# Visual Feedback based Trajectory Planning to Pick an Object and Manipulation using Deep learning

**Shraddha Chaudhary, Shobhit Zakhami, Sumantra Dutta Roy**
Department of Electrical Engineering
Indian Institute of Technology Delhi
chaudhary.shraddha18@gmail.com,shobhitzakhmi1@gmail.com,sumantra@ee.iitd.ac.in

## ABSTRACT

An automated approach to navigation and manipulation of general objects using YouBot has been discussed in this paper. The paper presents a machine learning based novel end to end solution to the object manipulation problem. Kinect sensor is placed at the base of the youBot for the sensing operation. This helps in the reconstruction of the environment and thereby makes it possible to stop near the target to be picked. A camera is mounted in eye in hand configuration on the YouBot. To perceive the depth of the target, an initial estimate is taken with a stereo image pair using a single camera, thus eliminating the need of multiple cameras or other sensors for depth estimation. A machine learning approach is then used to determine the grasping point of the object. Path planning plays an important role in the overall problem, therefore an adaptive visual servoing based solution is employed to make the picking solution robust. Parameters for trajectory planning are optimized by minimizing the error between the initial and the desired configuration while respecting the systems constraints. Further to make the algorithm robust, the validation is done on the COIL 100 dataset. Hence, this paper presents robust and complete solution to the navigation and manipulation of the objects using 8 Degree of freedom KUKA youBot.

## CCS CONCEPTS

• **Computer System Organization, Embedded and cyber-physical systems, Sensor Networks, Robotics, Real time System**;

## KEYWORDS

Computer vision, Machine Learning, Deep Learning, YouBot

## 1 INTRODUCTION

Automation is the process of coordination between our senses and motor control system in robots. This process involves several challenges like high dimensional configuration spaces make planning and executing the motions required for the tasks difficult. Robot manipulators are widely used in industrial automation as referenced in [1], such as metal parts packing, food and beverage packing [2]; [3]; [4]. Recently, human robot collaboration for assembly task has attracted a lot of research attention. In such tasks, the real-time object recognition and pick- place operations are crucial, and small size robot manipulators are utilized by non-experienced users in assembly tasks. In Reference [5] operator can complete tasks faster if the robot collaborates with the operator has been emphasized. For this purpose, the robot needs to navigate to the desired position, recognize and pick-up the right objects. Another aspect of this paper is using machine learning for visual servoing. The goal of visual servoing techniques is to control a dynamic system, such as a robot, by using the information provided by single or multiple cameras as shown in the setup Fig.1.


*©2019 AIR'19, July 2019, Chennai, India*
https://doi.org/xxxxxxxxx...$15.00

### 1.1 Related Work

A unified closed form Inverse Kinematics solution by presenting the youBot as an 8 DOF mobile manipulator with a 5 DOF arm and a 3 DOF moving base has also been obtained in [6]. The paper suggests a few redundancy parameters in the approach and appropriately takes them into account to get to a generalised final solution.

Object recognition has been research topic since long time. Reference [7] used generated cylinders for object segmentation and recognition. Recent methods based on deep learning [8], [9], [5] and [10] have demonstrated state-of-the-art performance in a wide variety of tasks. In [11], two separate deep belief networks were trained for object classification. A real-time grasp detection method based on convolutional neural networks is proposed in [12]. RGB-D depth image of Kinect sensor was used in [13] besides a two-step cascade structure of deep networks for robot grasping. A single object was used at a time for recognition. Reference [14] trained a convolutional neural network for predicting the grasp locations using trial-and-error experiments. The grasping model was improved by collecting additional data from Kinect and high-resolution camera during the real-time experiments. Authors in [15] proposed a multi-modal deep extreme learning machine structure for feature extraction and object recognition.

The basics of Visual Servoing algorithms have been provided in the tutorial by Chaumette and Hutchinson in [16], which describes the Classical Image based and Position based Visual Servoing techniques and the mathematics behind them. The second part of the above tutorial describes more recent and advanced approaches in the field. Another state-of-the-art approach which uses a depth estimation algorithm has been discussed in [17].

