Lekha Walajapet Mohan

Team D: HARP

Teammates: Alex Brinkman, Rick Shanor,Abhishek Bhatia,

Feroze Naina

ILR10

Mar. 31, 2016

I. Individual Progress:

For Progress Review 11, I was doing a literature survey for the perception system. As we had decided to use the convolutional neural network as a part of our algorithm for perception, we have to evaluate a best network architecture that would detect the object, give us a pose estimate and segment the object from the rest of the image or the point cloud. I am going to discuss about my survey in the following. I will be involved in implementing deep learning algorithms training our net work to output the correct detected item.

Fast Region-based Convolutional Network method (Fast R-CNN) is used for object detection. It is an efficient deep convolutional network, known for its robustness in object detection. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9× faster than R-CNN, is 213× faster at testing phase, and has good average precision on PASCAL VOC 2012. We are currently training our network to evaluate it's performance.

Using 3D models, this framework combines render-based images which when projected on real time training image, are called as synthesized images. We train the synthesized image on real-time images, which are captured by the RGB camera. 3D models can give us large amount of variation which can be trained used by deep learning networks(Fig 1). Owing to the high capacity learning rate by the deep nets, this framework proposes that, view point estimation is robust even under highly varying environmental conditions.

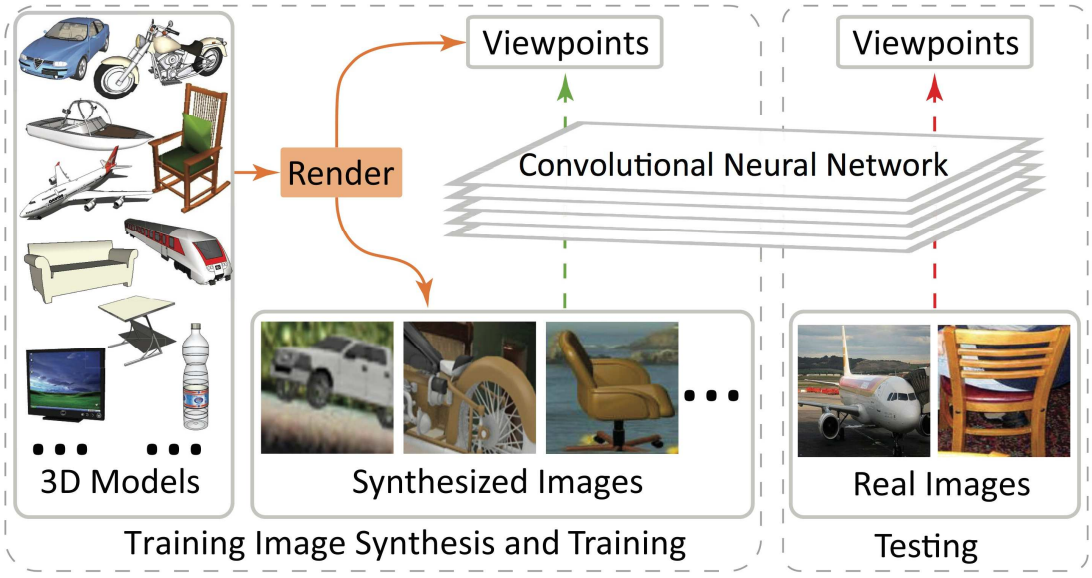

Fig 1: Rendering 3D models on real test images on a CNN network

3D models are rendered in various viewpoints and are highly varied. By variation I mean adding background to it, which in our case could be objects that would represent clutter scene. We will include variation in lightning which will be our noise. Although, we are taking all the steps to generate a robust data set, the hard truth is that, the data set generated is always

biased. In the field of vision, people usually employ mechanical turks to annotate data set after creating them.

Another method which we are trying to implement is viewpoints and keypoints. Viewpoint prediction can also be analyzed as predicting three euler angles (azimuth, elevation and cyclorotation). This proposal is training a CNN based architecture which can capture local patters of euler angles which will compute unique viewpoint estimation. It uses, again the convolutional neural network at varying scales to retrieve the heatmap that corresponds to the likelihood of viewpoint distribution for the given training image.

This allows us to train a CNN which can jointly predict viewpoint for all classes, thus enabling learning a shared feature representation across all categories. We use the Caffe framework [20] to train and extract features from the CNN described above. We augment the training data with jittered ground-truth bounding boxes that overlap with the annotated bounding box with IoU > 0.7. Xiang et al. [37] provide annotations for (φ, ϕ, ψ) corresponding to all the instances in the PASCAL VOC 2012 detection train, validation set as well as for ImageNet images.

Owing to the fact that we have to create huge amount of data set, there is a new video annotation toolkit called VATIC, developed by UC Irvine. It has an optimal way of creating dataset where a video of the scene is created. The video can be annotated easily by playing the video, pausing it at required frame, annotating it. This process gets repeated. The biggest advantage of this process is that it doesn't require manual overload for creating dataset.

Due to time constraints, it is impossible to analyze each and every network. We planned to pick up the best two and parallel train and test them. We also have other algorithms like PERCH to distinguish geometrically varying objects which gives us satisfactory output.

II **Challenges**

Biggest challenge that I faced for this week was to develop a robust algorithm to train our perception network. We still don't have a standard model or data set that we could train and test with. Creating 3D models for all the items in the Amazon dictionary is laborious and we are heavily constrained by time. If the algorithm fails it will be a waste of our time and resources and hence, we need to be very careful about choosing the best network and annotation method for perception

III **Team Work**

We have started testing for single item bin test where Feroze was working on setting up the grasping integration pipeline. Alex was working on trajectory planning and turntable image capturing script, which captures the item on the turntable, processes it by masking off the background. We can now generate approximately 600 training images for perception algorithm. Rick was working on training and testing the Faster RCNN network and setting up

the PERCH algorithm with our technical advisor Venkat. Abhishek was working on RGB-D segmentation for the filtered cluttered objects from the shelf

IV Future Plans

As we are close to our Spring Validation Experiment, we are fine tuning our individual subsystems. Alex and Feroze will be working on extensive fault testing. They will be analyzing various fault cases and debugging the error. Myself, Abhishek and Rick will be working on perception where we will be testing different networks and evaluating their performances. We will also be creating datasets for training and testing our perception algorithm.