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

# Three-dimensional Localization of Convolutional Neural Networks Based on Domain Randomization

Guoyu Zuo, Chengwei Zhang, Tao Zheng, Qingshui Gu and Daoxiong Gong
*Faculty of Information Technology*
*Beijing University of Technology*
*Beijing 100124, P. R. China*
*zuoguoyu@bjut.edu.cn*

***Abstract*— In this paper, we propose a 3D positioning framework based on convolutional neural network to realize the pose estimation of the target object during the mechanical arm grabbing process. First, the image data for training this network is completely synthesized by domain randomization technology to reduce the data acquisition cost and make up the gap between the simulated image and the real image. Then, the 3D positioning framework is designed based on the ResNet architecture, which consists of two parts. One is to obtain the classification of target object, and the other is to obtain the 3D coordinates of the object and the horizontal rotation angle. Finally, the target image classification and pose estimation are tested on the real images with interference and the synthetic images with interference, respectively. The results show that the proposed method has high prediction accuracy, and the effectiveness of the framework is proved by migrating it to the real physical robot arm experimental platform.**

***Index Terms*— domain randomization, three-dimensional localization, ResNet, sim2real.**

## I. Introduction

The manipulator grabbing tasks are regular operations in modern industrial manufacturing, and diverse grasping strategies are used for different grabbing tasks. For a target object of arbitrary pose, it is a great challenge in complex scenes to realize accurate classification and three-dimensional positioning by using image information collected by the camera. The computer vision technology is an important approach applied in intelligent manufacturing to improve the success rate of the manipulator grabbing tasks.

Traditional methods based on computer vision for three-dimensional positioning of target objects include manual features and template matching such as SIFT and HOG transform [1-4]. When dealing with blur objects or the objects without texture, the manual features positioning method is less effective, and the template matching is easily affected by the lighting conditions. In order to overcome the influence of illumination conditions, The 3D point cloud methods are proposed to randomly extract 3D feature points. The classical Point Pair Feature (PPF) method [5] relies on the extraction and indexing of point-to-point features, and utilizes the 3D point cloud features of the object surface to construct the point-to-point features. And the extracted point-to-point features are compared with the dataset to estimate the target object pose by using Hough voting and clustering. Since the method needs to acquire the 3D point cloud features of the object in advance, it has the problem of high data acquisition cost.

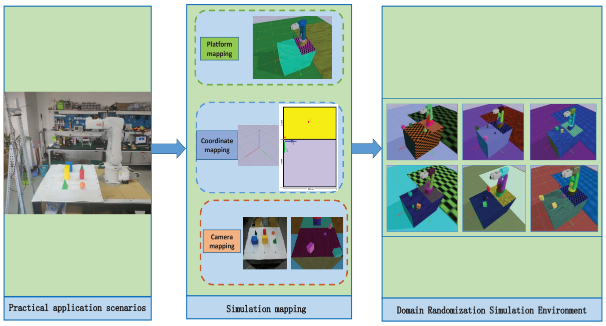
Platform mapping
Coordinate mapping
Camera mapping
Practical application scenarios
Simulation mapping
Domain Randomization Simulation Environment

Fig. 1. Schematic diagram of the simulation environment.

The target positioning method based on RGB image needs to find the position of the target on the 2D image, and then solves the three-dimensional coordinate position of the target object with respect to the camera by PnP [6]. However, the locating result is not too accurate and the stability is not too good. In response to the above problems, the method based on RGB-D images is proposed. Hinterstoisser [7] proposed a template matching method suitable for both color and depth. Rios et al. [2] used discriminative learning and cascade detection extension work to achieve higher accuracy and efficiency. Brachmann et al. [8] used the regression forest method for RGB-D images to segment and locate the target, which improves the accuracy and efficiency of three-dimensional positioning. But the RGB-D image data has higher acquisition cost.

