Predicting Sufficient Annotation Strength for Interactive Foreground Segmentation

Suyog Dutt Jain Kristen Grauman

Abstract

The mode of manual annotation used in an interactive segmentation algorithm affects both its accuracy and ease-of-use. For example, bounding boxes are fast to supply, yet may be too coarse to get good results on difficult images; freehand outlines are slower to supply and more specific, yet they may be overkill for simple images. Whereas existing methods assume a fixed form of input no matter the image, we propose to predict the tradeoff between accuracy and effort. Our approach learns whether a graph cuts segmentation will succeed if initialized with a given annotation mode. Whether given a single image that should be segmented as quickly as possible, or a batch of images with fixed annotation budget, we show how to use these predictions to select the easiest modality that will be sufficiently strong to yield high quality segmentations.[1]

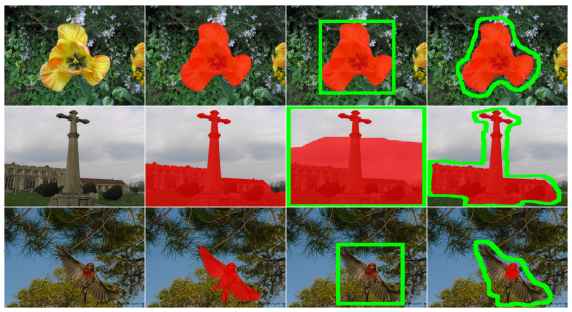
1. Introduction

Foreground segmentation is a fundamental vision problem with an array of applications. These include helping users perform precise visual search, training object recognition system, rotoscoping etc. In any such scenario, it is natural for humans to help annotate the foreground.

Research on *interactive segmentation* considers how a human can work in concert with a segmentation algorithm to efficiently identify the foreground region. Typically, the human gives high-level guidance—in the form of coarse spatial annotations, which the algorithm uses to learn foreground/background models and then refine the input down to the pixel level segmentation.

Existing methods (e.g. [3]) assume the user always gives input in a particular form (e.g., a bounding box or a scribble), and the focus is on using that input most effectively. However this leads to a suboptimal tradeoff in human and machine effort. The problem is that each mode of input requires a different degree of annotator effort. At the same time, depending on its content, an image may be better served by one form of input or another (Fig. 1 shows some examples).

In this work, we propose to learn the image properties that indicate how successful a given form of user input will be, once handed to an interactive segmentation algorithm. This enables us to develop an image annotation tool which utilizes human effort in an optimal manner, by carefully selecting which input modality is sufficiently strong and requires least effort for a given image.



(a) Image (b) Ground Truth (c) Bounding Box (d) Sloppy Contour

Figure 1: Interactive segmentation results (shown in red) for three images using various annotation strengths (marked in green). Note how the most effective mode of input depends on the image content. Our method predicts the easiest modality that will be sufficiently strong to successfully segment a given image.

Various recent methods attempt to reduce human labeling effort, for example by selecting the most useful frames for video segmentation [6, 5] or asking a human to click on informative object parts [7]. Whereas prior work predicts *which images should be annotated* (and possibly where) to minimize uncertainty, we predict *what strength of annotation will be sufficient* for interactive segmentation to succeed. Furthermore, whereas most existing methods assume a back-and-forth with the annotator, we take a "one-shot" approach that makes all requests simultaneously, a potential advantage for crowdsourcing or mobile interfaces.

2. Approach

In interactive segmentation, the user indicates the foreground with some mode of input. No matter the annotation mode, we use the pixels inside and outside the user-marked boundary to initialize the foreground and background models, respectively. Using these appearance models, we then define our segmentation model as the standard MRF based energy function with unary and pairwise terms. We use graph cuts [1] to minimize this energy, and use the GrabCut idea to iteratively refine the segmentation [3]. Our approach chooses from three annotation modalities:

(1) **Bounding box:** Tight rectangle around the foreground objects (fastest).
(2) **Sloppy contour:** Rough contour surrounding the foreground; provides tighter object boundary (intermediate).

[1] This work appeared in ICCV 2013 [2].

(3) **Tight polygon:** Tight polygon along the foreground boundaries, equivalent to perfect segmentation (slowest).

Given a training set for which the true foreground is known, we first simulate the human input for each training image by fitting a tight rectangle around the true foreground mask (for bounding box) and by dilating the true mask by 20 pixels (for sloppy contour). After applying graph cuts segmentation with these simulated inputs, we obtain a foreground estimate for each modality.