The use of Machine Learning to get over the problems of Classical Visual Servoing is the current topic of research. The primary problems of the old technique have been discussed and an Extreme Learning Machine to get over the same is presented in [18]. Another paper by [2] uses Neural Network to perform Visual Servoing. A state-of-the-art paper [19] employs Deep Neural Networks to

**Figure 1: Experimental Setup: Demonstrating a YouBot and an object lying in front of the youBot, that is to be picked.**

achieve trajectory planning of the manipulator. The paper describes in detail the method of dataset generation besides the addition of perturbations to the same through simplified approaches. Classical approaches to visual servoing rely on extraction, tracking and matching of a set of visual features. These features, generally points, lines, or moments, are used as inputs to a control law that positions (or navigates) the robot in a desired pose.

### 1.2 Main Contributions

We propose a path planning approach using neural networks with the following improvements and some limitations:

- Our proposed system reduces the cost of calculation using just single camera based visual servoing and omitting the need to include intrinsic matrices while generating depth map. Thus being more reliable for simple objects.
- Object position is constantly updated, thus we can cater to the objects in a dynamic environment and making the system more robust.
- End effector coordinates are calculated more efficiently using the system trained for inverse kinematics of the serial link robotic arm rather than using conventional methods which are based on nonlinear calculations.

Therefore, our work is focused on developing an improved method for object detection and manipulation using machine learning based visual servoing.

## 2 OBSTACLE DETECTION AND NAVIGATION

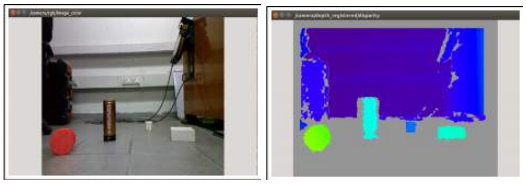
Our first objective is to automate the base of YouBot so that it can reach the object. Robot Operating System is used to plan the motion of the robot with the help of various inbuilt packages in its framework. We start by localizing the position of YouBot in given environment and mapping it. This is achieved with the help of 4D data stream from Microsoft Kinect. Once the map and pose is obtained, it searches for the target object. Object recogniton is done by applying machine learning on thee frames of video input from Kinect. After finalizing the object and thus, the goal, a path is planned. Velocity commands necessary to follow the path are published to base which then moves to begin towards the goal as shown in the Fig. 1.

### 2.1 3D Mapping and Localisation

In order to navigate the object to the desired location, a 3D view of its surroundings needs to be created. The rtabmap ros ROS package [14] is used to achieve the same. This uses the Kinect 4D video stream as the input to generate a continuous 2D occupancy grid map. This package basically uses Real time Appearance based Mapping which is used to construct a map of the surroundings and simultaneously track the location of the object within it.

### 2.2 Object Recogniton

Once, the map of the surroundings is created and the YouBot is localized within it, the next part is to recognize the object of interest. For this, the find object 2D ROS package [20] is used. The package basically contains BRIEF, FAST, SIFT, SURF and other feature detectors implemented in OpenCV. The matching of features in the scene generated by Kinect and the pre-fed image is done continuously. Once, the matching is successful, the pixel coordinates of the matched points are generated. By using the depth data from Kinect for the generated pixels as shown in Fig.2, we obtain a goal.



**(a) Kinect RGB Image.** **(b) Kinect Depth Image.**

**Figure 2: Images taken by the Kinect, that is placed at the platform of the YouBot.**

A problem with this algorithm is that the object of interest needs to be exactly in the same orientation for matching to be successful. If the pose of the object is not same as the pose of the object in the target image, the robot bypasses the object and moves forward. This weakness of the current technique has been targeted and a strong recognition algorithm has been proposed in the paper.

