Learning to Grasp Objects: A Novel Approach for Localizing Objects Using Depth Based Segmentation

Deepak Rao, Arda Kara, Serena Yeung
(Under the guidance of Quoc V. Le)
Stanford University

Abstract

We consider the problem of grasping novel objects with a robotic arm. A recent successful technique applies machine learning to identify a point in an image corresponding to the most likely location at which to grasp an object. Another approach extends their method to accomodate grasps with multiple contact points. This paper proposes a novel approach that tries to find graspable points in an object after localizing it. We present a depth based segmentation scheme for object localization and discuss how depth information can be combined with visual imagery to improve the performance of both segmentation and object localization.

1. Introduction

The problems of image segmentation and object localization remain great challanges for computer vision, having received continous attention since the birth of the field. The last few years have seen considerable progress in eigenvector based methods of image segmentation. These methods are too slow to be practical for many applications.

The state of the art approach for segmenting an image into regions uses graph based representation of the image[2]. Although, the method works better than its counterparts, it has some limitations. Since the algorithm is based entirely on pixel intensities it works poorly in scenes having shadows or more than one light source.

The task of object localization involves drawing a tight bounding box around all instances of an object class within the image. This paper aims to highlight some of the drawbacks of using color based segmentation for localizing an object. Figure 1 shows an example of an object composed of multiple colors. Performing color based segmentation on the object results in different segments all belonging to the same object. Taking into account these limitations and drawbacks we propose a segmentation framework for object localization that combines depth information with visual imagery (color/intensity). The next section gives a detailed comparison between the two approaches.

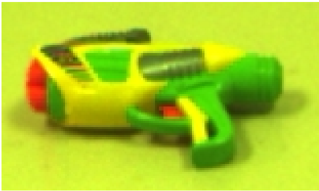


Figure 1: Example of an object composed of muliple colors

The intuition behind using the depth data for segmentation came from [4] in which Quigley et al used high resolution 3D data for improving object detection. We conducted experiments to prove that localizing objects after performing our proposed model of segmentation works better than localization performed after segmentation using [2]. A detailed explanation of our approach along with the experimental results have been given in the following sections.

2. Segmentation

Our system makes use of both depth and colored image data. To capture depth, we used an active triangulation sensor [4]. An important feature of this sensor is that it gives a very detailed depth data (also called depth map or point cloud). Since color is an important attribute for segmentation, we added a functionality in ROS [6] that enabled us to capture 3 channel colored images. Figure 2 gives an idea of tha data captured by our robot.

Graph based image segmentation [2] tries to create a graph based representataion of an image and finds the evidence of a boundary between regions based on L_2-Norm between the intensity of pixels.

$$||I(p_i) - I(p_j)||_2 > \tau \tag{1}$$

where $I(p_i) \in R^3$ is the intensity of the pixel p_i and τ is the threshold function.

If the L_2 Norm between the intensities is greater than

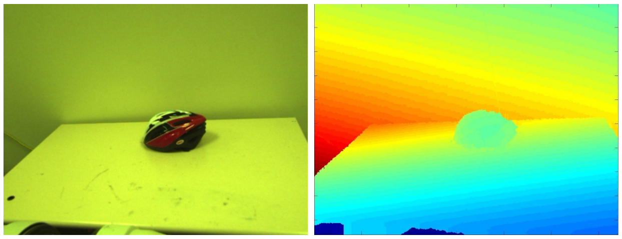


Figure 2: Image and depth data captured by our robot.

the threshold function then the pixels are considered to be in different regions.

As discussed earlier, segmentating an image truly on the basis of color is not the correct approach. The advantage of depth data allowed us to extend the approach in [2]. We stick with the same scheme of representing an image as a graph but updated the metric for finding boundaries between regions to include the depth data.

