COMPSCI 696-01 Independent Study Final Report Object Detection and 3D Pose Estimation For Robot Manipulation

Xiaoqiang Yan
CICS, Umass Amherst
xyan@cs.umass.edu

Abstract

Object detection and 3D pose estimation system is important for robot manipulations. In order to adapt to variety of objects and complexity of clutter scenes, we introduce a Convolution Neural Network to detect objects and regress their 3D poses simultaneously utilizing RGB images. This network, which is based on the Faster R-CNN, predicts object class and bounding box in each RoI generated by RPN, and outputs object 3D translation relying on the predicted bounding box center and regressed center depth. The 3D rotation is estimated via regressing using the quaternion representation. Experimental results on the YCB-Video dataset indicate that our method is highly robust to illumination change and occlusion. We also apply our model to a new camera to investigate the system's behavior. The results indicate our approach generalize well to a different camera and different environment.

1 Introduction

Object detection and localization have been widely applied in industrial applications, such as automated product line, automated warehouse delivery, and robot manipulations. Especially in the robotics field, an object detection and 3D pose estimation system enables robots to execute pick or place task more precisely. Hence, object detection and 3D pose estimation have received much research attention in last few years.

Traditional methods for object detection and 3D pose estimation are based on template matching [1,2]. These methods address the issue of object detection and 3D pose estimation through comparisons between images and 3D templates with varied viewpoints, which are generated from object 3D models offline. Preparing object 3D models, however, especially for objects with complex structure, is inconvenient and time-consuming. These methods are also sensitive to illumination and occlusion.

In recent years, researchers have tried to build object detection and 3D pose estimation systems based on deep neural networks [3, 4], which have been successfully applied in the computer vision tasks [5, 6]. The research studies have demonstrated deep networks own powerful representation and generalize well. Existing 3D pose estimation algorithms are usually not end-to-end trainable and typically involve refinements. Thus, the overall object detection and 3D pose estimation systems are complicated and the running speed is not real-time due to the refinements.

In this work, we investigate an object detection and 3D pose estimation system based on Convolutional Neural Network (CNN). In our network, we predict object class and bounding box, and meanwhile regress the 3D translation and 3D rotation. The architecture

of our system is shown in Figure 1. The problem of object 3D pose estimation is decoupled into 3D translation and 3D rotation predictions. Our network consists of three major components. First, it generates thousands of region proposals (that is candidate bounding boxes) using Region Proposal network (RPN) [7]. The most promising set of proposals are selected and further refined as final object detection. This part is identical to the Faster R-CNN [7]. Second, it regresses the 2D center's distance from camera principle point for each RoI (region of interest), which represented by object detection bounding box. The bounding box center is used to represent an object's 2D center. We then get the 3D translation leveraging the calibrated camera intrinsic matrix. Finally, the 3D rotation is regressed directly through the network.