Using these foreground estimates, we compute features which capture the degree of separation between foreground and background regions, which directly affects the performance of graph cuts segmentation. Our features capture the color dissimilarity between foreground and background regions, uncertainty in the graph cuts solution, foreground complexity, and how well the output segmentation aligns to strong image boundaries. We also use these foreground estimates to categorize every training image as "easy" or "hard" for a particular modality based on its overlap score with the ground truth mask (See [2] for details).

We train separate discriminative classifiers (for bounding box and sloppy contour) that take an image as input, and predicts whether a given annotation modality will be successful in segmenting it. Given a novel image at test time, we apply a saliency detector to coarsely estimate the foreground. Using that estimate, we extract the separability features as described above, and apply the difficulty classifiers to predict the relative success of each modality.

Having predicted the relative success of each modality, we can explicitly reason about the tradeoff in user effort and segmentation quality. We propose two ways to determine the appropriate annotation choice. In the first, we take a single image as input, and ask the human user to provide *the easiest (fastest) form of input that the system expects to be sufficiently strong*. In the second, we consider a batch of images and a fixed annotation budget. Using the classifier confidence scores, we design an optimization strategy to select the optimal annotation tool *for each image* that will maximize total predicted accuracy for the entire batch, subject to the constraint that annotation cost must not exceed the budget. We assume a fixed cost per modality for each image: Bounding box (7s), Sloppy contour (20s), Tight polygon (54s). These estimates come from the average time taken by the 101 users in a Mechanical Turk user study.

3. Results

We evaluate on three public datasets (IIS, MSRC and iCoseg) that provide pixel-level label and compare with the following baselines: **(1) Otsu thresholding:** a classic adaptive thresholding technique. **(2) Effort Prediction [4]:** state-of-the-art method for estimating image difficulty. **(3) Global Features:** difficulty predictors trained using global image features. **(4) GT-Input**: upper bound, uses ground truth masks instead of saliency based estimate at test time. **(5) Random:** random confidence value for each modality.

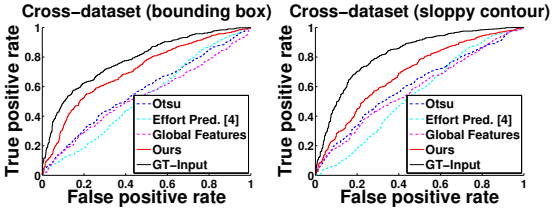


Figure 2: Difficulty prediction accuracy for cross-dataset experiments.

Figure 3: Example successful predictions per annotation modality.

Predicting difficulty per modality First we see how well all methods predict the success of each annotation modality. Fig. 2 shows the comparison of our method with other baselines in a leave-one-dataset-out experimental setup. Our approach consistently performs well for both input modalities. The high performance shows that our method is learning which generic cues indicate if a modality will succeed—not some idiosyncrasies of the particular datasets. Fig. 3 shows some example predictions.

Annotation choices to meet a budget Next we evaluate our idea for optimizing requests to meet a budget, with a Mechanical Turk user study. The budget values range from the minimum (bounding boxes for all images) to the maximum (tight polygons for all images). For each budget value we use our optimization strategy to make a collective decision for the entire set of images.

We then present users with the necessary tools to do each modality, and time them as they work on each image. We feed the boxes/contours annotated by the users to the graph cuts engine and compute the overlap score with ground truth mask. Fig. 4 plots the user timing information against the final segmentation accuracy. Our method consistently selects the modalities that best use annotation resources: at almost every budget point, we achieve the highest accuracy.

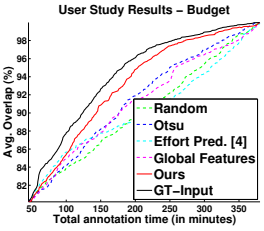


Figure 4: Annotation choices with a budget.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, Nov. 2001. 1

[2] S. D. Jain and K. Grauman. Predicting sufficient annotation strength for interactive foreground segmentation. In *ICCV*, December 2013. 1, 2

[3] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 1

[4] S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, 2009. 2

[5] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *ECCV*, 2012. 1

[6] C. Vondrick, D. Ramanan, and D. Patterson. Efficiently scaling up video annotation with crowdsourced marketplaces. In *ECCV*, 2010. 1

[7] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011. 1