$$||W^T * (F(p_i) - F(p_j))||_2 > \tau \tag{2}$$

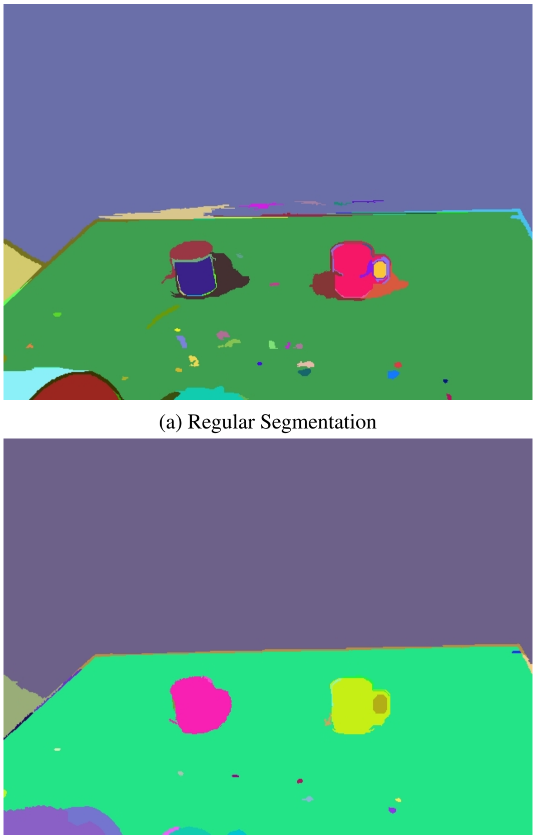
where $F(p_i) \in R^4$ is the intensity of the pixel p_i having an extra dimension of depth value corresponding to that location in 3D space, W $\in R^4$ is the weight vector assigning weights to different elements of F and τ is the threshold function.

The intuition behind including weights in the equation was to rank the elements in order of their importance. This approach allowed us to give greater weight to the depth value and smaller but equal weights to the intensities. Another approach could have been to learn the weights that give the best segmentation but due the comparitively small size of features, we decided to hand tune the weights to produce good results.

2.1. Comparison

Figure 3 shows the comparison between the two segmentation approaches. It can be seen in Figure 3a that athough the method proposed in [2] works well to segment the scene, it is not robust to small color changes. This results in a large number of artefacts on the table and the wall which ideally should have been a part of the single parent segment. The limitations of segmenting an image only on the basis of color can be cleary seen in this example. Another observation that made us to use a combination of depth and color for segmentation was the inability of the base algorithm to handle shadows. All these factors contributed to our hypothesis that segmentation based purely on color would not be ideal for object localization.

In order to prove our hypothesis we decided to localize objects using both the approaches and compare their scores. Object localization involves drawing a tight bounding box around an object. Over the years, the complexity of the problem has led to various approaches and different algorithms to solve the same. This made it difficult to choose a standard recognized scoring system to compare our approaches. All these constraints led us to make our own intuitive scoring scheme that was able to compare the approaches pretty well.



(a) Regular Segmentation

(b) Proposed Segmentation

Figure 3: Comparison between the two approaches.

We took 200 images from the robot having different objects at varied positions to compare our approaches. This was followed by manually labelling the objects using the Image Sequence Labeler in Stair Vision Library [5]. The human label for the object was considered to be the ground truth upon which our scoring scheme is based. Once the labelling was done we ran both the segmentation algorithms on the dataset and made bounding boxes around all the obtained segments. Figure 4 gives an illustration of the same. The green bounding box drawn around the object was the ground truth label.

The approach while designing the scoring framework was based on finding the individual scores for every image and then use those scores to determine the final score of the algorithm.

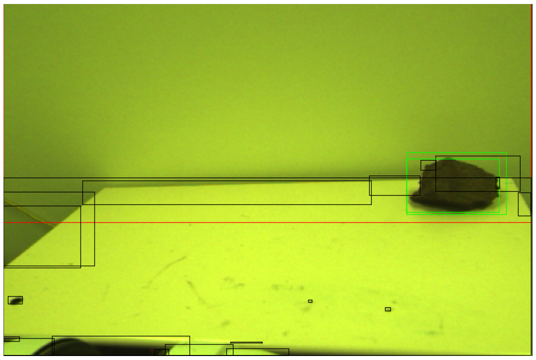


Figure 4: Image obtained after drawing bounding boxes around all the segments.