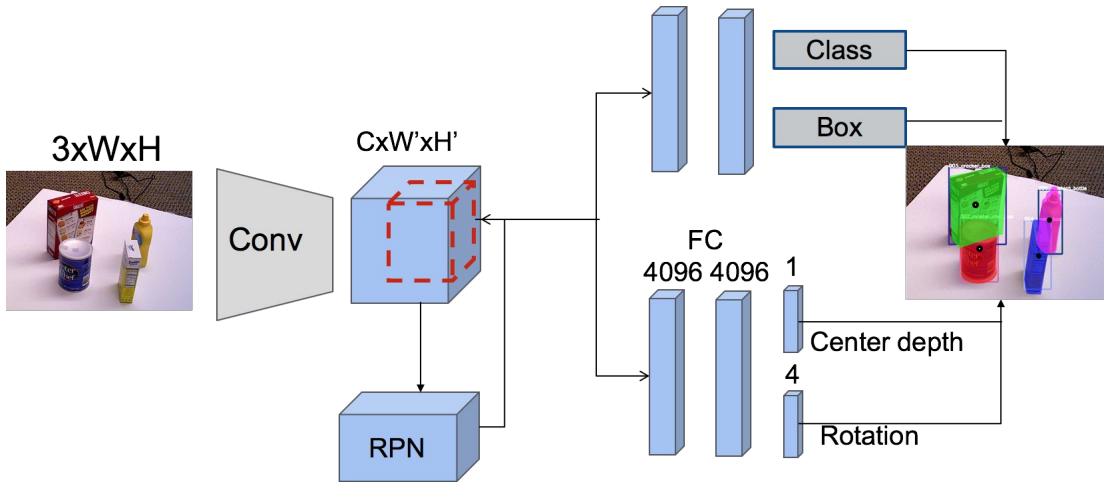

Figure 1: Architecture of our object detection and 3D pose estimation network.

In order to assess our system, we conduct experiments on the YCB-Video dataset introduced in [8]. This dataset includes RGB-D images, annotations of object bounding boxes, object 2D centers, class labels, 3D poses, and object 3D models. Hence, we can evaluate our system's performance by computing the average distance metric between the 3D pose ground truth and estimated results.

The reminder of this paper is organized as follows. After discussing related work in Section 2, we introduce our neural network and technical details in Section 3 and present our experimental results in Section 4. Finally, future work and conclusion for this project are discussed in Section 5.

2 Related Work

Research studies on object detection and 3D pose estimation systems have been conducted in two major directions.

Template matching methods. Most traditional approaches for object detection and pose estimation are matching input objects to the templates, which are generated from 3D model in varying viewpoints and scales [1, 2, 9–11]. In [1, 10], the authors leverage some algorithms to transform the images and measure the similarities between the templates and objects. Cao et al. [10] transform the images to the Laplacian of Gaussian space and

introduce a approach to reshape the template and images to perform real-time matching. He et al. [1] combine several point cloud processing algorithms to infer more accurate object 3D pose. In [9], the authors employ the Iterative Closest Point (ICP) algorithm to match the segmentation of objects and their pre-scanned 3D models to estimate the objects' poses. In [11], Latent-Class Hough Forests, a novel template-based function, is proposed to implement object detection and 3D pose estimation from background clutter and foreground occlusions. This method undertakes one-class learning in the training process, and infers latent class distribution in the test processing iteratively, and thus, gets a better detection rate. Although these methods can improve the performance of object detection and 3D pose estimation system, template-matching methods still need to prepare 3D models and numerous templates, which may restrict its applications in industry. Furthermore, the template-matching methods are sensitive to illumination and occlusion.

CNN-based methods With more applications of Convolutional Neural Network in object detection [6,7,12,13] and object segmentation [14,15], more and more scholars have started to leverage CNN to extract object pose [3, 4, 16, 17]. For object detection, a series of R-CNN [6, 7, 12] have achieved remarkable performance. The developers feed the images and object region proposals into CNN, where their final features are concatenated and a SVM classifier is used to acquire their object class labels and refine their bounding box coordinates. In [13], the depth modality is encoded from one channel to three channels which can extract more features from the depth images. Then the authors applied the CNN in [6] in both RGB and depth modalities to increase object detection accuracy.

The CNNs in [4,17] are applied to detect objects and estimate orientation of the objects. Schwarz et al. [4] initial the networks' weights using a pre-trained model, so that the training process becomes more concise and the model of CNN is more accurate. Braun et al. [17] build a deep CNN which has a similar architecture as [6], and they focus on the improvements caused by different 3D proposals. In [3], the authors feed local RGB-D patches and the feature votes into a CNN to regress the features which are stored in a codebook; thus, the CNN implements object detection and pose estimation. In a recent technical report [16], a new CNN architecture inspired by the YOLO object detection network [18] is proposed to predict the 2D image locations. The 2D image location is the projection of the object's 3D bounding box, and then estimate the object 3D pose using a PnP algorithm.

In this paper, we regress object 3D pose based on Faster R-CNN and RPN as two additional parallel branches. The Neural Network jointly detect object and regress 3D pose. Compared with recent works, our method is more concise than [8]. We only generate 2D proposals for objects, which saves more time than 3D proposals [17].