For the problem of high data acquisition costs, it is convenient and cheap to collect training image data by using the simulation environment. However, it is difficult to completely reproduce the real scenes in the simulation environment due to the lighting conditions in the real environment and the

*This work is supported by National Natural Science Foundation #61873008 and Beijing Natural Science Foundation #4182008, #4192010

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

surface texture of the object. Therefore, there is a big gap between real and simulation images. One way to reduce the difference between simulation and reality image data is to render images using a high-fidelity simulation environment. For example, Yair et al. [9] used a high-simulation composite image for target viewing angle evaluation, but the method has poor performance in a complex scene and requires high computing resources, and it needs high quality of the object model. The domain randomization technique is another method to reduce the difference between the simulation-reality image data, which has low requirements of the fidelity of the simulation environment. The technology has been successfully applied to four-rotor aircraft learning automatic obstacle avoidance [10]. [11-13] used domain randomization to obtain the grab target and Tremblay et al. [14] used domain randomization in CAR detection. [15] used domain randomization in the target direction estimation of industrial part shapes. Josh Tobin et al.

Domain randomization randomly generates various types of factors in the simulation environment, by which the gap between the real images and simulation images is narrowed and the cost of collecting real image data is greatly reduced. This paper proposes a target three-dimensional positioning framework for target object classification and three-dimensional positioning. First, a three-dimensional positioning framework of convolutional neural network is designed based on ResNet architecture. A RGB image dataset is then generated by using the domain randomization technique. The proposed model is trained by using the dataset as input data. We test the model by using real-image with obstruction, and migrate the model to the physical ABB IRB1200 robotic arm to verify its robustness.

## II. Establishment of domain randomization data set

In this section, the training data is generated by domain randomization which is used to create synthetic simulation data with multiple factors such as color, texture, illumination and other factors. When tested on the real image data, the scene information of the object is regarded as a randomized type. And the trained model focuses only on the objects' shape, and then performs classification and positioning tasks.

### A. Simulation environment construction

Bullet simulation engine is used for data generation, which can load 3D models in URDF, OBJ, SDF and other formats. Based on Bullet simulation environment, SolidWorks, 3dmax and other 3D object modeling software, it can construct a simulation environment similar to the real situation. Firstly, the whole simulation experimental platform is simulated by the same size as the real physical. And the texture, illumination angle and illumination intensity of the experimental platform in the simulation environment are randomly generated. Secondly, the coordinate origin of the simulation experiment platform is set and the world coordinate system is established. Finally, an analog camera is set up in front of the simulation environment experimental platform as a tool for collecting images. The images collected by the real and analog cameras are shown in the simulation environment built in Fig. 1.

We use 3D SolidWorks to collect the target and interferer models. The collected target object models are imported into the simulation environment. Then, the target object is programmed to control the feature properties randomly, such as scale, shape, and texture.

### B. Domain randomization

The key idea of domain randomization is to generate training data sets by changing factors in the image space and train the model by the data to make the model robust to the appearance and illumination of the object and the environment. Specifically, the following factors are randomly generated in the image: texture of workbench and manipulator; horizontal angle, size, position and texture of the interferer; position, horizontal rotation angle and texture of target object; illumination intensity and direction; background texture.

*1) Content Variation:* A random number of objects are randomly placed in a 3D background with random orientation. To achieve variations in shape and size, six different types of block models were used, and the dimensions of the model are randomly perturbed in a small range to provide finer geometric variations. The 3D interferer is a randomly generated polygonal geometric object, which is placed in the scene to simulate the appearance of a not-interested object in the scene.

*2) Style Variation:* Style variation are obtained by randomly changing the color and texture of all objects. We use the texture library with 8 different textures to realize style variation. In addition, light enhancement is applied to capture varying shadow conditions and temporal variations. The changes of illumination conditions are realized by changing the position and orientation of the light source.

*3) Camera Variation:* In reality, the position of the camera is not be exactly the same as that of the camera in the simulation environment,and the position error is caused when migrating to the real physical model. In order to reduce the error between the simulation and the reality due to the inconsistent camera position, the real environment uses ZED camera to acquire the image data. When the image data is collected in the simulation environment, the placement position of analog camera has 1.5 cm random jitter in four directions, the shooting angle has $1.5^{\circ}$ random in three directions and the field of view has a random size of 3.5%.