The scoring scheme can be broken into the following parts:

$$A_{i,j} = F(S_{i,j}, G_i)$$

where $A_{i,j}$ is the area of overlap between obtained segments $S_{i,j}$ and ground truth G_i.

$$R_{i,j} = min(\frac{A_{i,j}}{As_{i,j}}, \frac{A_{i,j}}{Ag_i})$$

where A$s_{i,j}$ is the area of segment $S_{i,j}$ and Ag_i is the ground truth area of G_i.

$$\Theta_i = \sum_{j}^{N_i} R_{i,j} exp(\zeta * R_{i,j})$$

$$\omega_i = \sum_{j}^{N_i} exp(\zeta * R_{i,j})$$

where N_i is the number of segments in image i and ζ is some constant.

$$Score = \frac{\sum_i^M \frac{\Theta_i}{\omega_i}}{M} \tag{3}$$

where M is the number of images in the dataset.

The scoring scheme to some extent can be considered as a weighted average of individual scores where the weights are exponential. The intuition behind exponential weights came from the idea that the segments similar to the groundtruth would have a higher value of R and hence should be given more weight.

After designing the scoring scheme we plotted a graph showing the comparison of scores between the two approaches at various thresholds. As portrayed in Figure 5 our algorithm outperforms the base algorithm at almost every threshold. The scores are relatively smaller at low and high thresholds because of over and under segmenation respectively. This proved our hypothesis that segmentation based only on color is not ideal for object localization.

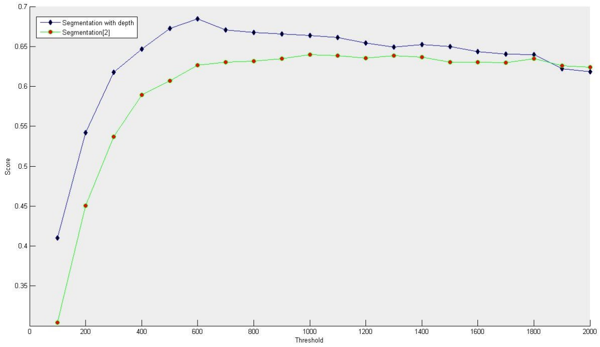


Figure 5: Performance of the two methods.

After observing the graph we decided to change Equation 2 to include the threshold that gives the optimal performance.

$$\tau_o = \arg\max_{\tau \epsilon T} \Phi(Score, \tau)$$

$$||W^T * (F(p_i) - F(p_j))||_2 > \tau_o \tag{4}$$

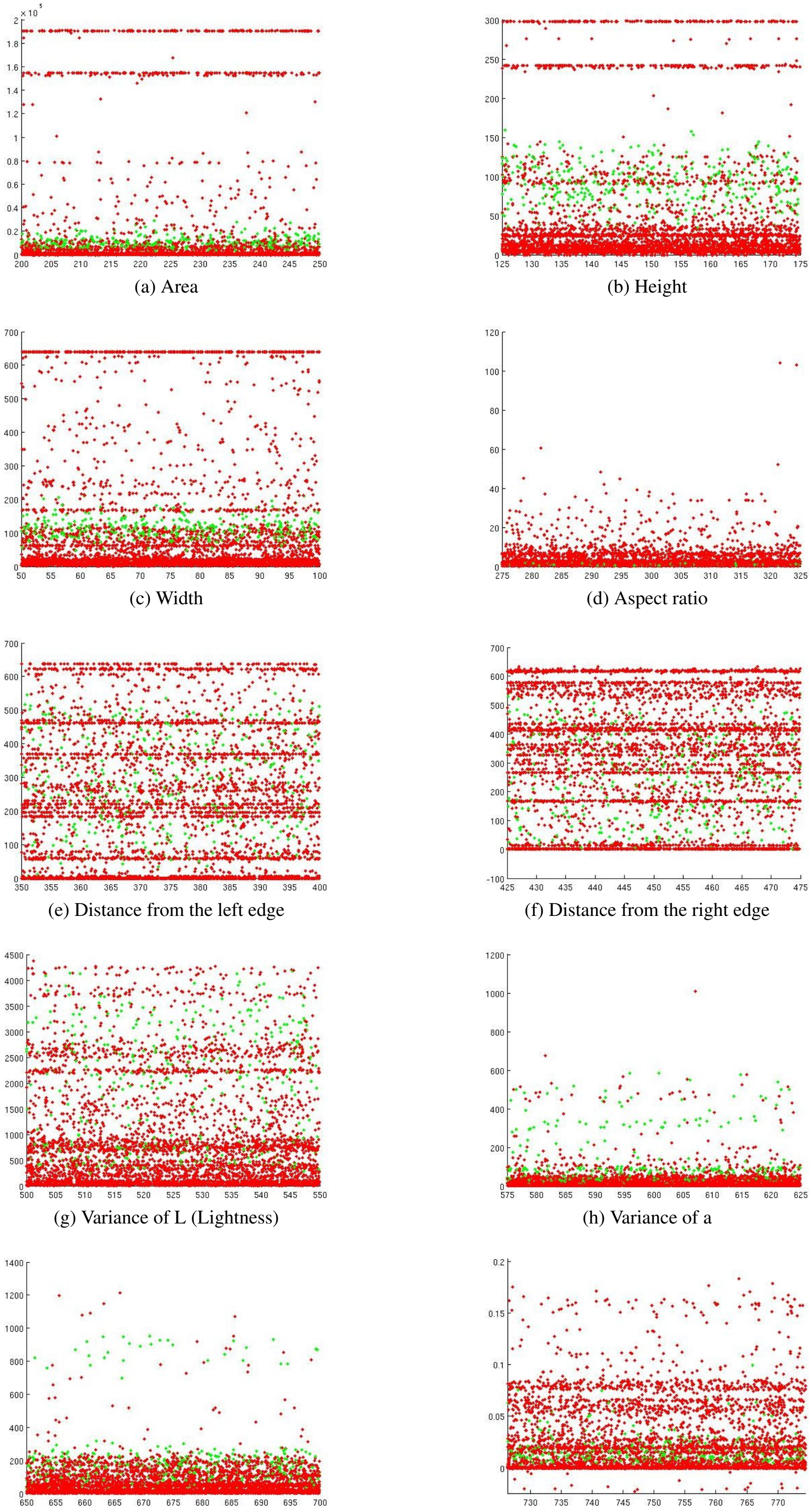
3. Classifier Design and Feature Selection

