper is effective and can be directly applied to industrial scenarios. This high success rate is also due in part to the fault tolerance of the gripper to the body's pose accuracy. The diameter of the grasping part on the bobbin was 18 mm, and the maximum distance between the two fingers of the gripper was 22 mm. The way to grasp the bobbin is shown in Appendix B.

The reason for most grasping failures is that when grasping an object at the edge of the box, the gripper and box collide, resulting in failure in grasping. In the future, we will consider improving the mechanical design of the gripper to solve this problem.

5. Conclusions

In this study, the characteristics of industrial stacked scenarios and axisymmetric bodies were analyzed. The difficulties in perception due to changeability and ungraspablity of axisymmetric bodies in industrial stacked scenarios were solved affordably by using Key-Yolact. The novel method to obtain the 6D pose of an object, proposed in this paper, essentially replaces the approach of directly obtaining a homogeneous transformation matrix with that of directly obtaining the keypoint coordinates, which ensures the simplicity of the CNN architecture and low cost of dataset annotation. Key-Yolact sacrifices the accuracy to a small extent to achieve fast inference, although its accuracy is sufficient for industrial stacked scenarios. However, it is not particularly suitable for scenarios that focus on detection accuracy. The decision-making subsystem in the multiobject grasping scenario solves the problem of determining the grasping order. The scoring network is able to perform a secondary evaluation of the detection results, which would be useful in some cases for optimal selection. In conclusion, the proposed approach is expected to be immensely helpful in the intelligent grasping of axisymmetric bodies in industrial stacked scenarios.

The proposed method has two limitations. Firstly, this method cannot deal with unknown and untrained objects. However, for untrained objects, if the backbone in Key-Yolact (i.e., ResNet-50) loads pre-training weights for transfer learning after the production of datasets, as shown in Figure 11, the model can converge easily. Secondly, we only discuss the axisymmetric body. This does not mean that the proposed method will not work for other geometries, but we have not found a suitable unified framework to define the keypoints yet. That is what we aim to do in the future.

Author Contributions: Conceptualization, S.S. and Y.L.; methodology, F.G. and Y.L.; software, Y.L.; validation, Y.L.; formal analysis, Y.L.; investigation, Y.L. and M.Z.; resources, Y.L. and J.L.; data curation, Y.L. and M.Z.; writing—original draft preparation, Y.L. and J.L.; writing—review and editing, Q.A. and Y.W.; visualization, Y.L. and Q.A.; supervision, F.G. and Y.W.; project administration, F.G. and Y.W.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The steps of PCA adopted in this paper are as follows:

Firstly, centroid m and corresponding vectors $\overrightarrow{y_i}$, which form the matrix Y are calculated.

$$\begin{cases} m = \frac{\sum_{i=1}^{n} x_i}{n} \\ \overrightarrow{y_i} = x_i - m \\ Y = \left[\overrightarrow{y_1} \ \overrightarrow{y_2} \ \cdots \ \overrightarrow{y_i} \ \cdots \ \overrightarrow{y_n} \right] \end{cases} \tag{A1}$$

Then, singular value decomposition of the matrix Y is performed:

$$Y = U\Sigma V^T \tag{A2}$$

The last column vector $\overrightarrow{n}$ of U is the normal vector of the plane C.

Appendix B

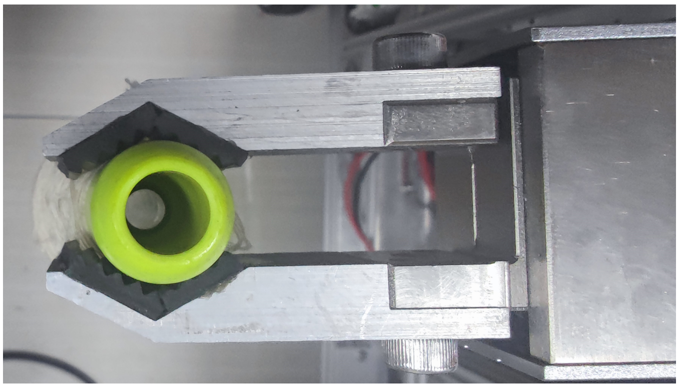


Figure A1. Way to grasp the bobbin.

