Crowdsourcing Image Segmentation using SVG

Edward Kim

Lehigh University, Department of Computer Science and Engineering
edk208@lehigh.edu

Xiaolei Huang

Lehigh University, Department of Computer Science and Engineering
xih206@lehigh.edu

Abstract

With the ubiquity of digital cameras, camera phones, web cams, etc., the amount of digital image data is exploding. However, to utilize this data for image annotation and recognition algorithms, a large amount of labeled data for training is required. But obtaining training data on large datasets of images is a very tedious and expensive process. To address this issue, we develop an online image annotation system that can collect annotation data from crowds. Additionally, we incorporate semi-automatic segmentation algorithms that are able to assist the user in creating accurate object boundaries. We show that our system is an effective and useful tool in collecting image annotation data.

Table of Contents

Introduction

In recent years, there has been an explosion of digital image data. However, when interpreted by a computer, this image data is essentially unstructured pixel data. The computer only sees pixel values, and has no knowledge of what is contained in the image. In order to create useful image data to computer recognition and detection algorithms, the content within an image must be labeled. For example, for the task of object recognition, the foreground object in the image must be outlined and separated from the background. With large amounts of training data, a computer algorithm may be able to learn how to separate pixels on a new image into coherent groups. This separation task is referred to as image segmentation, and although, fully automatic methods to perform this image segmentation problem have been proposed, these methods have yet to reach human level accuracy.

One significant hurdle for image segmentation algorithms is the collection of training data. The collection of image segmentation training data is a tedious and expensive task. Assuming that the average segmentation time of an image ranges between 30-60 seconds [KIM2011], a database of 1,000 images could take upwards of 17 hours. Further, new image datasets like ImageNet [DENG2009] contain more than 12 million images, which could take a single user 22.83 years to annotate! In our work, we address the problem of collecting image segmentation training data through the use of scalable vector graphics (SVG) and Amazon Mechanical Turk (Mturk). We build an annotation tool that utilizes the interactivity and flexibility of SVG to collect image annotations from users on the web. Our tool is accessible via Mturk, which is an online crowd sourcing platform where thousands of workers are able to work on provided tasks for a small profit. We show that our system is able to greatly simplify the collection and annotation of objects within the ETHZ Shape dataset [FER2009]. Some sample images from this dataset can be seen in Figure 1.

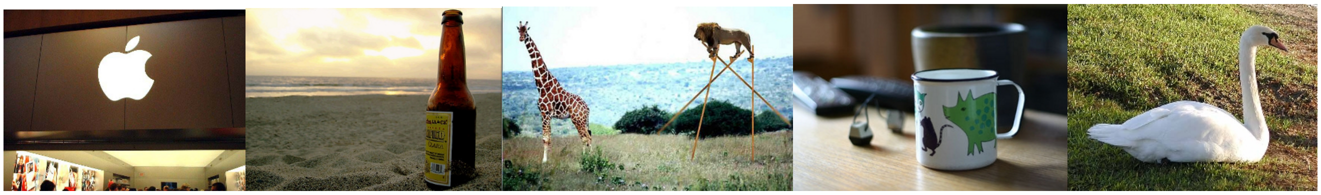


Figure 1. Sample images from the ETHZ Shape dataset.

Related Work

There have been several related works that work towards collecting large sets of annotated training data. The Lotus Hill [YAO2007] database contains a large quantity of manually annotated data. Online sites such as Flickr and Facebook have started allowing the annotation of images via bounding boxes. Probably the most successful and most similar system to ours is the LabelMe [RUS2008] tool. The LableMe system has also experimented with the possibilities of crowd sourcing their annotations on Mturk. However, there are two significant differences that exist between our system and theirs. LabelMe has built their own custom XML image annotation language and annotation system. In contrast, we utilize SVG, a W3C recommendation for our system, thus maximizing compatibility with existing and future browsers. Secondly, LableMe is completely manual, meaning that the user clicks around an object to achieve their final outline. In our system, we incorporate another level of computer assistance. We will describe in later sections that our system has the ability to refine a user's annotation to better fit the object boundary.

Methodology

In our work, we use SVG and Amazon Mechanical Turk, to gather segmentation results from users on the web. We first describe two segmentation methods (a manual method and a semi-automatic method) that we have implemented using SVG. Then, we will describe the implementation of these methods in Mturk.

