Image Segmentation and Dimension Analysis

Naman Goyal
2015csb1021@iitrpr.ac.in
Department of Computer Science and Engineering,
Indian Institute of Technology Ropar

Koustav Das
2015csb1017@iitrpr.ac.in
Department of Computer Science and Engineering,
Indian Institute of Technology Ropar

Abstract

Automated dimension analysis from an image is of immense importance in today's world. Current projects present in this domain do not provide a robust and easy to use UI. In the following paper we would like to present a project that creates a robust software that works for most of the pictures while keeping in mind the ease of access. At the present stage this provides all the basic dimension that are important.

1 Introduction

Around the world we find an rapid automation in all sectors of the industry. But still there lies one important industry where most of the work is done manually. Welcome to the packaging industry. Here an user needs to measure the height, width,length and various other dimensions manually. We wanted to tackle this problem with an easy to use software.

Currently there is only a few apps that are currently present in this domain. But there scope in practical sense is extremely limited. Most of them can determine the height of the phone or its distance from an object.

In our present work we propose a working model that lets us take a photo of the desired object with a reference object. The desired object is selected and further the reference objects help us to extract the real life dimensions.

2 Related Work

The major literature is based on methods for segmentation. The supervised techniques to semantically segment learn a mapping from pixel to class. The most of the techniques here are employ neural networks like Fully Convolution Networks, Segmentation networks which employ encoder-decoder networks, Dilated Convolution [4].

There are also unsupervised techniques using super-pixels, k-means and other clustering techniques.

While the original focus was on semantic segmentation of the image. Different supervised methods which models pixel to class mapping were tested such as DeepLab, a state-of-art deep learning system for semantic image segmentation built on top of Caffe. It combines densely-computed deep convolutional neural network (CNN) responses with densely connected conditional random fields (CRF) on MS COCO dataset.

But soon it was realized that semantic segmentation is not fit for task since it can only segment objects on which present with corresponding labels in the training dataset. Hence whenever a new object is given; any pretrained model needs to be trained again and cannot work on the fly.

The focus was then shifted on segmenting the object using initial seeds. While k-means and superpixels segment the whole image into a number of cluster or super-pixels; we required actually to extract the foreground from background.

Finally Lazy Snapping was used an interactive user input based technique based on Graph cut that can be formulated in terms of energy minimization can be approximated by solving a maximum flow problem in a graph. The idea was presented in ACM SIGGRAPH 2004 [2].

3 Method

```
1. Foreground extraction of the desired Object
2. Get the image to pixel ratio
3. Estimation of dimension the perimeter(
    boundary) and bounding box.
```

Listing 1: Workflow

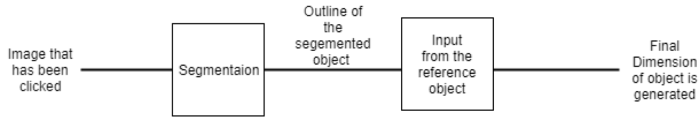

Figure 1: Overall Pipeline

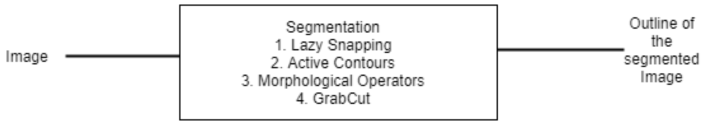

Figure 2: Segmentation Pipeline

3.1 Foreground extraction of the desired Object

There are three main segmentation algorithms that have been deployed in the first segment of the code.

Active contours to separate the image into foreground and the background. The whole algorithm works in an iterative method. So it was extremely essential to properly tune the number of iterations so that foreground and the background is effectively separated without too much loss in the essential information.

After the separation is done we apply morphological operators on the image. The information that are partially present in only the boundaries are removed. Eroding has been done in order to extract only the desired information. Till some extent dilation(hole filling) had also been used to get a proper segmented image. We then extract the boundaries of this foreground object.

Lazy Snapping an interactive image cutout tool. Lazy Snapping separates coarse and fine scale processing, making object specification and detailed adjustment easy. It works on max-flow min-cut algorithm to minimize energy based on graph cut [1].

Was used in *LAB color space* as an region of interest(ROI) based model. This also tries to segment the image into foreground and the background. But in this the user needs to select some portion of what he considers like the foreground and what portion it considers as the background. Based on this selection the code separates out the foreground part of the image. This foreground image is then converted to binary image. In the last part we extract the boundary of the segmented image.

