Detection and Pose Estimated Grasping in Industrial Robot for Bin Picking Operations

Avinash Sen
M.Tech Student
Department of Production Engineering
Government Engineering College
Thrissur, India
avinashsen707@gmail.com

Aju S S
Asst. Proffessor
Department of Production Engineering
Government Engineering College
Thrissur, India
ajusainu@gectcr.ac.in

Dr. Lalu P P
Asst. Proffessor
Department of Mechanical Engineering
Government Engineering College
Thrissur, India
lalujesus@yahoo.co.in

Sudeesh R S
Asst. Proffessor
Department of Mechanical Engineering
Government Engineering College
Thrissur, India
rssudheesh@gectcr.ac.in

***Abstract*—The technique used by a robot to grab objects that are randomly placed inside a box or a pallet is called bin picking. Bin picking has evolved greatly over the years due to tremendous strides empowered by advanced computer vision technology, software development and gripping solutions. However, the creation of a versatile system, capable of collecting any type of object without deforming it, regardless of the disordered environment around it, remains a challenge. The solution for this problem can be achieved by learning the appearance of the object in the bin using convolutional neural network(CNN). In the dataset that is used for training has photorealistic images with precisely annotated 3D pose for all objects. Such a dataset is generated by synthetically overlaying object models and backgrounds with difficult composition and high graphical feature. We can find the object poses with adequate accuracy using this network for physical world semantic grasping in a disordered bin by actual robot.**

Keywords—Convolutional neural networks, pose estimation, computer vision, disordered bin, semantic grasping.

I. INTRODUCTION

As companies identified the increase in the demands on how labour-intensive tasks are performed, there was a growing need to automate as many manual processes as possible, thus ensuring improvements in quality and consistency while simultaneously decreasing production time. One of the fastest growing automation strands is certainly industrial robotics. According to the International Federation of Robotics, industrial robot sales rose 15% in 2015 and ABI Research estimates that sales in this industry will triple by 2025 [1]. The applications of the robots are quite diverse, however, high precision, productivity and flexibility in the tasks performed are guaranteed [2]. The evolution of the perception systems was crucial to promote the development of robotic systems that were more versatile and able to perform tasks of greater complexity. Machine Vision and Robot Vision, together with the Image Processing and Computer Vision techniques can be used to improve the quality of images and extract information from them, respectively, are the best global perception systems which used since no contact with the outside world is required. In an industrial environment, the objects and materials used are not always arranged in an organized and structured way, therefor there is a need to automatically manipulate objects in these situations.

Bin-picking is the name of the technique used by a robot to grab objects that are arranged randomly inside a box or on a pallet. This technique has been researched by numerous organizations for over 3 decades, and one of the first attempts to create such a robust system was developed in 1986 at MIT [3]. Initially only 2D images were used, coupled with distance sensors, to acquire data, but the continuous technological evolution allowed for this to be done with 3D devices. In contrast, the Random Bin-picking process is considered the ultimate goal for creating a vision-controlled robot, also known as VGR (Vision-guided robots) [3]. However, the creation of this type of versatile and precise system, capable of collecting any type of object without deforming it, regardless of the disordered environment around it, remains the main objective. Although several companies have already proposed different solutions to date, these are indicated to solve specific problems and are still not sufficiently versatile. In addition, most of the existing processes are used to grab non-fragile objects, due to the high degree of precision needed to avoid deforming sensitive objects.

II. RELATED WORKS

A. Computer vision and Convolutional neural networks

Recently, developments of deep learning and convolutional neural networks (CNN) made major breakthroughs in the classification and detection tasks in the field of computer vision [4]. Thus the industrial [5] and academicals [6] problems solving led to increased need of convolutional neural network solutions. Even though CNN is so much popular, they haven't yet been more used for bin picking task. Some of the areas have been described in [7].

In [8], Convolutional Neural Network is used to section an item in an RGBD image (a color image with an extra depth channel), which is tailed by shape corresponding of the segmented item and the acknowledged 3D model, which is conceded out with iterative nearby point [9] and 3DMatch [10]. One more application is obtainable in [11], where the job of 3D pose estimation is separated into two portions: first, a

descriptor of an image patch is generated; and second, the nearest neighbour search is used to determine the orientation and class of the object presented in the image. The descriptor is created in such a way that like image patches are defined by like vectors (whose total difference is a lesser value), and this lets generating a database of images of objects saw from different locations, and determine the orientation of an item in a test image as the nearby neighbour (by relating the descriptor vectors) in the database. The job of making a descriptor for an image patch (RBG or RGBD) is controlled by a CNN. A cutting-edge grasping method is defined in [7]. Here, learning of the grasping behaviour is controlled in an endwise method, without a halfway demonstration of the object pose. Learning the grasping object in this way is theoretically neat, but is impracticable for many researchers because of the rate of the equipment (numerous robotic manipulators are used for learning).