Segmentation Methods

For our first method, we implemented a manual segmentation in SVG. Given the interactive capabilities of SVG with Javascript, we can allow users to manually draw an outline around objects within images. We display a raster image under an SVG <g> using the <image xlink:href> which catches onclick events. As the user clicks around the image, we can create <circle> elements and <line> elements that follow their clicks. Finally, the user can close the path to create a polygon that surrounds the foreground region by clicking on their first point. The onclick event added to the first point will clear the circle and line elements, and construct a <polygon> element, with the points constructed from the user clicks. Figure 2 illustrates this process.

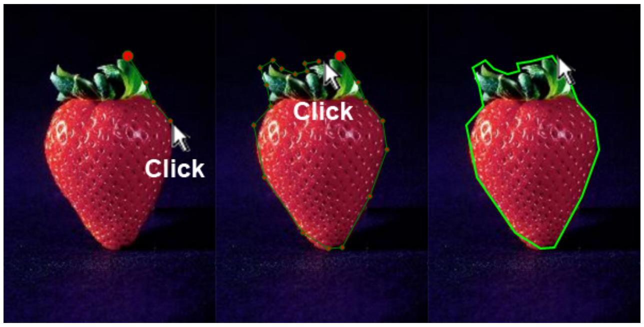
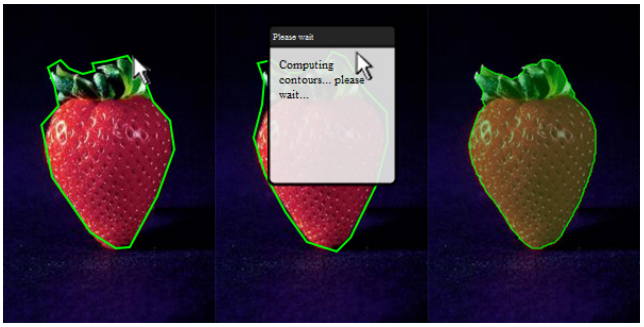


Figure 2. Manual annotation using SVG.

Sample code associated with the final manual segmentation is shown below.

```
<?xml version="1.0" ?>
<svg  baseProfile="full"   xmlns="http://www.w3.org/2000/svg"   xmlns:xlink="http://www.w3.org/1999/xlink"
        width="200" height="300" viewBox="0 0 200 300" version="1.1">
<title> Strawberry example</title>
<desc>Manual annotation strawberry example</desc>
<g xmlns="http://www.w3.org/2000/svg" id="area">
   <image xmlns:xlink="http://www.w3.org/1999/xlink" id="rawimage" xlink:href="strawberry.jpg" x="0" y="0"
                width="200" height="300"/>
   <g id="manual" transform="translate(0,0)">
      <polygon points="123,57 132,76 143,90 158,110 154,148 138,200 115,229 95,230 74,208 47,163 31,113 52,91
                46,74 43,64 49,51 55,57 69,64 86,62 83,54 106,50 " stroke="rgb(0,255,0)" stroke-width="2" fill="none"/>
   </g>
</g>
</svg>
```

Next, we present a semi-automatic segmentation method, where a computer algorithm assists the user in the segmentation process. In Figure 3, we demonstrate the use of SVG to initialize the contour for an active contour segmentation method [KASS1988]. Similar to the manual method, the user clicks around an object they wish to segment from the background. After closing the polygon, an AJAX function will send these points to a server-side script. The server-side script will compute the edges of the image and the corresponding distance transform of the edge information. See [KIM2011] [KASS1988] for specific algorithmic details. The active contour method will evolve the given polygon, and after several computational iterations, the polygon will fit to the image boundaries. The server-side script will then send back a <polygon> object which can be appended to our SVG document to visualize these results.



Please wait

Computing contours... please wait...

Figure 3. The first image is the manual initialization given by the user. The third image is the semi-automatic result displayed using SVG.

Sample code associated with the final semi-automatic segmentation is shown below.