### C. data collection

The collection process of data set includes image data collection and image data label information annotation. When generating image data by the physics engine, the acquisition cost is extremely low with about 700 pictures obtained every minute. The dataset contains 57600 images that are collected in the simulation environment. The target objects in the data set are 9600 images of cubes, spheres, cuboids, cylinders, cones and pyramids. The label information of each image contains 3 center coordinates of the target object, 24 vertex coordinates surrounding the 3D bounding box of the target object, 1 horizontal rotation angle, and 1 target object category. Their units are meters and degrees. For the rotationally symmetric objects such as sphere, cylinder, and cone, the horizontal rotation angle is uniformly set to 0.

## III. Target 3D positioning network architecture

When the grab task is performed, the grasping strategy is designed according to the horizontal rotation angle and the contour size of the target object, which is to improve the success rate of the robot arm to grasp the target object. Fig. 2 shows a three-dimensional positioning network structure of the target object, which can realize classification and target object 3D bounding box and its horizontal rotation angle prediction. YOLO-6D [16] predicts the position of the 8 vertices surrounding the 3D bounding box of the target object on the 2D image. Different from YOLO, the 3D bounding box and the target coordinate prediction module directly predicts the three-dimensional coordinates of the 8 vertices surrounding the 3D bounding box and a relatively inaccurate three-dimensional coordinate of the target object. And then, the more accurate target object coordinates and horizontal rotation angles are obtained through a fully connected network.

### A. *Target classification*

As the number of network layers increases, the error rate of residual neural network training is continuously decreasing. The target classification is to process the input RGB images by using the ResNet network-based convolutional neural network. Finally, the full-connection layer Softmax classifies the six kinds of targets and outputs the probability distribution as follows:

$$Z^{'(k)} = softmax(a^{'(k)}) = e^{a^{'(k)}} / \sum_{i=1}^{n} e^{a^{'(k)}} \tag{1}$$

where, $a^{'}$ is the predicted value of the neural network output, $n$ is the number of categories of the target classification. The formula outputs the probability value of the category, mapping the output of the neuron between (0, 1), and the maximal probability value of classification is selected as the final output. The cross entropy is used as the loss function of the classification model as follows:

$$loss(Z^{'}, Z) = 1/m \sum_{i=1}^{m} \sum_{j=1}^{n} Z_i^{(j)} log \; Z_i^{'(j)} \tag{2}$$

where $z^{'}$ represents the predicted neural network after Softmax processing, $z$ is the tag information, and $m$ is the number of data.

### B. *Estimation of 3D bounding box and target coordinates*

The estimation of the 3D bounding box and the target coordinates contains two parts. The first part outputs a total 9 key point coordinates, including the coordinates of the 8 vertices surrounding the 3D bounding box and the coordinates of the center point of the target object. Based on the first part of the output, the second part extracts more precise target coordinates and horizontal rotation angle. It is a regression problem to predict the three-dimensional coordinates of the key points of the target object and the horizontal rotation angle. The mean square error function (MSE) is used as the loss function of the estimated network. The loss function is as follows:

$$loss(y^{'}, y) = 1/m \sum_{i=1}^{m} (y_i^{'} - y_i)^2 \tag{3}$$

where $y^{'}$ is the value predicted by the neural network, $y$ is the value of the data tag, and $m$ is the number of data. And the mean square error function is convenient for the gradient to descend.

In the positioning framework, for the classification part, it predicts the probability that the input data belongs to each category, and outputs the maximum value of the prediction probability. The larger the value is, the better the classification effect is. For the 3D bounding box and the target coordinate estimation model, it predicts the position coordinate and the horizontal rotation angle task of the target object. The smaller the error is, the better the prediction effect. Because the parameters of these two parts are shared during the first phase of training, when the parameters are updated during the training process, both two parts will affect the parameter update of the shared part of the network. According to the importance of the task, the accuracy of the predicted object position and the horizontal rotation angle should be higher. The 3D bounding box and the target coordinate estimation part has a greater influence on the parameter update of the shared part. When updating the parameters, to increase the weight of this part to improve the prediction accuracy. We design a new loss function with different weight coefficients. The weight loss function of the classification part is multiplied by 0.7, and the loss function is given as follows:

$$loss(y^{'}, y, z^{'}, z) = loss(y^{'}, y) + 0.7 * loss(z^{'}, z) \tag{4}$$

### C. *Training hyperparameter settings*

The network training has two stages. The first stage is to train the target classification, the 3D bounding box and the first part of the target coordinate estimation. The train process contains a total of 60 epochs. The learning rate of the first 12