3 Technical Approach

In this work, we leverage the architecture of Faster R-CNN [7] to implement object detection. Meanwhile, we propose two additional branches to regress 3D pose in each Region of Interest (RoI), that is object detection bounding box. For the 3D pose regression, 3D translation $\mathbf{T}$ and 3D rotation $\mathbf{R}$ will be achieved separately. Here, $\mathbf{T} = (T_x, T_y, T_z)$ stands for the coordinates of object center in the camera coordinate system. $\mathbf{R} = (\alpha, \beta, \gamma)$ means the object rotation angles around X-axis, Y-axis and Z-axis, respectively.

3.1 Network Overview

Our network's architecture, which is based on the Faster R-CNN [7], is illustrated in Figure 1. Faster R-CNN is the state-of-the-art network. It introduces a RPN (region proposal network) to generate region proposals, and combines RPN with the Fast R-CNN [12] into a single network. This makes the network end-to-end trainable. We refer readers to [7] for more details. We use this architecture to implement object detection and 3D pose estimation from RGB images.

3.2 3D Translation Estimation

In the 3D translation regression branch, instead of regressing (T_x, T_y, T_z) directly, $\mathbf{T}$ will be estimated as the 2D object center in the image (c_x, c_y) and object distance from the camera principle point T_z. The object center is represented by bounding box center. Thus, T_x and T_y are computed according to projection equation as follows:

$$\begin{bmatrix} T_x \\ T_y \end{bmatrix} = \begin{bmatrix} \frac{c_x - p_x}{f_x} T_z \\ \frac{c_y - p_y}{f_y} T_z \end{bmatrix}. \tag{1}$$

Here (p_x, p_y) is the principle point of the camera and (f_x, f_y) stand for the focal lengths, which can be obtained via calibration of a certain camera.

3.3 3D Rotation Regression

The final branch of our network is to regress object rotation angles directly. In order to compute the angular distance between two 3D rotations more efficiently, we transfer the rotation matrix into a quaternion. Hence, we need to recover a vector with four numbers for each RoI.

3.4 Multi-task Loss Function

In order to train the network, we define a multi-task loss to jointly train the classification, bounding box regression, the translation regression, and rotation regression.

$$L = \alpha L_{cls} + \beta L_{bbox} + \gamma L_{trans} + \eta L_{rot}.$$

The classification loss L_{cls} is the softmax loss. The other three regression loss terms L_{bbox}, L_{trans}, L_{rot} are the smooth L_1 loss [7]. $\alpha, \beta, \gamma, \eta$ are scale invariant, which need to be specified by concrete experiments. In our experiments, we set $\alpha = \beta = 1, \gamma = 2, \eta = 1$

3.5 Evaluation Methods

For 3D pose estimation, we present translational error (E_{te}) and rotational Error(E_{re}) [19].

$$E_{te} = ||\hat{T} - \bar{T}||_2,$$
$$E_{re} = arccos((Tr(\hat{R}\bar{R}^{-1}) - 1)/2),$$

where $\hat{T}$ and $\hat{R}$ denote 3D pose prediction, respectively. $\bar{R}$ and $\bar{T}$ are 3D pose ground truth.

4 Experiments

We conduct experiments on the YCB-Video dataset [8]. What we focus on is a subset of 15 YCB objects, which are shown in Figure 2. We split the dataset randomly, where 6000 images are used for training and 1611 for testing. We present translation and rotation error [19] and leverage $5cm5°$ metric to evaluate the 3D pose error. In $5cm5°$ metric, if $E_{te} < 5cm, E_{re} < 5°$, the estimated pose is accepted to be accurate.