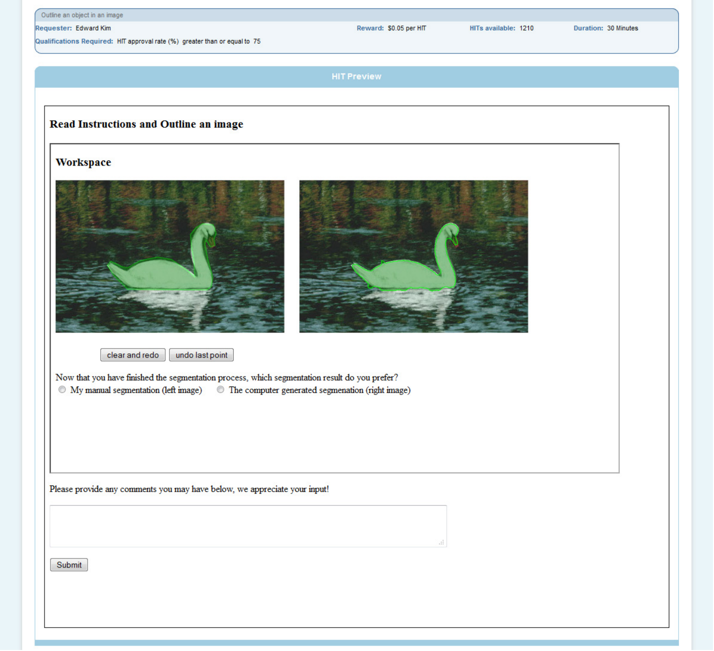
```
<?xml version="1.0" ?>
<svg  baseProfile="full"   xmlns="http://www.w3.org/2000/svg"   xmlns:xlink="http://www.w3.org/1999/xlink"
        width="200" height="300" viewBox="0 0 200 300" version="1.1">
<title> Strawberry example</title>
<desc>Semi-automatic annotation strawberry example</desc>
<g xmlns="http://www.w3.org/2000/svg" id="area">
   <image xmlns:xlink="http://www.w3.org/1999/xlink" id="rawimage" xlink:href="strawberry.jpg" x="0" y="0"
                width="200" height="300"/>
   <g id="manual" transform="translate(0,0)">
      <polygon fill-opacity="0.4" fill="green" stroke="rgb(0,255,0)" stroke-width="1" points="40,129 41,128 41,127
41,126 41,125 41,124 42,123 42,122 42,121 43,120 43,119 43,118 44,117 44,116 45,115 45,114 45,113 46,112 46,111 46,110
46,109 47,108 47,107 48,106 48,105 48,104 49,103 49,102 50,101 51,100 52,99 52,98 53,97 53,96 53,95 53,94 53,93 53,92
53,91 52,90 52,89 52,88 52,87 52,86 52,85 53,84 52,83 52,82 51,81 51,80 50,79 50,78 49,77 48,76 48,75 47,74 46,73 46,72

        ...cut several lines of points...

 59,181 58,180 57,179 56,178 56,177 55,176 54,175 54,174 53,173 52,172 51,171 51,170 50,169 50,168 50,167 49,166 49,165 48,164
48,163 47,162 47,161 46,160 46,159 45,158 45,157 45,156 44,155 44,154 44,153 43,152 43,151 43,150 42,149 42,148 42,147 42,146
42,145 41,144 41,143 41,142 41,141 41,140 40,139 40,138 40,137 40,136 40,135 40,134 40,133 40,132 40,131 40,130 40,129"/>
   </g>
</g>
</svg>
```

Amazon Mechanical Turk

To effectively and efficiently collect human level segmentations, we can use an online crowdsourcing tool, Amazon Mechanical Turk. This online platform allows a researcher to create Human Intelligence Tasks (HITs), where thousands of workers can participate. Given our SVG representation, we are uniquely positioned to take advantage of the online environment for our segmentation task. We create segmentation HITs where we ask a user to outline a specific object within an image. Our system uses the manual outline as an initialization to our semi-automatic method and presents the user with both results i.e. their manual annotation and the computer assisted annotation. Next, we ask the user if the semi-automatic method outperforms their manual annotation. Once the user completes the work, we collect and store both their manual polygon representation and our computer generated polygon in a MySQL database and pay the worker a small sum of money (0.05 dollars). A sample interface of our HIT can be seen in Figure 4.



Outline an object in an image

Requester: Edward Kim | Reward: $0.05 per HIT | HITs available: 1210 | Duration: 30 Minutes

Qualifications Required: HIT approval rate (%) greater than or equal to 75

Read Instructions and Outline an image

Workspace

Now that you have finished the segmentation process, which segmentation result do you prefer?

○ My manual segmentation (left image) ○ The computer generated segmenation (right image)

Please provide any comments you may have below, we appreciate your input!

Figure 4. Partial screen shot of our Amazon Mechanical Turk interface.