GrabCut is an image segmentation method based on graph cuts. It works both on the ROI model and the foreground and the background selection. [3] The algorithm estimates the color distribution of the target object and that of the background using a Gaussian mixture model. This is used to construct a Markov random field over the pixel labels, with an energy function that prefers connected regions having the same label, and running a graph cut based optimization to infer their values. As this estimate is likely to be more accurate than the original, taken from the bounding box, this two-step procedure is repeated until convergence. Estimates can be further corrected by the user by pointing out misclassified regions and rerunning the optimization. The method also corrects the results to preserve edges.

Morphological operators using sobel edge detector and simple morphological operators to segment the image into foreground and the background. The algorithm detects edge over a certain threshold value. The next we implement dilation and hole filling in on this edge map. This gives us complete object such that there are no holes or void left in the image. Next we apply erosion to smoothen the boundaries of the foreground. On this segmented foreground we extract the boundary.

Figure 3: Shoe Lazy Snapping

3.2 Get the image to pixel ratio

To make implementation generic the user is asked choose the dimension of any object according to their convenience of which they know the real world dimension.

A scaling factor is then computed based on

$$factor = \frac{Actual\ Distance}{Euclidean\ Distance\ in\ image}$$

This factor can be multiplied by the euclidean distance in the image to compute the real world dimension of the objects.

3.3 Estimation of dimension the boundary and bounding box

A close bounding box is fitted around the segmented object. The length and width of is obtained by multiply by scaling factor.

The perimeter of the object that has been segmented is also estimated. A Depth First search is performed from starting a source and each neighbour is added with euclidean distance with previous neighbours; to get cumulative sum of distances between adjacent points.

This data is helpful in for calculating the length and width of the packing that might be required for such an object.

3.4 Results

The results are compared using the methods for active contour, lazy snapping, grab cut, morphological operators.

Results using Lazy Snapping

Image	**Width**	**Height**	**Boundary**	**Time**
Shoe	7.785	12.947 cm	38.931 cm	0.132 sec
Phone 1	16.426 cm	9.230 cm	59.559 cm	5.573 sec
Phone 2	17.069 cm	13.036 cm	56.122 cm	5.606 sec
Mouse	10.073 cm	6.352 cm	35.097 cm	5.592 sec

Results using Active Contour

Image	**Width**	**Height**	**Boundary**	**Time**
Shoe	8.409	14.137 cm	50.36 cm	5.512 sec
Phone 1	15.634 cm	8.626 cm	51.111 cm	54.756 sec
Phone 2	17.306 cm	13.405 cm	51.955 cm	53.003 sec
Mouse	10.001 cm	6.211 cm	33.144 cm	45.268 sec

Results using Morphological

Image	**Width**	**Height**	**Boundary**	**Time**
Shoe	12.039	14.760 cm	47.476 cm	0.027 sec

Results using inbuilt Matlab

Image	**Width**	**Height**	**Boundary**
Pen	15.076	4.830 cm	40.191 sec

3.5 Observations

Observation The lazy snapping is most generic with decent output; and low running time.

Explanation Instant results is made possible by a novel image segmentation algorithm which combines graph cut with pre-computed over-segmentation.