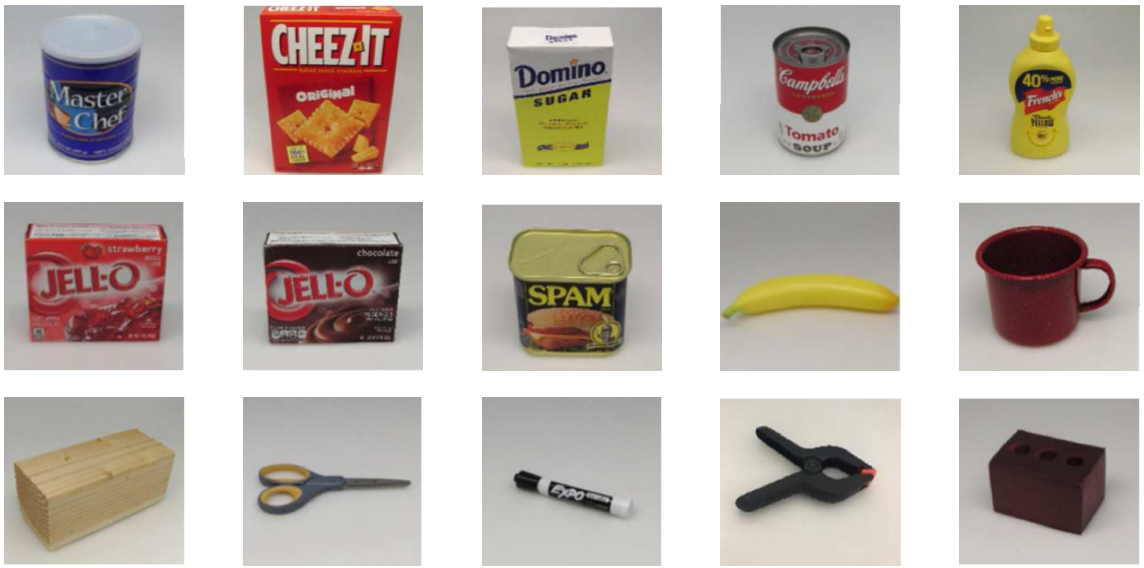

Figure 2: The objects in the YCB-Video Dataset

All the detection and estimation models, initialized with the ImageNet pre-trained VGG16 [20], are trained for 50K iterations using a learning rate of 0.001 with the momentum SGD optimizer, which is reduced by 0.1 every 20K iterations. It takes about 6 hours on a NVIDIA TitanX GPU to train the model.

4.1 Experiments on the YCB-Video Dataset

In order to get a good model, we regress the translation leveraging two methods to compare the accuracy. One is to regress the depth information and compute the translation (T_x, T_y) according to Eq.(1); the other is to regress the translation (T_x, T_y, T_z) directly. The images in our dataset include multiple objects, annotated ground truth object classes, bounding boxes, and 3d poses. In the training process, we regress translation and rotation for each proposal, and choose the Smooth L_1 loss to train the model. In the testing, in order to avoid multiple results for an object, we use non-maximum suppression (NMS) to remove duplicates. The effectiveness of Faster R-CNN [7] has been proved in lots of fields [13, 21]. Hence, we only focus on the pose estimation results in our experiments.

Table 1 presents the 3D pose estimation errors and accuracy under the $5cm5°$ evaluation metric with two translation regression methods. As shown in table 1, the first translation achievement method (center+T_z) gives significantly better results (mean accuracy 75.3% in column A versus 48.0% in column D). We can see that smaller translation error (mean Ere $3.8cm$ in column B versus $5.6cm$ in column E) has contributed to the higher pose estimation accuracy. Also, "scissor" has the worst pose estimation results in the dataset. There is a strong possibility that the appearance of the scissor may influence the accuracy