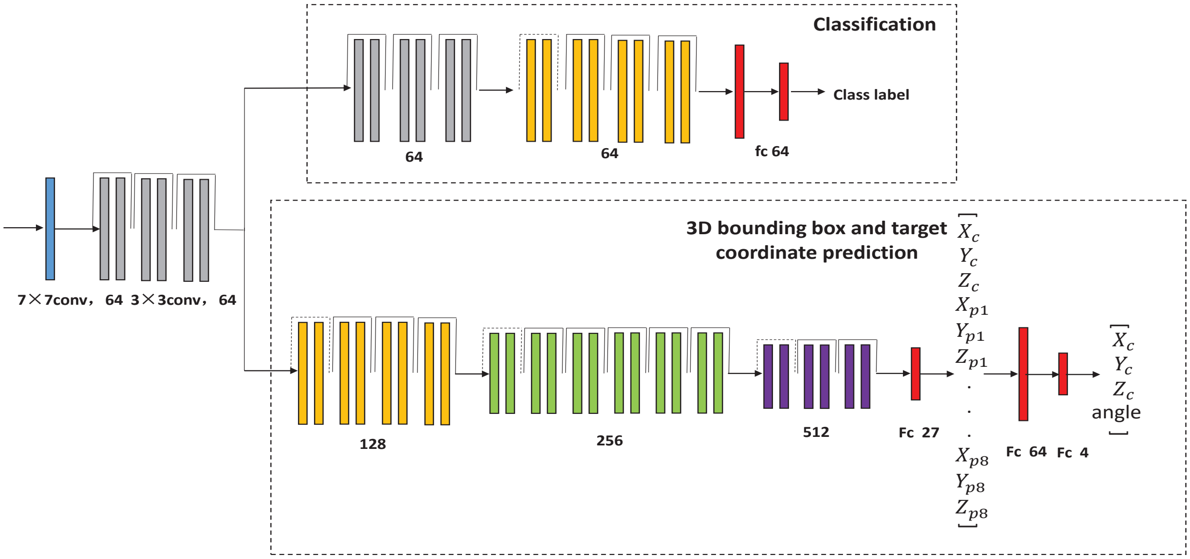

Fig. 2. A schematic diagram of a target three-dimensional positioning network architecture. The classification module mainly consists of three phases. The first stage contains three residual blocks, each of which contains two 33 convolution operations. The solid polylines across the two convolutional layers in the Fig. 2 represent the short-cut connections of the residual blocks, and the batch normalization is after convolution operation. The second stage contains 4 residual blocks. The dashed line represents the shortcut connection of the residual block, but it contains a 11 convolution operation to change the channel number of the residual block input feature map and the channel of the output feature map consistently. A 128-dimensional feature vector is generated by an average pooling layer as an input to the third stage. The third phase is two fully connected layers that implement the classification task. The 3D bounding box and the target coordinate estimation are divided into four phases. The network structure of the first two phases is the same as the network structure of the first two phases of the target classification module. The third stage consists of 6 residual blocks, and the output is obtained from the feature map. The fourth stage contains 3 residual blocks. After an average pooling layer, a 512 eigenvector is obtained, which is used as the input of the fully connected layer to predict the coordinates of the nine key points, followed by two fully connected layers for coordinate fitting, resulting in more accurate position coordinates and Horizontal rotation angle.

epochs is 1e-4, and the learning rate of the 12∼24th epochs is 1e-5, the subsequent learning rate is 5e-6. The number of images per batch is 64.

The second stage is to train the 3D bounding box and the second part of the target coordinate estimation. The training of this stage takes the prediction results of the first part of the module as input data, that is, the predicted 8 vertex coordinates of the 3D bounding box surrounding the target object and the inaccuracy center coordinates of the target object. This train process contains a total of 16 epochs with a learning rate of 1e-4 and 64 data per batch. The Adam is used as the two-stage optimization algorithm. The hyperparameters adopt the recommended parameters in the Adam optimizer [17]. The exponential decay rate of the first-order moment estimation is 0.9.