Convolutional Neural Networks have been also applying for pose calculation of a human body from ordinary RGB images. In [12], a setup for pose regression from raw image is hypothesised, where a deep neural network is explained with an RGB patch with a human, and it outputs the total picture coordinates of the adjoins (the joints values are outputted one by one). In order to rise the accuracy of the finding, a set of networks is standardised, in which the accuracy is increased with each stage. Another area has been prescribed in [13], where the joints are not found in total picture coordinates, but rather in three dimensional coordinates.

III. Proposed Approach

To make robots to do jobs more efficiently in difficult situations, researchers from NVidia have established a Deep learning method that consents robot to study and pick domestic object with ease. Using a camera mounted on the robot, the AI continuously detects and estimates the full pose of the objects. Knowing the location and orientation of items in the environment, then referred to 6DOF pose is acute. This made robot to move objects even if they are not in the similar location every instance. This research which made on previous work developed by NVidia Researches, allows robot to accurately understand the pose of objects around them while trained only on photorealistic data. The advantages of such data have over real world data, is that it is now potential to create infinite quantity of annotated training data for deep neural networks.

This is a state of art pose estimation and we decided to use this mainly for our research. So that we can focus on eliminating other problems not just about the concept of pose estimation and its mathematics [14]. They suggest a two-step result to report the problem of identifying and estimating the 6-DoF pose of all occurrences of a set of identified domestic objects from a solo RGB image first, a deep neural network finds belief maps of 2D keypoints of all the objects in the image coordinate system. Secondly, highest from these belief maps are forwarded to a standard perspective-n-point (PnP) algorithm [12] to find the 6-DoF pose of respective object occurrence. In this part we explains these steps, along with our novel steps of creating synthetic data for training the neural network.

A. Network architecture

Inspired by convolutional pose machines (CPMs) [15][16], their state of the art completely a convolutional deep neural network detects keypoints using a several stage architecture. The feed forward network accepts as input an RGB image of size w*h*3 and divides to make two other outputs, namely, belief maps and vector fields. There are nine belief maps, one for each of the represented 8 vertices of the 3D bounding boxes, and one for the centroids. Similarly, there are eight vector areas representing the direction from each of the 8 vertices to the respective centroid, as same to [17], to give the detection of many instances of the same type of object (In their experiments, w = 640, h = 480).

The network works in steps, with each stage work on not only the image features but also the results of the immediately backward stage. Since all steps are convolutional, they influence an increasingly bigger active accessible field as data permit from end to end the network. This belonging allows the network to solve uncertainties in the initial phases due to trivial receptive areas by incorporating progressively higher quantities of framework in later phases.

Image features are calculated by the first ten layers from VGG-19 [18] (pretrained on ImageNet), trailed by two 3*3 convolution layers to reduce the feature dimension from 512 to 256, and from 256 to 128. These 128-dimensional features are fed to the first stage consisting of three 3*3 *128 layers and one 1*1*512 layer, trailed by either a 1*1*9 (belief maps) or a 1*1*16 (vector fields) layer. The residual five stages are similar to the first phase, except that they receive a 153 dimensional input (128 +16 +9 = 153) and comprise of five 7*7*128 layers and one 1*1*128 layer before the 1*1*9 or 1*1*16 layer. All phases are of size w=8 and h=8, with ReLU activation functions interleaved throughout.

B. Detection and pose estimation

After the network has worked on an image, it is needed to extract the each objects from the belief maps. In relevance to other method in which difficult architectures or procedures are needed to distinguish the objects [19], their method follows on a simple post processing method that finds for local peaks in the belief maps above a threshold value, trailed by a greedy assignment algorithm that associates projected vertices to detected centroids. For individual vertex, this later step checks the vector field analysed at the vertex with the direction from the vertex to individual centroid, pointing the vertex to the nearest centroid within this angular threshold of the vector.

Since the vertices of individual object occurrence have been founded, a PnP algorithm [20] is used to perceive the pose of the object, same to [21]. This method uses the detected projected vertices of the bounding box, the camera intrinsics, and the object dimensions to restore the last translation and rotation of the object with respect to the camera.

C. Data generation

The important question put forward by this research is how to create a good training data for the network. In respective to 2D object finding, for which labeled bounding boxes are almost easy to annotate, 3D object finding requires labeled data that is like impossible to create manually. Any way it is possible to semi-automatically label data (using a tool such as Label Fusion [22]), the man power intensive style of the purpose however associates the power to create training data with needed variation.

The photorealistic data was generated by putting the YCB object models in various virtual environments. All data were made by a custom added plugin developed for Unreal Engine 4 (UE4) called NDDS [23]. By allowing distinctive,

adaptable successive frame screenshotting, the plugin creates data at a rate of 50–100 Hz, which is really faster than either the in-build UE4 screenshot function or the publicly available Sim4CV tool [24].