	A	B	C	D	E	F
	$(Center + Tz)$ Regression			(T_x, T_y, T_z) Regression		
Category	Accuracy (%)	E_{te} (cm)	E_{re} (°)	Accuracy (%)	E_{te} (cm)	E_{re} (°)
chef can	**93.0**	2.4	1.6	62.6	4.7	1.6
crack box	71.5	3.8	1.9	37.4	6.1	1.8
sugar box	78.0	3.3	1.6	54.7	5.3	1.6
tomato soup can	84.2	3.0	1.7	64.8	4.5	1.8
mustard bottle	70.3	3.9	1.9	39.2	6.4	1.9
pudding box	67.2	4.3	1.3	30.1	6.9	1.4
gelatin box	**89.3**	2.3	1.6	59.6	4.7	1.8
meat can	63.6	4.1	1.2	3.67	6.5	1.3
banana	67.0	4.9	1.2	47.6	6.5	1.4
mug	**89.8**	2.6	1.6	72.3	4.1	1.7
wood block	69.7	4.1	1.6	38.5	6.4	1.7
scissor	51.3	5.7	1.7	36.1	6.6	1.7
large marker	56.2	4.9	1.2	43.4	7.0	1.2
extra large lamp	85.2	4.8	1.7	64.9	4.5	1.8
foam brick	**93.8**	2.3	1.0	64.6	4.4	1.1
mean	**75.3**	**3.8**	1.5	48.0	**5.6**	1.6

Table 1: 3D pose evaluation and error on the YCB-Video dataset.

of translation regression. In some viewpoints, the center of the bounding box may not align with the object center well.

Figure 3 displays the object detection and 3D pose estimation results on the YCB-Video dataset, where the 3D mesh are overlaid on the corresponding objects according to estimated 3D pose. We see from the images that our model can detect and estimate 3D pose accurately even if the object is partially visible in the image. Last raw in figure 3 shows some failure cases in pose estimation. According to these images, we find our network may not estimate 3D pose well if the contour and texture of an object are not obvious, such as the wood block in Column C in the last raw.

When we conduct testing on the YCB-Video dataset, the running time is 0.19s per image for 300 object proposals. So our approach can be applied on robot real-time manipulation.

4.2 Experiments on new Camera

Now we achieve a good object detection and 3D pose estimation network model, can the model generalize well to a new camera? How about the performance in different environments?

We apply the model trained on the YCB-Video dataset to a new Asus Xtion Pro Live RGB-D camera, which provides RGB images with 1280 × 1024 resolution. We build a real-time object detection and 3D pose estimation system based on ROS. In order to leverage our network model, we crop 640×480 resolution images from center region of the original image.

Because we didn't create any ground truth information for the new images, we present