The Algorithm can be described as follows:

(1) **Feature Extraction:** Feature extraction is done using the Scale Invariant Feature Transform (SIFT) which gives a large number of descriptors of 128 dimension each for every image.
(2) **Feature Encoding:** Using the SIFT descriptors for each image as input, K-means clustering is used to form clusters and the cluster-centroids called Words of Vocabulary are generated.
(3) **Bag of Words:** Each feature of an image is mapped to the cluster center which is at the minimum distance from it. Thus, each cluster contains a bag of features associated with it. A normalized histogram is obtained for each image.
(4) **Feature Classification:** The histograms now act as features and using the known labels, one vs all SVM is used for multiclass classification. The number of SVM's equal to the number of categories are trained and the one which has maximum confidence or distance from the margin is the one to which a test feature is assigned.

**Figure 3: COIL-100 Dataset Images**

**Figure 4: Real Dataset image**

## 2.3 Motion Planning

Now having tracked the location of the object using the Object recognition algorithm, the robot is just required to move closer to it. This is done using the move base ROS package. The package uses a global and local cost planner which calculates the cost of various possible trajectories and selects the best out of them as shown in Fig. 5. The following algorithm is used:

(1) The Robots current control space (dx, dy, dz) is sampled.
(2) Forward simulation is used to predict the state of the robot if a sample velocity is applied for a small time step at each sample.
(3) The score of each trajectory is calculated using metrics that incorporate various characteristics like closeness to obstacles and goal, deviation from global path etc.
(4) The trajectory which has the highest score is selected and the corresponding sample velocity is applied to the robot.
(5) The above steps are repeated for the new position.

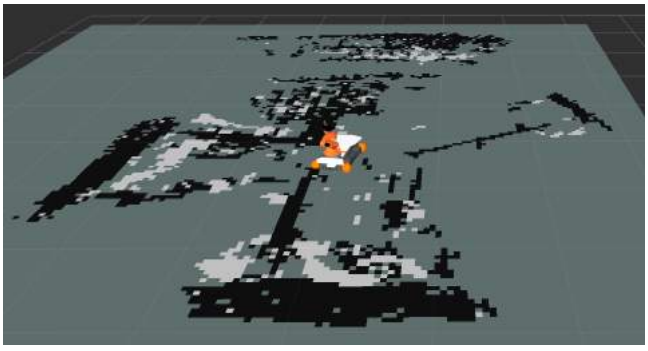


**Figure 5: Map generation in Gazebo simulator**

# 3 POSE ESTIMATION USING MONOCULAR CAMERA MOUNTED ON THE YOUBOT ARM

Next step is to determine how to pick up the target object. A simple camera is mounted on the YouBot arm and then depth data is obtained from multiple images. The depth image is then feed into neural network and subsequently the grasp points are estimated.

## 3.1 Stereo Camera Calibration

Stereo camera calibration is necessary for the depth estimation. This was done using the Stereo Camera Calibration app available in the Image Processing and Computer Vision Toolbox of as shown in Fig. 6.

The calibration part is required for two purposes:

(1) Reconstruction of 3D scene: This function is used to calculate the actual depth of the image points using the disparity map Fig. 2.
(2) Rectification of images: This is used to bring the corresponding points in the left and right images in the same horizontal row. It is an essential step before 3D reconstruction.

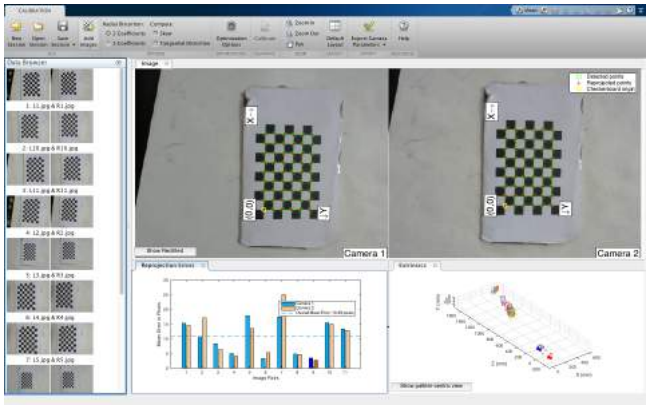



**Figure 6: Stereo camera calibration using MATLAB app**

## 3.2 Depth estimation

In order to calculate the depth of points in the image, firstly a disparity map is constructed using left and right images of the object scene as shown in Fig. 2. Block Matching algorithm [18] is used to calculate the disparity by comparing the sum of absolute values of differences of each block of image. Once the disparity map is obtained, the reconstructscene function is used to obtain the depth values with the use of stereo camera parameters.