## IV. Experiment and result analysis

### A. Classification experiment

The target classification model is tested and its performance is evaluated by accuracy and recall rate. Fig. 3 shows the classification accuracy and recall rate during training. We can see that after 50 epochs, the curves of accuracy and recall rate become stable, and both of them are above 90%.

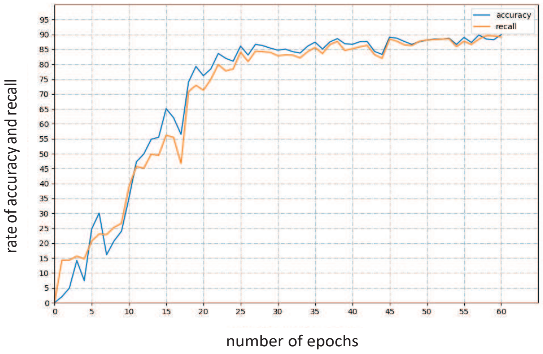

Fig. 3. Curve of accuracy and recall rate during training

The simulated images and real images are used to analyze the accuracy and recall rate of the classification with interference and in the absence of interference. The interferences in the real world are some common tools that never appear during the training phase. There are still some simulation-reality deviations when classifying real images. The accuracy and recall rate are lower than that of the simulated images, and the classification accuracy of the sphere is the worst. The experiment found that the sphere is always classified as a rectangular parallelepiped, even if there is no target object on the desktop. Since the radius of sphere is only 3 cm, the sphere is too small to be detected effectively in the input images with low image resolution. In the erroneous classifications, the objects are often mistaken into three categories: cubes, spheres, and cuboids.

### *B. Target coordinate and horizontal rotation angle prediction experiment*

The prediction module predicts the 3D bounding box surrounding the target object and extracts the coordinates and horizontal rotation angle of the target object according to the information of the 3D bounding box. To a certain extent, the prediction accuracy of the 3D bounding box determines that of the object coordinates and horizontal rotation angle. Fig. 4 shows the rendered dataset, which visualizes the 3D bounding box surrounding the six target objects. The red bounding box represents the actual bounding box and the green bounding box represents the bounding box of the neural network prediction.

This experiment is carried out on the simulated and real datasets respectively with six types of target objects, and their vertex positions are estimated. The diameter of the circular surface or sphere objects involved in this paper is 5 cm, and the length, width and height of other objects are also about 5 cm. The position and pose error estimates are obtained without interference respectively. The errors of $x$, $y$, $z$ coordinates of the six types of target objects and the horizontal rotation angle error are recorded during experiment. For the rotationally symmetric objects such as sphere, cylinder and cone, the rotation angle errors is set to zero.

Table I shows the estimated errors of the $x$,$y$,$z$ coordinates and horizontal rotation angle with the simulation image data and the real image data without interference. It can be seen from the Table I that for the simulated image data set, the predicted average error on the $x$, $y$, and $z$ axes is less than 0.5 cm. While the estimated $x$, $y$, and $z$ axis errors increase when estimated on the real image data, but they are still limited to 2.5 cm. Therefore, the results satisfied the positioning accuracy requirements when the robot arm operates the target object, and proved the effectiveness of our domain randomization method and the proposed three-dimensional positioning network framework.