References

1. Tulsiani, S.; Malik, J. Viewpoints and Keypoints. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
2. Brachmann, E.; Krull, A.; Michel, F.; Gumhold, S.; Shotton, J.; Rother, C. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In Proceedings of the European Conference on Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014.
3. Tekin, B.; Sinha, S.N.; Fua, P. Real-Time Seamless Single Shot 6D Object Pose Prediction. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
4. Sundermeyer, M.; Marton, Z.C.; Durner, M.; Triebel, R. Augmented autoencoders: Implicit 3d orientation learning for 6d object detection. *IJCV* **2020**, *128*, 714–729. [CrossRef]
5. Tremblay, J.; To, T.; Sundaralingam, B.; Xiang, Y.; Fox, D.; Birchfield, S. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv* **2018**, arXiv:1809.10790.
6. Yisheng, H.; Wei, S.; Haibin, H.; Jianran, L.; Haoqiang, F.; Jian, S. PVN3D: A Deep Point-Wise 3D Keypoints Voting Network for 6DoF Pose Estimation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
7. Song, S.; Xiao, J. Sliding Shapes for 3D Object Detection in Depth Images. In Proceedings of the European Conference on Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014.
8. Song, S.; Xiao, J. Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
9. Qi, C.R.; Wei, L.; Wu, C.; Hao, S.; Guibas, L.J. Frustum PointNets for 3D Object Detection from RGB-D Data. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
10. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
11. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017.

12. Yu, X.; Schmidt, T.; Narayanan, V.; Fox, D. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In Proceedings of the Robotics: Science and Systems—RSS 2018, Pittsburgh, PA, USA, 26–30 June 2018.
13. Bertram, D.; Markus, U.; Nassir, N.; Slobodan, I. Model Globally, Match Locally: Efficient and Robust 3D Object Recognition. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
14. Paul, J.B.; Neil, D.M. A Method for Registration of 3-D Shapes. *IEEE T-PAMI* **1992**, *14*, 239–256.
15. Xu, D.; Anguelov, D.; Jain, A. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
16. Wang, C.; Xu, D.; Zhu, Y.; Martín-Martín, R.; Lu, C.; Li, F.F.; Savarese, S. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
17. Xinke, D.; Arsalan, M.; Yu, X.; Fei, X.; Timothy, B.; Dieter, F. PoseRBPF: A Rao-Blackwellized Particle Filter for 6D Object Pose Tracking. *arXiv* **2019**, arXiv:1905.09304.
18. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision(ICCV), Venice, Italy, 22–29 October 2017.
19. Fan, Z.; Yu, J.G.; Liang, Z.; Ou, J.; Gao, C.; Xia, G.S.; Li, Y. Fgn: Fully guided network for few-shot instance segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
20. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
21. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact: Real-time instance segmentation. In Proceedings of the 2019 IEEE International Conference on Computer Vision(ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
22. Xie, E.; Sun, P.; Song, X.; Wang, W.; Luo, P. PolarMask: Single Shot Instance Segmentation With Polar Representation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
23. Chen, L.C.; Hermans, A.; Papandreou, G.; Schroff, F.; Wang, P.; Adam, H. Masklab: Instance segmentation by refining object detection with semantic and direction features. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
24. Dai, J.; He, K.; Li, Y.; Ren, S.; Sun, J. Instance-sensitive fully convolutional networks. In Proceedings of the European Conference on Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 8–16 October 2016.
25. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully convolutional instance-aware semantic segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017.
26. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE T-PAMI* **2021**, *43*, 3349–3364. [CrossRef] [PubMed]
27. Insafutdinov, E.; Pishchulin, L.; Anres, B.; Anrriluka, M.; Schiele, B. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In Proceedings of the European Conference on Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 8–16 October 2016.
28. Pishchulin, L.; Insafutdinov, E.; Tang, S.; Anres, B.; Anriluka, M.; Gehler, P.; Schiele, B. Deepcut: Joint subset partition and labeling for multi person pose estimation. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
29. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE T-PAMI* **2021**, *43*, 172–186. [CrossRef] [PubMed]
30. Kocabas, M.; Karagoz, S.; Akbas, E. MultiPoseNet: Fast Multi-Person Pose Estimation Using Pose Residual Network. In Proceedings of the European Conference on Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018.
31. Pishchulin, L.; Jain, A.; Anriluka, M.; Thormahlen, T.; Schiele, B. Articulated people detection and pose estimation: Reshaping the future. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
32. Gkioxari, G.; Hariharan, B.; Girshick, R.; Malik, J. Using k-Poselets for Detecting People and Localizing Their Keypoints. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
33. Hoppe, H.; Derose, T.; Duchamp, T.; Mcdonald, J.; Stuetzle, W. Surface Reconstruction from Unorganized Points. *ACM Siggraph* **1992**, *26*, 71–78. [CrossRef]
34. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017.
35. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE T-PAMI* **2017**, *39*, 640–651.
36. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *IEEE T-PAMI* **2020**, *42*, 318–327. [CrossRef] [PubMed]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.

38. Tsung-Yi, L.; Michael, M.; Serge, B.; James, H.; Pietro, P.; Deva, R.; Piotr, D.; Lawrence, Z. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014.
39. Courbariaux, M.; Bengio, Y.; David, J.P. Training deep neural networks with low precision multiplications. *arXiv* **2014**, arXiv:1412.7024.
40. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision(ICCV), Venice, Italy, 22–29 October 2017.
41. Zhou, X.; Wang, D.; Krhenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.