## 3.3 Grasp Detection

Once, the depth image is obtained using the above techniques, the original algorithm is employed. The training is done on the Cornell grasping dataset mentioned in[13]. The dataset has a total of 1035 images of 280 objects. Some of the objects of the dataset are shown in Fig.8. An instance of the running detection algorithm is shown in Fig. 4. The yellow-green lines represent the best rectangle for the grasping point as shown in Fig. 7

# 4 OBJECT PICK AND PLACE USING IMAGE BASED VISUAL SERVOING

The Visual Servoing Algorithm is a closed loop control system in which continuous feedback through the vision sensor ensures that the manipulator always moves in the right direction continuously minimizing the error.

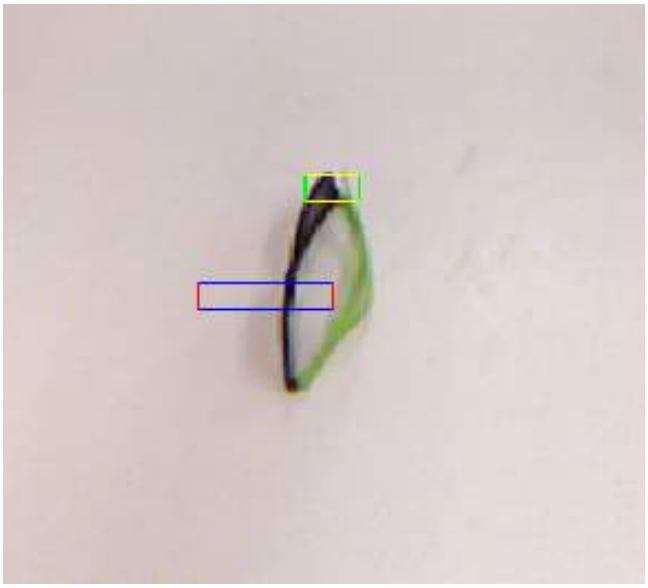


**Figure 7: Grasp Detection Algorithm: The blue-red lines represent the current rectangle whereas the yellow-green lines represent the best rectangle till that instant**

**Figure 8: Objects in Cornell grasping dataset**

**Algorithm 1** Flow of Neural Network

1: **procedure** GRASP POINTS DETECTION
2: **Load Data:** Dataset Loaded for training.
3: **Data processing:** Splitting data into training and testing followed by whitening is done.
4: **Training:** The parameters of the model including the weights are learnt and stored.
5: **Detection:** The learnt model is now used to detect the grasping points in the input images.

The use of Machine Learning makes the new algorithm better as compared to the classical counterparts in term of time reduction and easy implementation. The algorithm not only bypasses the Visual Servoing formulation but also avoids the heavy non linear computation of Inverse Kinematics of the KUKA YouBot in our case. The following sections discuss in detail the various Machine Learning algorithms used to solve the regression problem starting with the dataset construction.

The Machine Learning Algorithm that we aim to formulate uses the current position of the end effector in joint angles domain and the target image centroid pixel coordinates in the image plane as input and returns the change in joint angle values as output. The end effector is fixed to stay in the horizontal orientation throughout this algorithm. The z coordinate of the centroid is also fixed for this algorithm. The regression function can be represented as

$$\Theta_1, \Theta_2, \Theta_3, \upsilon_c, \nu_c \rightarrow \Delta\Theta_1, \Delta\Theta_2, \Delta\Theta_3 \quad (1)$$

In order to run regression based machine learning, a large dataset is required to find out the optimal parameters of the model. The dataset is generated using forward and inverse kinematics of the KUKA YouBot as follows:

(1) Using the initial joint angles of the YouBot manipulator, forward kinematics is used to calculate the Cartesian coordinates of the gripper
(2) The camera calibration parameters are used to calculate the world cartesian coordinates of the centroid with respect to the current and target image both.
(3) The delta position vector in Cartesian system is calculated as the difference of the above two. This is basically the direction of translational velocity vector.
(4) Multiplying the velocity vector with the time step provides the change in position of the end-effector.
(5) The change in position vector is added to the original position vector to obtain the new position vector.
(6) Finally, Inverse Kinematics is used to calculate the new joint angles of the manipulator and change in angles can be calculated.