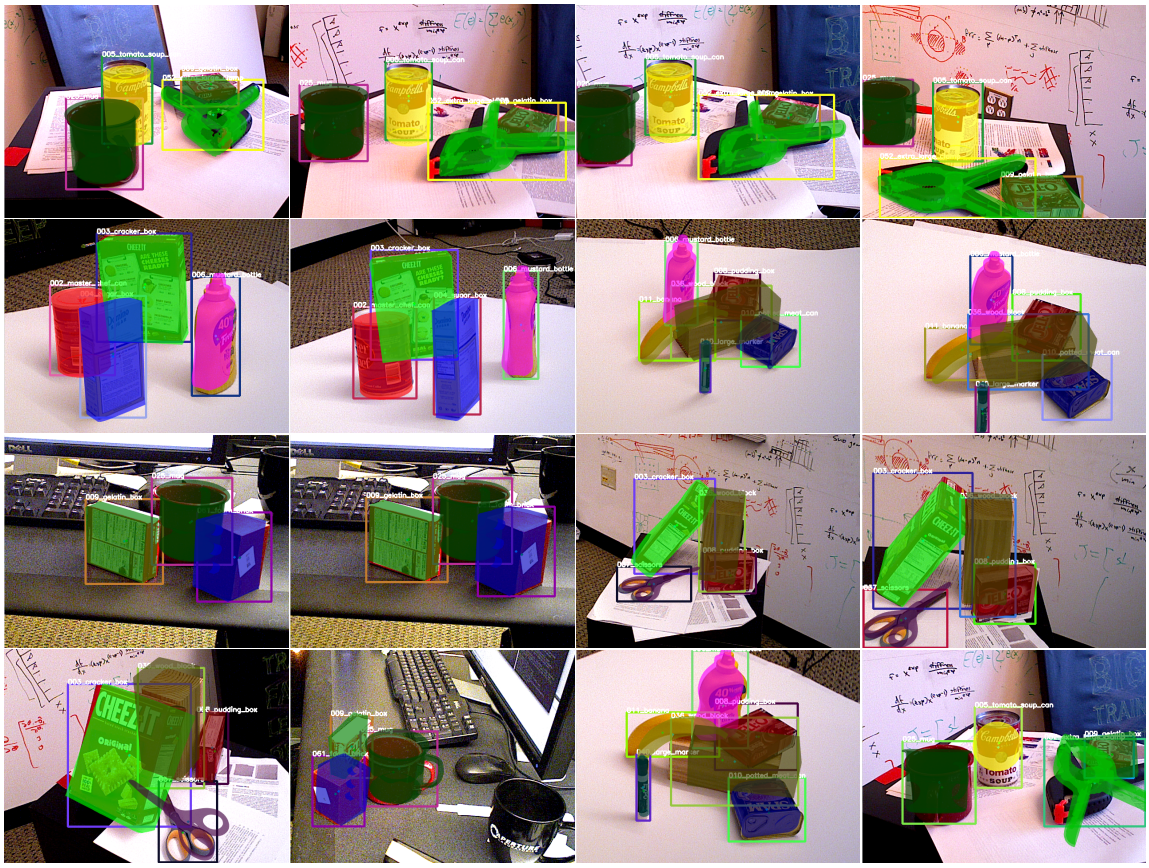

Figure 3: Object detection and 3D pose estimation results on the YCB-Video dataset. Last rows are failure cases.

the object detection results using object label and bounding box, and rely on the alignment of the overlaid 3D mesh model and RGB image to evaluate the pose estimation performance. Figure 4 shows the model detection and pose estimation results in the new camera. We can see from these images that our model can detect objects and estimate object 3D pose in different viewpoints although sometimes it works not very well. The estimated object translation is accurate because the position of 3D mesh matches the object position in the image. However, the estimated rotation is not as accurate as the testing results on the YCB-Video dataset. According to the analysis to the results, we find it is possible that the quaternion regression might be more sensitive to the environments, such as illumination and color information. And because there are four values in a quaternion vector, even if each corresponding value has little difference, the quaternions may represent diverse rotations. Also, in the new camera, due to the GPU limitation, we decrease the image size from 800 to 700, which may have influenced the results. Furthermore, in the experiments we found that for the objects with less textured information, such as "foam brick" and "big marker", our model may not detect them well in the new camera.

In order to evaluate the uncertainty of our real-time object detection and 3D pose estimation system, we collect the system outputs for the object in fixed position. In these images, the background for the objects changes and the illumination conditions are not exactly the same. Table 2 shows 3D pose estimation uncertainty analysis. We find the outputs have little fluctuation, indicating our system is stable for same object in fixed position and similar environments.