IV. Experimental Results

The main goal of project is robotic pick and place of domestic objects in the Falling Things Dataset (FAT: A Synthetic Dataset for 3D Object Detection and Pose Estimation) which is available in open source to download. In the dataset that is used for training has photorealistic images with precisely annotated 3D pose for all objects. Such a dataset is generated by synthetically overlaying object models and backgrounds with difficult composition and high graphical feature. Their dataset contains 60k annotated photos of 21 household objects taken from the YCB dataset. For each image, they provide the 3D poses, per-pixel class segmentation, and 2D/3D bounding box coordinates for all objects. To provide trails in various input environments, they provide mono and stereo RGB images, along with registered dense depth images. So I used this dataset for training the selected object like cracker, spam, and soup from this dataset.

Fig. 1. The subset of 21 YCB Objects selected to appear in our dataset

A. Training

The training procedure was used with ~60k photorealistic image frames. We selected object Cracker from the 21 objects of FAT dataset. Each object comprises of three virtual environments. In that each environment there are five manually specific locations covering variety of landscape and lighting conditions. Together these had 15 folder locations for each objects. We applied these object folder as the training data input and output of each epoch is stored in separate folder. From the results, the epoch with minimum loss function is selected as the final weights for DOPE algorithm to be detected. These steps are repeated for other two objects.

Their network was implemented using PyTorch v0.4 [25]. The VGG-19 feature extractions were taken from publicly available trained weights in torchvision open models. The networks were trained for 120 epochs with a batchsize of 16. Adam [26] was used as the optimizer with learning rate set at 0.0001. The system was trained on an Dell Precision7820 workstation (containing NVIDIA Quadro P4000), and testing used an Google Colab VM.

B. Robotic manipulation

For our project, the final test of a pose finding method is whether its precision is acceptable for robotic grasping. We attached an Intel Realsense D435i depth camera to the 6th link of a Aubo i5 robot, and calibrated the camera to the robot base using a standard auruco-board target visible to camera. The parallel jaw gripper moves from an opening of approximately 10 cm to 6 cm, or from 8 cm to 4 cm, depending on how the fingers are attached. Any way the moving distance of the gripper is just 4 cm, says that the error can be no more than 2 cm on both side of the object during the grasp.

For better results, we took the object cracker, almost random, at 4 various areas on a table in front of the robot, at 3 various orientations for each of the 4 places. The robot was programed to go to distance before grasp coordinates above the object. The prototype object detected successfully in various orientations. We rotated the objects 6-DOF and the object detection was quite successful. The figures are shown in below.

The robotic manipulation of the project is in the undergoing stage and it will be fully made into grasp the respective object from the randomly ordered bin.