In order to learn a highly non linear input-output relationship as in our case, complex algorithms like Neural Networks and Support Vector Regression are used. Using regression based learning approach to learn the trajectory of the pick up and thus eliminating the need for complex computation speeds up the process. This is very helpful during real time applications. Results obtained using these algorithms is shown in section 5.3.

## 5 EXPERIMENTAL RESULTS

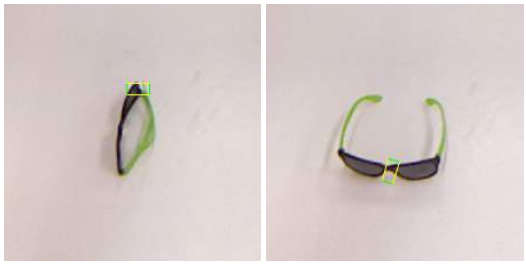
This section of the paper describe the results obtained at different stages to pick-up an object using youBot.

### 5.1 Grasp Detection

In the training part of the algorithm, the former network uses 200 hidden units to generate top 100 rectangles whereas the latter one uses 50 units to find the best out of them. The following results are obtained on the training and testing parts of the dataset:

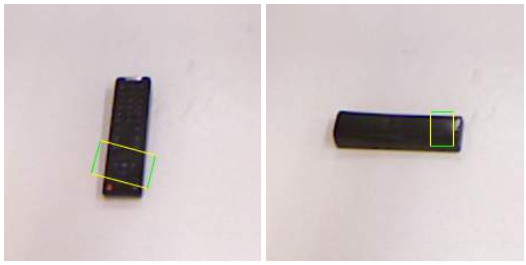
(1) Training: No. of miss-classifications: 0 from 6415 (100.00%).
(2) Testing: No. of miss-classifications: 91 from 1604 (94.33%)

The results on some of the objects from Cornell Grasping dataset are as shown in Fig. 9, 10 and on the real dataset are shown in Fig. 11, Fig.12 The green lines correspond to the gripper plates. Green and yellow rectangle in the figures is the grasp point, ans is well placed for all the real time objects. This shows that algorithm works efficiently on real time objects.



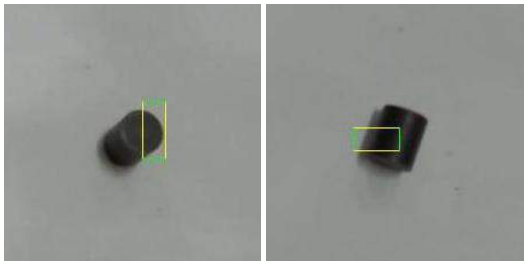
(a) Spectacle View 1 (b) Spectacle View 2

**Figure 9: Results for Grasping Point on Spectacles**



(a) Remote View 1 (b) Remote View 2

**Figure 10: Results for Grasping point on the Remote**



(a) Standing Pellet View (b) Fallen Pellet View

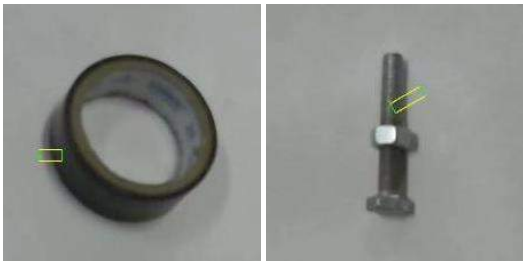
**Figure 11: Results for Grasping point on the Standing and sleeping Pellet.( Cylindrical Texture-less object)**

(a) Pen View (b) Pencil View

**Figure 12: Results for Grasping point on the pen and pencil**

## 5.2 Orientation Detection During Pickup

Visual feedback based pick-up is demonstrated in the Fig.14. Each time robotic arm moves to the new position, target and current image



(a) Tape View (b) Screw View

are compared. Hence, a Constant feedback is taken while picking up the object, to accommodate for changes in the target and current position of the object. In Fig. 15 it is seen that orientation of the marker pen is changed, so to supply the robot with the desired position, image based visual servoing is done. While the features are matched using SURF algorithm. This gives robustness to the system while pick-up.