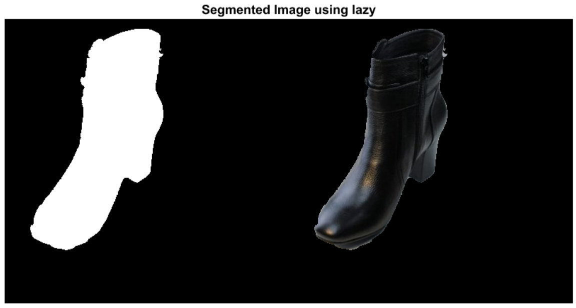

Figure 4: Shoe - Segment - Lazy Snapping

Figure 5: Phone 1 Lazy Snapping

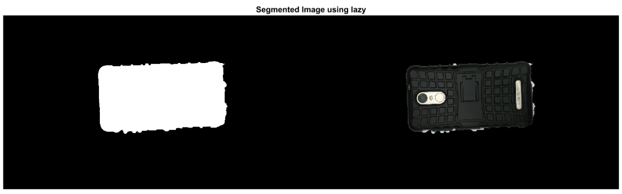

Figure 6: Phone 1 - Segment - Lazy Snapping

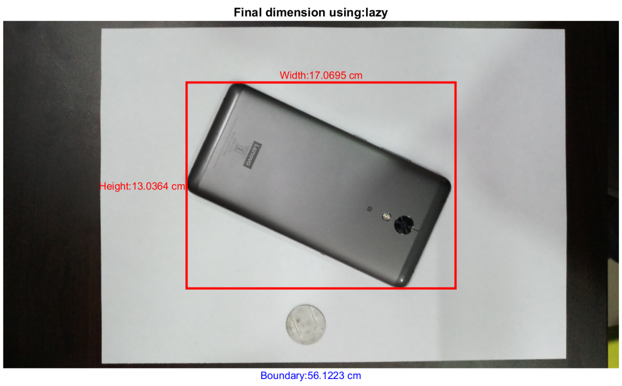

Figure 7: Phone 2 Lazy Snapping

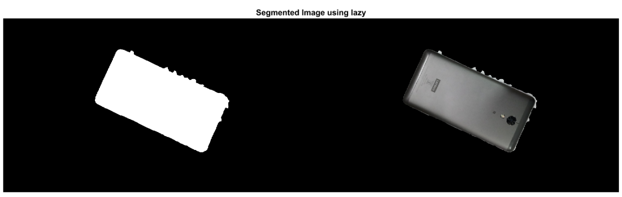

Figure 8: Phone 2 - Segment - Lazy Snapping

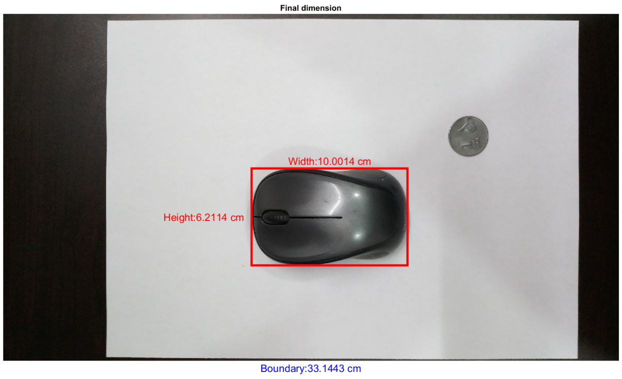

Figure 9: Mouse Active Contour

Figure 10: Phone 1 Active Contour

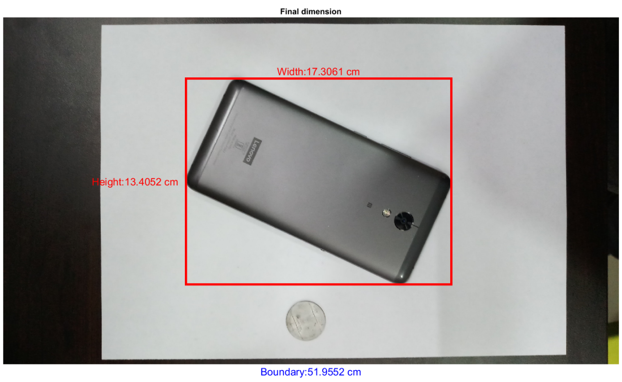

Figure 11: Phone 2 Active Contour

Figure 12: Shoe Morphological

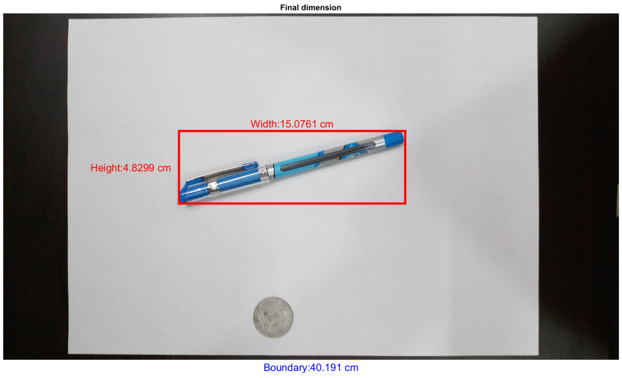

Figure 13: Pen Inbuilt Matlab

Observation Morphological are least generic for segmentation

Explanation Morphological operators have very large number of parameter like size of kernel, kernel itself and hence a single model cannot work for all instances.

Observation Active contour takes highest running time and gives best segmentation

Explanation Active contour has large number of iterations and removes background pixels to maximal extend.

3.6 Future Works

The model has variety of areas for improvement

1. The model currently neglects the depth information; which can lead large error since the scaling factor varies across the depth. To accommodate the same; a depth map needs to obtained using the multi-view geometry using 2 or more images.

2. The reference object can be detected using a deep neural network, cascade object detection or any other method to minimize requirements of inputs by user.

References

[1] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003.

[2] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 303–308. ACM, 2004.

[3] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[4] Martin Thoma. A survey of semantic segmentation. *arXiv preprint arXiv:1602.06541*, 2016.