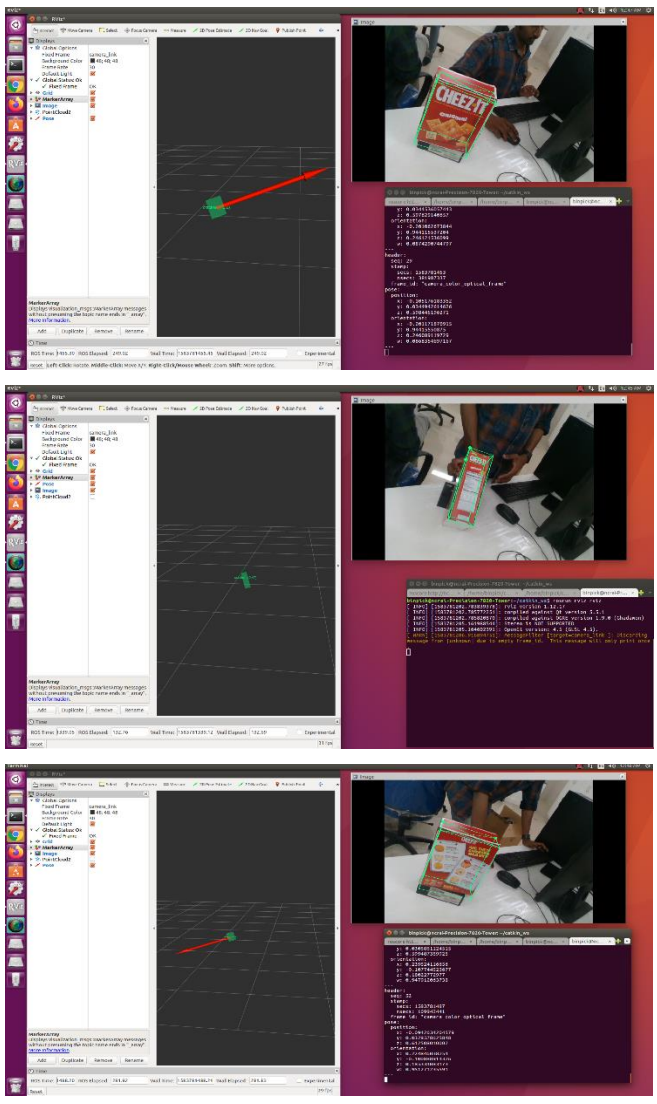

Fig. 2. Detection and pose annotation of cracker object

V. CONCLUSION

I have made package for detecting and estimating the 6-DoF pose of known objects using a novel architecture and data generation pipeline using the state of the art algorithm DOPE in Aubo i5 collaborative robot with Intel Realsense D435i camera. The neural network consists several steps to refine and estimates of the 2D coordinates of projected vertices of each object's 3D bounding cuboid. These vertices are then used to output the final pose using PnP, with known camera intrinsic and object dimensions. We have made that a neural network trained only on photorealistic data can attain state of the art results compared with a neural network trained on real world data and resulting poses are with much needed accuracy for robotic pick and place.

VI. FUTURE SCOPE

As a future scope of this project, we are focusing on creating a custom dataset for the objects in our problem domain using NDDS (NVidia Deep Learning Data Synthesizer) and testing in DOPE algorithm.

AKNOWLEDGMENT

This work has been supported by Nodal Centre for Robotics and Artificial Research (NCRAI) of Government Engineering College, Thrissur.

REFERENCES

[1] The Economist. The growth of industrial robots - Daily chart. 2017. url: https:// www.economist.com/blogs/graphicdetail/2017/03/daily-chart-19 (visited on 03/09/2018).

[2] RobotWorx. Advantages and Disadvantages of Automating with Industrial Robots. https://www.robots.com/blog/viewing/advantages-and-disadvantages-of- automating-with-industrial-robots (visited on 03/06/2018).

[3] RIA - Robotic Industries Association. Robotics Industry Insights - The Pervasive Rele- vance of Bin Picking in Nature and Business. 2011. url: https://www.robotics.org/ content-detail.cfm?content_id=3080 (visited on 03/07/2018).

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deepresidual learning for image recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016,pp. 770–778.

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, BernhardFirner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel,Mathew Monfort, Urs Muller, Jiakai Zhang, et al., End to endlearning for self-driving cars, arXiv preprint arXiv:1604.07316 (2016).

[6] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (2016), no. 7587, 484–489.

[7] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen, Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, arXiv preprintarXiv:1603.02199 (2016).

[8] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker Jr, Alberto Rodriguez, and Jianxiong Xiao, Multiview self-supervised deep learning for 6d pose estimation in the amazon picking challenge, arXiv preprint arXiv:1609.09475 (2016).

[9] Paul J Besl and Neil D McKay, Method for registration of 3-d shapes, Robotics-DL tentative, International Society for Optics and Photonics, 1992, pp. 586–606.

[10] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, and Jianxiong Xiao, 3dmatch: Learning the matching of local 3d geometry in range scans, arXiv preprint arXiv:1603.08182 (2016).

[11] Paul Wohlhart and Vincent Lepetit, Learning descriptors for object recognition and 3d pose estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3109–3118.

[12] Alexander Toshev and Christian Szegedy, Deeppose: Human pose estimation via deep neural networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1653–1660.

[13] Sijin Li and Antoni B Chan, 3d human pose estimation from monocular images with deep convolutional neural network, Asian Conference on Computer Vision, Springer, 2014, pp. 332–347.

[14] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects, 2nd Conference on Robot Learning (CoRL 2018), Z¨urich, Switzerland.

[15] S.-E.Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In CVPR, 2016.

[16] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In CVPR, 2017.

[17] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In RSS, 2018.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[19] M. Rad and V. Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In ICCV, 2017.

[20] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. International Journal Computer Vision, 81(2), 2009.

[21] B. Tekin, S. N. Sinha, and P. Fua. Real-time seamless single shot 6D object pose prediction. In CVPR, 2018.

[22] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake. LabelFusion: A pipeline for generating ground truth labels for real RGBD data of cluttered scenes. In ICRA, 2018.

[23] T. To, J. Tremblay, D. McKay, Y. Yamaguchi, K. Leung, A. Balanon, J. Cheng, and S. Birchfield. NDDS: NVIDIA deep learning dataset synthesizer, 2018. https://github.com/NVIDIA/Dataset_Synthesizer.

[24] M. Mueller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem. Sim4CV: A photo-realisticsimulator for computer vision applications. International Journal of Computer Vision, pages 1–18, 2018.

[25] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison,L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In NIPS Autodiff Workshop, 2017.

[26] L. Rupert, P. Hyatt, and M. D. Killpack. Comparing model predictive control and input shapingfor improved response of low-impedance robots. In IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2015.

[27] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine. End-to-end learning of semantic grasping. In CoRL, 2017.