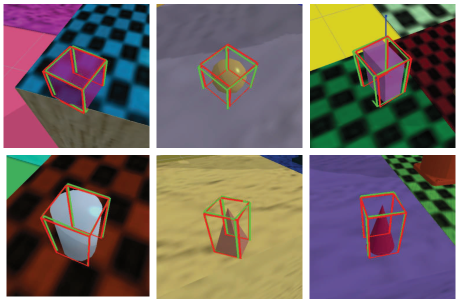

Fig. 4. Boundary box and schematic envelop bounding box

TABLE I
PREDICTION ERRORS OF THE OBJECT COORDINATES AND HORIZONTAL ROTATION ANGLE WITHOUT INTERFERENCE

| target object | Simulation data error | | | |
|---|---|---|---|---|
| | $x$/cm | $y$/cm | $z$/cm | *angle* /° |
| cube | 0.49 | 0.33 | 0.38 | 6.16 |
| sphere | 0.38 | 0.38 | 0.21 | - |
| cuboid | 0.41 | 0.50 | 0.31 | 7.06 |
| cylinder | 0.32 | 0.24 | 0.11 | - |
| pyramid | 0.37 | 0.55 | 0.11 | 6.40 |
| cone | 0.30 | 0.25 | 0.11 | - |
| average value | 0.38 | 0.38 | 0.21 | 6.54 |
| target object | Real data error | | | |
| | $x$/cm | $y$/cm | $z$/cm | *angle* /° |
| cube | 1.7 | 1.9 | 0.8 | 5 |
| sphere | 2.5 | 1.4 | 1.0 | - |
| cuboid | 1.2 | 1.6 | 1.1 | 6 |
| cylinder | 2.1 | 1.3 | 0.9 | - |
| pyramid | 2.4 | 1.7 | 0.9 | 8 |
| cone | 2.5 | 1.3 | 1.2 | - |
| average value | 2.1 | 1.5 | 1.0 | 6.3 |

There are two kinds of interferences: randomly generated block interference and the interference generated by the robot arm entering the camera field of view. The prediction of the errors of $x$, $y$, $z$ coordinates, angle under the simulation image data and the real image data with two kinds of interferences are obtained in Table II. It can be seen that the simulated image data with robot arm interference have a great influence on the prediction performance of the network, because the interference of the robot arm has never appeared in the training data set. In the case of interference blocks, the difference of the prediction effect between simulation image data with interference and without interference is small. There is an increase in the prediction errors on the real image data, but the overall error does not exceed 3.5 cm. This proves that the method can be transplanted to the actual experimental platform with little interference from the physical robot arm.

### *C. Physical robot arm experiment*

The experiment takes the result of real-time prediction of the target three-dimensional positioning framework as the input to the robot arm system which is trained by the reinforcement learning to operate the object. Fig. 5 shows the process of manipulating the robot arm on the target object. It proves that the target three-dimensional positioning method can be applied to the problem of the actual robot arm operating on the target objects.

TABLE II

PREDICTION ERRORS OF COORDINATES AND HORIZONTAL ROTATION ANGLE WITH BLOCK INTERFERENCE AND ROBOT ARM INTERFERENCE

| target object | Simulation data error | | | |
|---|---|---|---|---|
| | $x$/cm | $y$/cm | $z$/cm | $angle$ /° |
| cube | 0.56/1.46 | 0.37/0.68 | 0.39/0.55 | 6.36/6.44 |
| sphere | 0.75/1.88 | 0.93/1.05 | 0.31/0.44 | -/- |
| cuboid | 0.55/1.95 | 0.52/1.09 | 0.35/0.46 | 7.02/7.20 |
| cylinder | 0.47/1.77 | 0.34/0.85 | 0.13/0.41 | -/- |
| pyramid | 0.45/3.32 | 0.68/2.13 | 0.13/0.46 | 6.57/6.45 |
| cone | 0.60/2.06 | 0.54/1.08 | 0.13/0.46 | -/- |
| average value | 0.56/2.07 | 0.56/1.15 | 0.24/0.46 | 6.65/6.70 |
| target object | Real data error | | | |
| | $x$/cm | $y$/cm | $z$/cm | $angle$ /° |
| cube | 2.0/1.6 | 1.2/1.8 | 1.1/0.9 | 4/6 |
| sphere | 2.0/2.5 | 2.1/2.7 | 1.1/0.9 | -/- |
| cuboid | 2.5/1.8 | 1.7/2.5 | 0.8/0.8 | 7/8 |
| cylinder | 1.8/2.2 | 2.0/2.5 | 0.9/1.2 | -/- |
| pyramid | 1.8/1.7 | 2.2/3.2 | 1.0/1.2 | 6/6 |
| cone | 2.1/1.8 | 1.4/2.2 | 1.0/1.0 | -/- |
| average value | 2.0/1.9 | 1.8/2.5 | 1.0/1.0 | 6/7 |