In this section we briefly touch upon the features that were used by our classifier to localize objects. Before designing the classifier for object localization we needed to label segments of all the images as positive or negative. The bounding boxes of all the segments were assigned positive or negative labels based on their overlap ratio with the groundtruth.

In our proposed framework, we consider features from visual and depth data. A brief description of all the features and the intuition behind choosing them is elucidated in the following sections

3.1. Bag of features

The features were chosen such that they give a true representation of the bounding box. The width, height, area and aspect ratio are a measure of the geometry of the bounding box while distance from left and right edge gives an idea about its position. Historically, researchers have avoided the use of color based features due to computational and philosophical reasons. The computational trade-off is obvious, and the philosophical reason being that humans can perform these task of localization without color information as well. Since, the proposed segmentation framework already uses color data we decided to use some color features as well.



(a) Area

(b) Height

(c) Width

(d) Aspect ratio

(e) Distance from the left edge

(f) Distance from the right edge

(g) Variance of L (Lightness)

(h) Variance of a

(i) Variance of b

(j) Variance of depth

Figure 6: Feature vector plots for all the features.

In order to make the classifer robust to lighting changes we converted the color space from RGB to Lab. Instead of using raw color values or mean as features we decided to go with variance. Variance of L, a and b within the bounding box were chosen to be the color features. To complete the feature vector we needed a depth feature and hence decided to go with variance of depth within the bounding box. The intuition behind chosing variance came from the fact that it is a higher degree feature and probably a better indicator of the state of the object within the bounding box.

3.2. Feature selection

After selecting the features explained in previous section, we trained a SVM model to classify bounding boxes as positive or negative. Surprisingly, the SVM failed to converge. To get a better understanding of the reason we created a feature vector plot for all the features (see Figure 6). There were some interesting insights after studying the feature plots which in turn served as a motivation for our next step. We decided to perform forward search to find the optimal subset of features. As expected, the best feature subset returned by forward search was:

$$\{width, height, area, variance \quad of \quad L, variance \quad of \quad a, variance \quad of \quad b, variance \quad of \quad depth\}$$

It can be seen from Figure 6 that the feature plots of distance from both edges and aspect ratio are randomly distributed and were the main reason for making the data inseparable. Hence, these features were omitted from the feature vector and SVM model was trained using the remaining features. Table 1 shows the rank of different features.