(a) Home Position (b) Position during Pick-up

**Figure 14: YouBot Positions while picking up the target object**

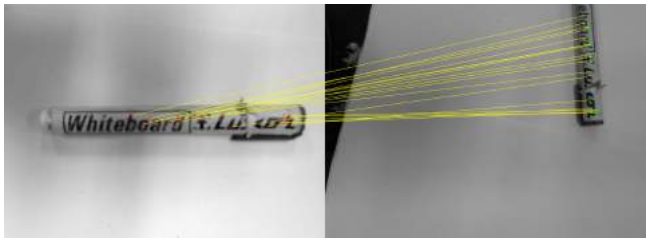



**Figure 15: Matched points during pick up for visual servoing.**

## 5.3 Machine Learning Approach to Object Picking

The Neural Network trained is implemented using MATLAB. The optimal results are obtained by choosing the number of hidden units to be 15. Levenberg-Marquardt optimization algorithm is used. The R values calculate the correlation in between the target and output images.Fig. 16 The results of the SVR algorithm are also achieved through its MATLAB implementation. K-fold cross validation is used with K=6 on 90,000 data points (75,000 training and 15,000 testing) to compare with NN (75,000 training, 10,000 validation and 15,000 testing) is shown in Fig. 17

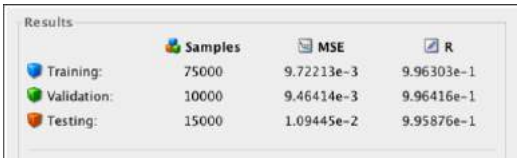



**Figure 16: Mean Square Error and R value for the NN model**

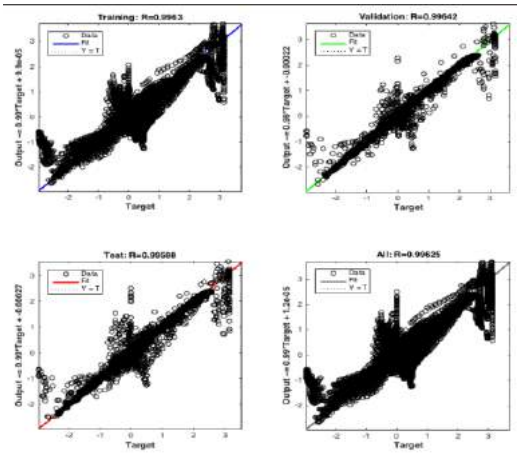



**Figure 17: Regression plots for Neural Network model**

The comparison of the mean square values of the two models shows that the results obtained using the Neural Network are much better as compared to SVR. The mean square error obtained with NN is one third of the error obtained by SVR is shown in Fig. 18

| | |
|---|---|
| A | *1524410x8 double* |
| fileID | 4 |
| formatSpec | '%f %f %f %f %f %f %f %f' |
| MdlGau1 | *1x1 RegressionPartitionedSVM* |
| mseGau1 | 0.3164 |
| sizeA | [8,Inf] |
| X | *90000x5 double* |
| Y | *90000x3 double* |
| Y1 | *90000x1 double* |
| Y2 | *90000x1 double* |
| Y3 | *90000x1 double* |

**Figure 18: Results of the Support Vector Regression Model**

## 6 CONCLUSION AND FUTURE WORK

An end to end solution to navigate to an object and effectively pick it up from appropriate grasping points using the KUKA youBot Mobile Manipulator is proposed in this paper. State of the art algorithms were employed at various stages to meet the objectives. The algorithms were implemented through the available open source literature and softwares. The Grasp Detection Algorithm uses deep learning to obtain the optimal grasping rectangle for the 2 finger gripper. High accuracy is obtained on the training and testing dataset and the results map well to real data. The basic Image based Visual Servoing algorithm is implemented and tested on the robot. The inherent problems in the classical implementation are tackled by using a machine learning based approach which uses Neural Network and Support Vector Regression and compares the results obtained by the two. Direct depth estimation is not possible with the 2D Webcam available which forces the use of Stereo Vision for the same. Hence, integrating the above systems has led to the creation of an overall improved system for object pickup using the KUKA youBot.