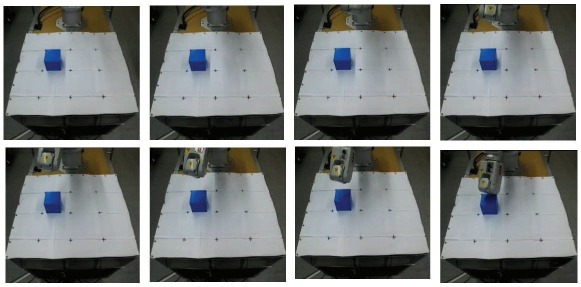

Fig. 5. Physical experiment platform to grasp test results

## V. CONCLUSION

This paper proposes a target three-dimensional positioning framework based on convolutional neural network. The framework uses the realistic RGB image data as input and predicts the classification, three-dimensional coordinates and horizontal rotation angle of the target object in the image. In the robotic arm operation, the framework helps the robotic arm acquire the class, position and pose information of the target. The image data used in the training in experiment is completely based on the image data set synthesized in the simulation environment, which greatly reduces the workload caused by collecting data. The domain randomization method reduces the deviation between the simulated image and the real image. The proposed network model completely uses the simulated image as the training data, which can be used to predict the real image and migrate to the real physical platform to achieve high success rate. In the next work, the experimental data can be further expanded, and the simulation images generated by domain randomization during training is to be combined with the real image data to further improve classification and prediction accuracy.

## REFERENCES

[1] Lowe D G , "Distinctive Image Features from Scale-Invariant Key-points," International Journal of Computer Vision, 2004, 60(2):91-110.

[2] Hinterstoisser S , Lepetit V , Ilic S , et al. "Technical Demonstration on Model Based Training,Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes," 2012.

[3] Rios-Cabrera R , Tuytelaars T . "Discriminatively Trained Templates for 3D Object Detection: A Real Time Scalable Approach," 2013 IEEE International Conference on Computer Vision (ICCV). IEEE Computer Society, 2013.

[4] Kouskouridas R , Tejani A , Doumanoglou A , et al. "Latent-Class Hough Forests for 6 DoF Object Pose Estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, PP(99):119-132.

[5] Drost B , Ulrich M , Navab N , et al. "Model globally, match locally: Efficient and robust 3D object recognition," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010.

[6] Lepetit V , Moreno-Noguer F , Fua P . "EPnP: An AccurateO(n) Solution to the PnP Problem," International Journal of Computer Vision, 2009, 81(2):155-166.

[7] Hinterstoisser S , Holzer S , Cagniart C , et al. "Multimodal Templates for Real-Time Detection of Texture-less Objects in Heavily Cluttered Scenes," IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011.

[8] Brachmann E, Krull A , Michel F , et al. "Learning 6D Object Pose Estimation Using 3D Object Coordinates," Computer Vision -ECCV 2014.

[9] Movshovitz-Attias, Yair , T. Kanade , and Y. Sheikh . "How useful is photo-realistic rendering for visual learning?," European Conference on Computer Vision Springer International Publishing, 2016.

[10] Sadeghi F , Levine S . "CAD2RL: Real Single-Image Flight without a Single Real Image," 2016.

[11] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P.Abbeel. "Domain randomization for transferring deep neural networks from simulation to the real world," In Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on, pages 23-30.

[12] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. "Sim-to-real transfer of robotic control with dynamics randomization," arXiv preprint arXiv:1710.06537, 2017.

[13] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel. "Asymmetric actor critic for image-based robot learning," arXiv preprint arXiv:1710.06542, 2017.

[14] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Bircheld. "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," arXiv preprint arXiv:1804.06516, 2018.

[15] M. Sundermeyer, Z. Marton, M. Durner, and R. Triebel. "Implicit 3d orientation learning for 6d object detection from rgb images," In Proceedings of the European Conference on Computer Vision (ECCV), pages 699-715, 2018.

[16] Tekin B, Sinha S N, Fua P. "Real-Time Seamless Single Shot 6D Object Pose Prediction," 2017.

[17] Kingma D P , Ba J . "Adam: A Method for Stochastic Optimization," Computer Science, 2014.