Object Detection using RGB camera and Force/Torque sensor fusion in Peg-in-Hole task

Othman Mostafa Osama Metwally
Robotics and Computer Vision
Innopolis University
Innopolis, Russia
M.Othman@Innopolis.University

Shaarawy Abdelaziz Wasfy
Robotics and Computer Vision
Innopolis University
Innopolis, Russia
A.Shaarawy@Innopolis.University

***Abstract*—In this paper, We implement a full task for pick and place for peg-in-hole task where a single eye on hand RGB camera is used to detect cylindrical featureless object among other shapes and estimates it's 3D pose in world space. A Kuka iiwa 7 serial Robot is used for perfoming the task, to get more accurate results we introduced the force and torque readings in the Kuka arm joints. using sensor fusion of camera with force and torque sensors readings we could get more accurate results in faster way.**

***Index Terms*—Object detection,classification , YOLOv5 , Sensor Fusion , Pick-Place.**

All the codes of this project are uploaded on Github

I. Introduction

Assembly operations are crucial tasks and can be found in different fields of the industry that in most cases require high accuracy which is difficult to maintain in assembly operations with low clearances. Therefore, robots are looked at to perform these operations to achieve high accuracies. However, when it comes to very low clearances (0.1 mm), still robots can fail to achieve such precise fits [1].

That being the case, research is aimed to integrate different types of sensors with robots to develop better perception of these low clearances and compensate the errors. An example of assembly using Force/Torque sensor and position sensing is discussed in So, the aim of this project is to combine a Camera and Force/Torque sensor (sensor fusion) with a serial robot manipulator to enhance the accuracy of assembly operations in general [2]. The role of the camera is to locate, detect the specimen, and provide live feedback of the position/pose of the assembly parts. As for the Force/Torque sensor, it will be used to give readings of the contact between the assembly parts [3].

II. Related Work

A. *Object Detection and Classification*

A typical autonomous Pick and Place task can be split into two main tasks: object detection and object handling. With the aid of a vision system, the targeted object can be identified in the working environment and localized, in another word the object is now said to be detected. Now the object is known to the working environment and can be interacted with according to the specified application task. In robotics applications, this methodology is commonly used with the requirement of speed and accuracy. Thus, Object Detection task has become a field of interest in hardware applications [4]. Various methods are now being developed and tested to aim for robust, speed detection using Computer Vision tools. Additionally, Machine Learning has been involved in such tasks to provide not only accurate results but also a wide range of classes' classification.

Object detection is widely used and implemented in many robotics and computer vision applications, we have seen various ways and techniques in order to detect objects. Also the classification problem is widely used in the mean time. there are two main approaches for solving such major problems, Either by using computer vision techniques and algorithms in order to manipulate the image and enhance the output of it. or by using machine learning algorithms, where some prepossessing are performed on the images then these images are fed into a neural network model (CNN ,R-CNN) where this model is then trained to detect specific objects from any images or classify objects in them [5].

These two main tracks have advantages and disadvantages, and we should choose accordingly for each task what will be more suitable and convenient for our application [6].

1) Computer Vision: Detecting Objects entails two processes; firstly localizing objects in the environment, then classifying them. There are plenty of computer vision detection techniques, each aims to extract certain features of the given environment in order to locate the object of interest. Some of these techniques are color-based, shape-based, and feature-based, some of these were implemented before [7].

Color-based The color of the object is the essential feature that the object is being detected accordingly. It is used to segment the object of a certain primary color from the environment enabling object detection and tracking [8]. This approach can be refined to be effective in recognizing multiple colored objects even if there is non-uniform illumination, or geometrical variations [9].

Shape-based In this approach the geometric features of the object are extracted. These features are used for classification based on feature matching from a given dataset [10]. In addition, Hough Transfrom is a common

2) Machine Learning: we have searched for various object detectors and classification algorithms and found lots of them, However accuracy, speed and light implementations are the most important factors that we will need in order to accomplish our task. We found that YOLO5 v4 is the most recent YOLO version [11] and the most accurate and fastest of its older versions [12], [13]. It is used for object classification and detection where it can recognize various objects by only looking once so its very fast compared with others.

We have used the Yolov5s version which is considered the smallest YOLO model and the fastest one in implementation with processing speed reaching 455 fps and 2.2ms speed. YOLOv5 has about 25 hyper-parameters used for various training settings, these hyper-parameters affect in the training speed and it is always better to tweak them before training according to the images you are feeding to the model, better initial guesses will get better results for your model. the model is trained initially on COCO dataset with over 40000 images of about 80 classes of different objects, but for our work we need a custom dataset of cubes and cylinders where we could train the model on only 3 classes red cube, blue cube and cylinder.

III. Implementation

A. Machine Learning

For Our task we need to implement the YOLO algorithim on the objects at the table in front of the arm so as we discussed we have multiple shapes and we need first to classify them and then detect the black texture less cylinder, after that we try to detect the cylinder position in the camera frame

We have captured about 60 images in different configuration for the objects and in occluded condition, after that we used a software called Roboflow for labeling every image. Where, a text file is generated with class number , center of x, center of y, width and height for each object in the image in a separate row and then these files are organized into two datasets one for training and the other for validation.

For our Model we

using Yolo5 v4 [11] our model data and implementation details can be found in this Report with full description.

Acknowledgment

We would like to thank Prof.Mohamed Fahim and Riby Abraham for there support and Karam el maghoot for implementation on Kuka iiwa.

References

[1] "A system for learning continuous human-robot interactions from human-human demonstrations." [Online]. Available: http://ieeexplore.ieee.org/document/7989334/

[2] J. Gamez Garcia, A. Robertsson, J. Gomez Ortega, and R. Johansson, "Sensor fusion of force and acceleration for robot force control," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3. IEEE, pp. 3009–3014. [Online]. Available: http://ieeexplore.ieee.org/document/1389867/

[3] S. Briot, M. Gautier, and A. Jubien, "In situ calibration of joint torque sensors of the KUKA LightWeight robot using only internal controller data," in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, pp. 470–475. [Online]. Available: http://ieeexplore.ieee.org/document/6878122/

[4] "A simple robotic eye-in-hand camera positioning and alignment control method based on parallelogram features," vol. 7. [Online]. Available: http://www.mdpi.com/2218-6581/7/2/31

[5] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," 06 2018.

[6] R. A. Boby and S. K. Saha, "Single image based camera calibration and pose estimation of the end-effector of a robot," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2435–2440. [Online]. Available: http://ieeexplore.ieee.org/document/7487395/

[7] R. A. Boby, "Hand-eye calibration using a single image and robotic picking up using images lacking in contrast," in *2020 International Conference Nonlinearity, Information and Robotics (NIR)*, Dec 2020, pp. 1–6.

[8] R. Verma, "An efficient color-based object detection and tracking in videos," *International Journal of Computer Engineering and Applications*, vol. 11, pp. 172–178, 11 2017.

[9] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," p. 12.

[10] S. Gupta and Y. Jayanta, "Object detection using shape features," 12 2014.

[11] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, A. Chaurasia, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, F. Ingham, Frederik, Guilhen, A. Colmagro, H. Ye, Jacobsolawetz, J. Poznanski, J. Fang, J. Kim, and K. Doan, "ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration," Jan. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4418161

[12] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.