Results

For our experiments, we collected image annotation results for a standard dataset, the ETHZ Shape dataset. This set contains 255 (we use 242) images from five different classes (apple logos, bottles, giraffes, cups, and swans). For redundancy, we published 5 HITs per image for a total of 1,210 HITs on Mturk. Within 3 hours and 11 minutes of publishing our HITs, all 1,210 HITs were completed by 71 unique workers. The average time spent per HIT was 1 minute 39 seconds, at a cost of 0.05 dollars per HIT. The total cost for the annotations, plus processing fees was $66.55.

We were also interested in whether or not our semi-automatic segmentation algorithm would be helpful in the segmentation task. For each HIT, the user was asked whether or not they preferred their manual segmentation or our semi-automatic result. In 93 (out of 242) images, at least one user believed that our computer assisted segmentation outperformed their manual segmentation. This demonstrates that our semi-automatic method is a useful and helpful addition for collecting ground truth segmentation data.

Worker accuracy

To ensure the accuracy of the worker's segmentations, we issued five HITs per image. We could reject or accept a segmentation based upon the agreement between users. Additionally, we take the users history into account. Generally speaking, if the user has performed the task correctly and accurately the first several times, we accept their subsequent submissions. In Figure 5, we present some of the collected results with their segmentations overlaid on top of the image.

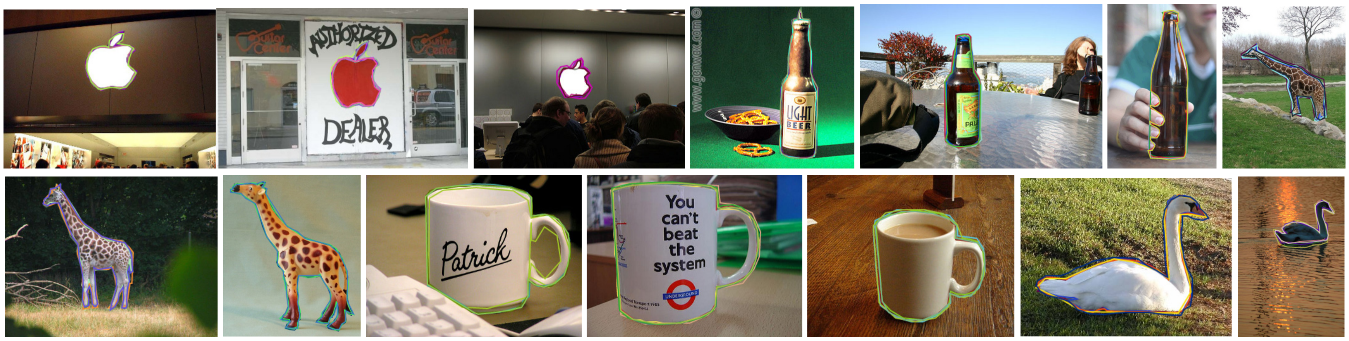


Figure 5. Mturk user segementations. Multiple user provided polygons are overlaid on top of the image.

Conclusion

In conclusion, by using the capabilities of SVG interaction and visualization, we can facilitate the process of image segmentation data collection. Further, the online support of SVG provides a stable environment, even when dealing with complex algorithms. Ultimately, we showed that using SVG for online data collection is both feasible and effective.

Bibliography

[KIM2011] E. Kim. X. Huang. *Markup SVG - An Online Content-Aware Image Abstraction and Annotation Tool*. IEEE Transactions on Multimedia. Volume 13, Issue 5, 2011.

[DENG2009] J. Deng. W. Dong. R. Socher. L.J. Li. K. Li. L. Fei-Fei. *ImageNet: A large-scale hierarchical image database*. IEEE Conference on Computer Vision and Pattern Recognition. pp. 245-255. 2009.

[FER2009] V. Ferrari. F. Jurie. C. Schmid. *From Images to Shape Models for Object Detection*. International Journal of Computer Vision, IJCV 2009.

[YAO2007] Z.Y. Yao. X. Yang. S.C. Zhu. *Introduction to a Large Scale General Purpose Groundtruth Dataset Methodology, Annotation Tool, And Benchmarks*. 6th Int'l Conference on EMMCVPR, 2007.

[RUS2008] B.C. Russell. A. Torralba. K.P. Murphy. W.T. Freeman. *LabelMe: a database and web-based tool for image annotation*. International Journal of Computer Vision, IJCV 2008.

[KASS1988] M. Kass. A. Witkin. D. Terzopoulos. *Snakes: Active contour models*. International Journal of Computer Vision, IJCV 1988.