3.3. Grasping

The localization model we proposed takes a new test scene and finds all bounding boxes that have a high probability of containing an object. This is followed by calculating centroids of the bounding boxes in 3D space by mapping them with the point cloud. To test our localization approach we considered the centroid to be best grasping point and tried to grasp the object at that point.

4. Experimental Results

We consider two sets of experiments. The first set of experiments is performed offline to the test the accuracy of the classifier for localizing objects. In the second set we compare our method with [1] when grasping a novel object. We perform all experiments with the Barrett hand having three fingers.

Rank	Features
1	Variance of a
2	Variance of b
3	Variance of depth
4	Area
5	Variance of L
6	Width
7	Height

Table 1: Feature Ranks

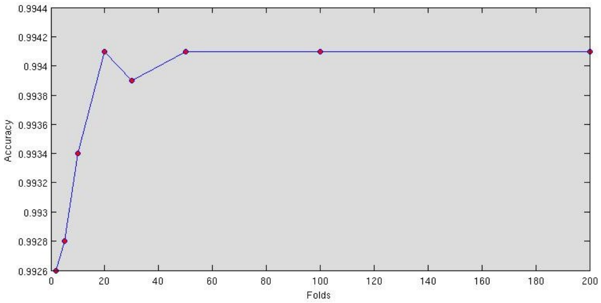


Figure 8: Accuracy versus number of folds of the SVM model for object localization.

4.1. Offline test

Our dataset consists of 200 images and their correspondig depth data. The dataset consists of different objects at varied positions. It is further divided into 353 positive segments and 4216 negative segments. This test is totally offline, i.e. without robotic execution. The goal of the test is to determine the accuracy of the classifier for object localization. We tried to test the accuracy by performing k fold cross validation on the dataset. Figure 8 shows the results of the test. The accuracy increases as we increase the number of folds (i.e. training data) and becomes constant after some time. This shows that the classifier needs only a certain amount of data for training after which the improvemet in accuracy becomes negligible.

4.2. Grasping novel objects

Figure 9 shows the STAIR2 robot grasping a nerf gun. In this section we try to compare our grasping results with [1]. The comparison is not a totally fair because our grasping approach remains static and is independent of the size,shape and orientation of the object. Still, there were some interesting insights for objects that dont necessarily have an ideal grasping point like nerf gun, football, foam and joystick. Surprisingly, our algorithm outperformed the algorithm in [1] for those objects that are non-uniform in their compos-