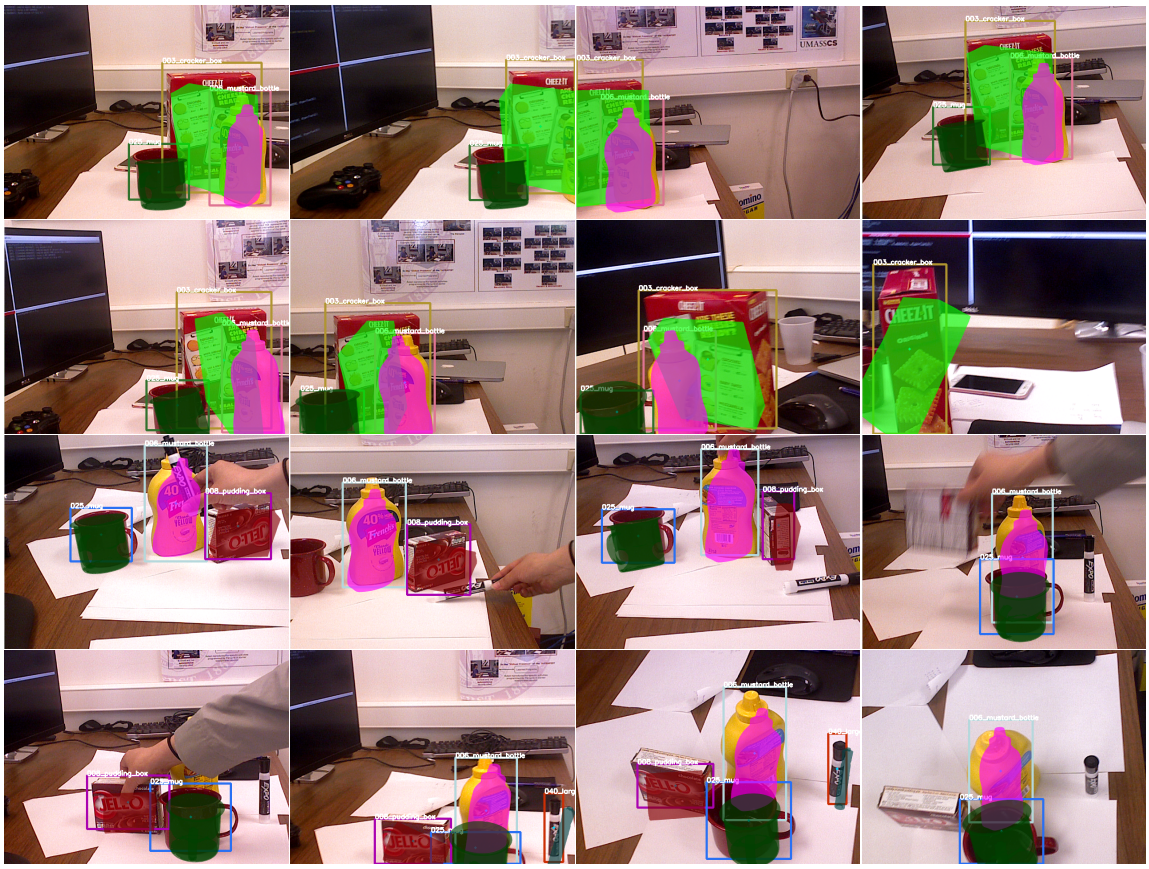

Figure 4: Object detection and 3D pose estimation results under a new camera.

A snapshot of our system can be found in Figure 5. In addition to overlaid 3D mesh models on RGB images, the point cloud is also projected to the image. Therefore, the accuracy of estimated pose can be checked visually.

5 Future Work and Conclusion

According to [13], combination of RGB and depth information gives better results on object detection. Hence, we plan to explore RGB-D object 3D pose estimation in the following work. Also, it is also worthy to further investigating different regression methods to 3D pose rotation. Comparison on different approaches might give us more indications on how to get higher accuracy as transferring models to a new system.

To conclude, in this work, we propose an deep CNN approach to detect object and recover object 3D pose jointly. The network estimates 3D translation by predicting 2D center and center depth information. 2D center is the center of object bounding box. 3D rotation is regressed directly as a quaternion vector. The results on the YCB-Video dataset indicates the approach is effective and stable. Also, we applies the model to a new camera. Although the estimation for 3D pose is not as accurate as in the YCB-Video dataset, the application clarifies the feasibility of transferring our model to new environments. Furthermore, the results encourage us to undertake more investigations on this research field.

	Translation			Rotation (quaternion)			
	T_x	T_y	T_z	a	b	c	d
mean	7.25e-02	-4.55	77.1	1.14e-01	-5.01e-02	-4.43e-01	3.16e-01
covariance	2.26e-06	-1.75e-06	1.98e-05	-2.65e-05	3.05e-06	-7.37e-06	3.32e-05
	-1.75e-06	1.25e-05	-3.53e-05	6.39e-05	7.96e-05	3.37e-05	2.63e-05
	1.98e-05	-3.53e-05	2.52e-04	-3.25e-04	-7.69e-05	-1.645e-04	2.79e-04
	-2.65e-05	6.39e-05	-3.25e-04	5.06e-04	2.88e-04	2.43e-04	-2.11e-04
	3.05e-06	7.96e-05	-7.69e-05	2.88e-04	7.56e-04	1.62e-04	5.14e-04
	-7.37e-06	3.37e-05	-1.63e-04	2.43e-04	1.62e-04	4.20e-04	-1.94e-04
	3.32e-05	2.63e-05	2.79e-04	-2.11e-04	5.14e-04	-1.94e-04	1.06e-03

Table 2: Uncertainty analysis of our system.