The proposed approach is highly practical and efficient for building autonomous system to perform various tasks through manipulation and navigation for simple objects. Such systems can be employed in domestic sectors with some improvement for daily tasks like serving food and beverages etc. Another application in the professional sphere would be exchange of memorandums and parcels between different departments of the same organization.

## REFERENCES

[1] Torgny Brogaardh. Present and future robot control development - an industrial perspective. *Annual Reviews in Control*, 31:69–79, 2007.
[2] Yang Yanxi, Ding Liu, and Han Liu. Robot-self-learning visual servoing algorithm using neural networks. volume 2, pages 739 – 742, 02 2002.
[3] Edgar Boef, Jan Korst, Silvano Martello, David Pisinger, and Daniele Vigo. Erratum to the three-dimensional bin packing problem: Robot-packable and orthogonal variants of packing problems. *Operations Research*, 53:735–736, 08 2005.
[4] P.Y. Chua, T Ilschner, and D.G. Caldwell. Robotic manipulation of food products - a review. *Industrial Robot: An International Journal*, 30:345–354, 08 2003.
[5] Zulkifli Mohamed, Mitsuki Kitani, and Genci Capi. Adaptive arm motion generation of humanoid robot operating in dynamic environments. *Industrial Robot: An International Journal*, 41, 03 2014.
[6] Shashank Sharma, Gerhard Kraetzschmar, Christian Scheurer, and Rainer Bischoff. Unified closed form inverse kinematics for the kuka youbot. pages 1–6, 01 2012.
[7] Ramakant Nevatia and Thomas O. Binford. Description and recognition of curved objects. *Artif. Intell.*, 8(1):77–98, February 1977.
[8] Mohammad Ali Keyvanrad and Mahdi Homayoonpoor. A brief survey on deep belief networks and introducing a new object oriented matlab toolbox (deebnet v2.0). 08 2014.
[9] Geoffrey E. Hinton and Simon Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.
[10] Biao Leng, Xiangyang Zhang, Ming Yao, and Zhang Xiong. 3d object classification using deep belief networks. In *Proceedings of the 20th Anniversary International Conference on MultiMedia Modeling - Volume 8326*, MMM 2014, pages 128–139, 2014.
[11] Dong Liang, Kaijian Weng, Can Wang, Guoyuan Liang, Haoyao Chen, and Xinyu Wu. A 3d object recognition and pose estimation system using deep learning method. pages 401–404, 04 2014.
[12] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *International Conference on Robotics and Automation (ICRA)*, 2015.
[13] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps, 2016.
[14] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. 09 2016.
[15] Jie Wei, Huaping Liu, Gaowei Yan, and Fuchun Sun. Robotic grasping recognition using multi-modal deep extreme learning machine. *Multidimensional Systems and Signal Processing*, 28, 03 2016.
[16] Francois Chaumette and Seth Hutchinson. Visual servo control, part i: Basic approaches. *IEEE Robot. Autom. Mag.*, 13, 01 2006.
[17] M. Keshmiri and W. Xie. Visual servoing of a robotic manipulator using an optimized trajectory planning technique. In *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–6, May 2014.
[18] Tolga Yuksel. Intelligent visual servoing with extreme learning machine and fuzzy logic. *Expert Systems with Applications*, 72, 10 2016.
[19] Quentin Bateux, Eric Marchand, Juxi Leitner, FranÃğois Chaumette, and Peter Corke. Visual servoing from deep neural networks. 05 2017.
[20] Labbé, M. Find-Object. http://introlab.github.io/find-object, 2011.