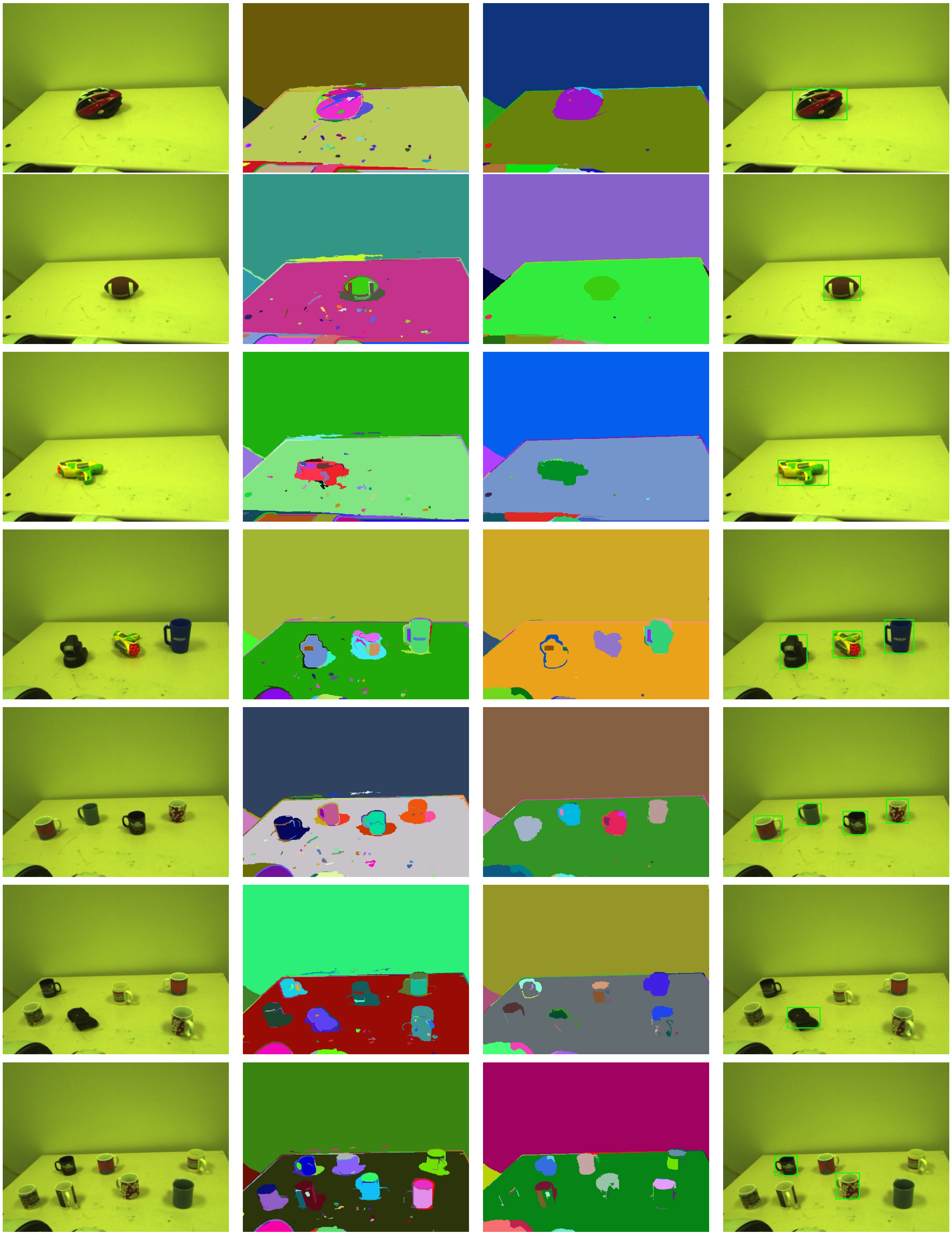


Figure 7: Results.(From left to right) original image, segmentation[2], proposed segmentation, object localization

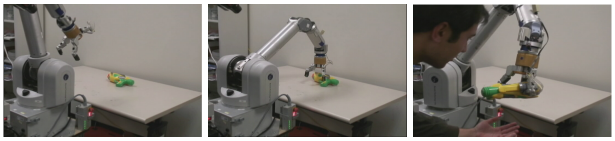


Figure 9: An image sequence in which STAIR2 grasps a nerf gun.

tion and don't necessarily have an ideal grasping point. This builds a foundation for our future work in which we will try to find graspable points based on different objects after localizing them.

5. Acknowledgement

The project has been made under the guidance of Quoc V. Le. It would not have been possible to implement this idea without Quoc's sound mathematical ideas and direction. Siddharth Batra's help with the Stair Vision Library for labelling the dataset while training is also acknowledged.

6. Video

We have posted a video on youtube giving a demonstration of our project.
http://www.youtube.com/watch?v=3Bb2rsC2WDA

We also have a *Stair Fetches A Stapler* section at the end of the video.

References

[1] A. Saxena, J. Driemeyer and A. Y. Ng. Robotic Grasping of Novel Objects Using Vision *International Conference on Robotics Research*, 27(2):157–173, 2008.

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Effecient Graph-Based Image Segmentation *International Journal of Computer Vision*, 59(2):167–181, 2004.

[3] Q. V. Le, D. Kamm, A. F. Kara and A. Y. Ng. Learning to Grasp Objects with Muliple Contact Points

[4] M. Quigley, S. Batra, S. Gould, E. Klingbeil, Q. Le, A. Wellman and A. Y. Ng High Accuracy 3D Sensing for Mobile Manipulators: Improving Object Detection and Door Opening *ICRA*,2009.

[5] S. Gould, O. Russakovsky, I. Goodfellow, P. Baumstrack, A. Y. Ng and D. Koller The STAIR Vision Library (v2.2) *http://ai.stanford.edu/sgould/svl*, 2009.

[6] The ROS (Robot Operating System) framework is an open-source. peer-to-peer, cross-platform message passing system being jointly developed by Stanford University and Willow Garage. ROS is available on Sourceforge. Documentation is avaiable at *http://pr.willowgarage.com/wiki/ROS*.