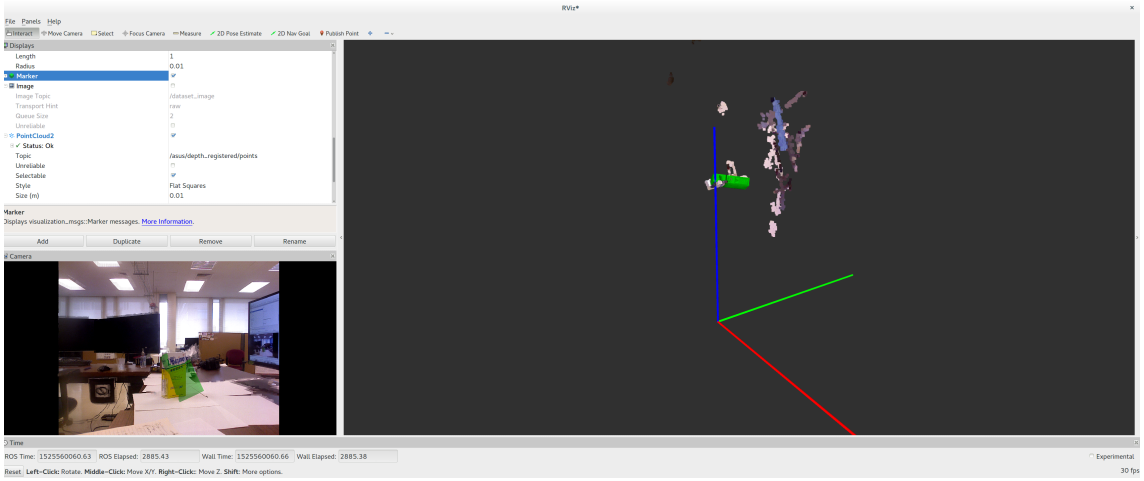

Figure 5: The real-time object detection and 3D pose estimation based on ROS

References

[1] Ruotao He, Juan Rojas, and Yisheng Guan. A 3d object detection and pose estimation pipeline using RGB-D images. *CoRR*, abs/1703.03940, 2017.

[2] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter F. Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5):876–888, 2012.

[3] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep learning of local RGB-D patches for 3d object detection and 6d pose estimation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, pages 205–220, 2016.

[4] Max Schwarz, Hannes Schulz, and Sven Behnke. RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, pages 1329–1335, 2015.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

[6] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[7] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[8] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *Robotics: Science and Systems (RSS)*, 2018.

[9] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker Jr., Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 1386–1383, 2017.

[10] Zhe Cao, Yaser Sheikh, and Natasha Kholgade Banerjee. Real-time scalable 6dof pose estimation for textureless objects. In *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pages 2441–2448, 2016.

[11] Alykhan Tejani, Rigas Kouskouridas, Andreas Doumanoglou, Danhang Tang, and Tae-Kyun Kim. Latent-class hough forests for 6 dof object pose estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(1):119–132, 2018.

[12] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[13] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014*, pages 345–360. Springer, 2014.

[14] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[16] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. *CoRR*, abs/1711.08848, 2017.

[17] Markus Braun, Qing Rao, Yikang Wang, and Fabian Flohr. Pose-rcnn: Joint object detection and pose estimation using 3d object proposals. In *19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016, Rio de Janeiro, Brazil, November 1-4, 2016*, pages 1546–1551, 2016.

[18] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6517–6525, 2017.

[19] Tomas Hodan, Jiri Matas, and Stepán Obdrzálek. On evaluation of 6d object pose estimation. In *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III*, pages 606–619, 2016.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[21] Huaizu Jiang and Erik G. Learned-Miller. Face detection with the faster R-CNN. In *12th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2017, Washington, DC, USA, May 30 - June 3, 2017*, pages 650–657, 2017.