**Amazon Picking Challenge 2016**

**Picking task:**

Challenge:

* Objects given are 40. Few objects had redundant copies.
* Objects were placed in the rack based on photograph by the organizers.
* JSON file will be given where selected objects need to be picked from selected bin and placed in tote.
* Objects were not too cluttered and were not placed too deep into the bin
* Rack is made of steel plates, and boxes were made of cardboard

Flow of robot:

1. When robot is turned on, robot moves and face the rack.
2. Robot has 7 DOF
3. Arm has Kinect mounted on it. The RGB camera on Kinect detects the 4 corners of the rack.
4. Once corners are detected, all the 12 bins a detected.
5. Based on the rack orientation, robot localizes itself.
6. Now as per JSON file robot will go to the respective bin and stops.
7. Robot orients itself so that it can have a clear view of the bin.
8. Now RGB image of the bin is take and it is sent to Faster R-CNN (Region based Convolutional Neural Network).
9. Faster R-CNN detects the object of interest and draws a bounding box around it.
10. Now point cloud inside this bounding box is taken.
11. We already have taken a 360 degree point cloud data of almost all the objects. Each object already has 5-6 point cloud images in database.
12. For some objects like mesh cup and polythene bag, point cloud is not possible because mesh cup cannot have point cloud at all, whereas polythene back doesn’t have a fixed shape. So matching shape is not possible.
13. Shape back projection is used and exact location of the object is figured out based on point cloud and centroid of the object is extracted.
14. This co-ordinate is given to the motion planner which reaches that location.
15. Once target area is reached the vacuum pump is turned on, which grips the object.
16. Arm then puts the object in the tote.

Challenges in Perception:

1. As objects were not too cluttered challenge was relatively easy.
2. During 2016 scemantic segmentation concept was not there. So only bounding boxes were available. Object could be anywhere in the bounding box. That is the reason why point cloud image was needed to figure out the object in the bounding box and take the centroid of it.
3. VGG was used as CNN for Faster R-CNN. Initially ZF net was used but results were not so good.
4. Training dataset was collected by keeping objects in the bin alone and with other few objects as well. Then a rectangle was drawn around these objects do mark object of interest. Almost 1000-2000 images per object were taken. Total images were around 80000.
5. When training only the last layer where all the classes are there will be replaced with new set of classes and model will be trained with the data. Training is done completely, no particular layers are freezed and trained. Reason being, we do not know what will be the effect if certain layers are freezed and other layers are trained.
6. Generally convergence happen very soon with a pre-trained network. So training all layers does not add much of an overhead.
7. Due to metal surface of rack, 3D point cloud from Kinect-1 was not coming proper. Kinect-2 point cloud was not that good compared to Kinect-1. RealSense R200 was also tried but results were not so good.
8. The winning team used Ensenso Depth camera. Their point cloud was really good.
9. Few objects were black listed e.g. mesh cup, as we couldn’t detect it well and we couldn’t grip it.

About Faster R-CNN:

* Run the image through a CNN to get a Feature Map
* Run the Activation Map through a separate network, called the Region Proposal Network(RPN), that outputs interesting boxes/regions
* For the interesting boxes/regions from RPN use several fully connected layer to output class + Bounding Box coordinates.

Read through the below blogs.

<https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/>

<https://medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8>

Challenges in Embedded:

* When motion planner gets the coordinate, the tip of the arm goes till the point and turn on vacuum. Say the object moved a bit and vacuum is turned on, now vacuum will catch the metal rack base plate. Now if the arm tries to pull the rack, due to safety precautions in robot the robot will be powered off automatically.
* This problem was solved by having a metal plate on one end of bellow in which 5V was being sent. On the other side of the bellow another metal plate is connected which was pulled down. Now if the bellow touch the metal plate the 5V metal plate will transfer current to the open metal plate. The ADC reading will change because of which gripping of metal plate can be detected.
* When an object is picked, there is a chance that the robot may drop the object on the way. In that case completing the motion planning is a waste of time. So an IR sensor is fixed to the tip of the robot gripper.
* Only the transmitter and receiver are kept at the tip. Rest of the circuit is kept at the back. This is done to avoid blocking of air flow.

About Robot:

Robot used was a Barret arm with 7 DOF

