

# FreeLabel: A Publicly Available Annotation Tool based on Freehand Traces

Philippe A. Dias<sup>1</sup>

Zhou Shen<sup>1</sup>

Amy Tabb<sup>2</sup>

Henry Medeiros<sup>1</sup>

<sup>1</sup>Marquette University (EECE), USA

<sup>2</sup>U.S. Department of Agriculture (USDA)

{philipe.ambroziodias, zhou.shen, henry.medeiros}@marquette.edu

amy.tabb@ars.usda.gov

## Abstract

*Large-scale annotation of image segmentation datasets is often prohibitively expensive, as it usually requires a huge number of worker hours to obtain high-quality results. Abundant and reliable data has been, however, crucial for the advances on image understanding tasks achieved by deep learning models. In this paper, we introduce FreeLabel, an intuitive open-source web interface that allows users to obtain high-quality segmentation masks with just a few freehand scribbles, in a matter of seconds. The efficacy of FreeLabel is quantitatively demonstrated by experimental results on the PASCAL dataset as well as on a dataset from the agricultural domain. Designed to benefit the computer vision community, FreeLabel can be used for both crowd-sourced or private annotation and has a modular structure that can be easily adapted for any image dataset.*

## 1. Introduction

The rapid rise in popularity of deep learning models in computer vision has brought a corresponding demand for labeled data. Depending on the image understanding task, the required annotations may range from tags at the image level (image classification), to bounding boxes (object detection) or pixel-level annotations (image segmentation).

Varied and high-quality image annotations are crucial for both training and evaluation of models that are accurate and robust. Currently, most of the Convolutional Neural Networks (CNNs) models successful at image understanding tasks [12, 29, 33] are pre-trained on the ImageNet [20] and COCO [35] datasets, due to their large variability.

Manual labeling of large datasets is challenging and time-consuming. The costs reported for the COCO dataset in [35] illustrate these difficulties. Containing over 2.5 million object instances, its labeling using Amazon’s Mechanical Turk (AMT) required: 20k worker hours for category labeling at image-level; 10k hours for instance spotting; and, staggering, 55k hours for instance segmentation.

To meet the need for large, labeled datasets, several approaches have been proposed. Different types of crowd-

sourcing have been used to generate labeled data quickly, from commercially available solutions such as the AMT, annotation parties [26], volunteer/citizen science initiatives [28], and custom-built pipelines [14].

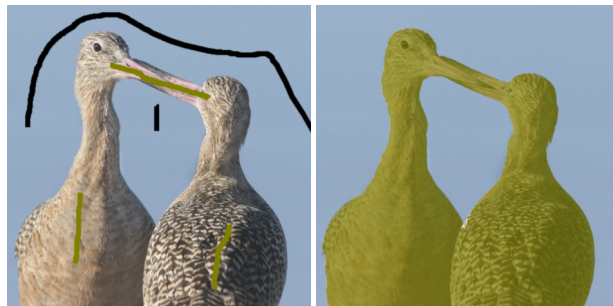


Figure 1. This paper describes an annotation tool that generates high-quality segmentation masks using simple freehand traces as input. From the few user traces illustrated in the left image, our FreeLabel tool outputs the object segmentation indicated by the yellow overlay in the right image.

Rather than selecting individual pixels, a popular strategy consists of approximating segmentations as polygons, which can be problematic for objects with complex boundary structures. Other strategies focus on labeling pre-segmented regions, such as superpixels [9, 45]. Although these strategies accelerate the annotation process, the segmentation quality is at risk in scenarios where the pre-computed regions fail to properly attach to boundaries.

To minimize the needs for finely-annotated training data, the development of weakly-supervised training methods is also a very active field of research. Strategies for propagation of sparse annotation include graph cuts [40], level sets [51] and graphical models [34]. As the leaderboard of the PASCAL VOC 2012 dataset<sup>1</sup> shows, the performance of models trained in this way is still noticeably worse than models trained with fully annotated masks.

We combine ideas from both the existing annotation tools and the field of semi-supervised learning to facilitate and minimize the amount of user interactions for annotating image segmentation masks, ultimately reducing label-

<sup>1</sup><http://host.robots.ox.ac.uk:8080/leaderboard/>

ing costs. Our contribution consists of a web-based tool, named FreeLabel, which allows the user to trace lines or “freehand” scribbles of different thicknesses for the different categories present on an image. These scribbles are propagated to the remaining unlabeled pixels using the Region Growing Refinement (RGR) algorithm introduced in [21] for semantic segmentation refinement. Compared to other algorithms, RGR has the advantages of being fully unsupervised (thus category agnostic), simple to implement, with computational time and parameterization that allow quick and simple user interactions.

We assess the applicability of our tool in two contexts: the first is general object segmentation, exemplified by the PASCAL VOC dataset that has pixel-accurate labels for multiple different categories; and the second is the annotation of images of fruit tree flowers, which has applications for precision agriculture [22] [24]. In the first context, we analyze how long it takes for users to become familiar with our tool, and also the average annotation time and the segmentation quality they obtain in comparison with the official PASCAL ground-truth.

The first context serves as a training for the second, where images of flowers of multiple fruit tree species are annotated [22]. In this scenario, we evaluate how well users can annotate images for which no ground-truth is available, and thus no intermediate feedback is provided.

Our contributions to the state-of-the-art are:

1. a web-based tool, FreeLabel, for interactive annotation that is shown to be intuitive and effective, with users obtaining high-quality segmentations in an average time of 60 seconds per object for the PASCAL dataset;
2. FreeLabel can be easily configured for any object category or dataset, an advantage inherited from the underlying unsupervised growing algorithm and the modular implementation of the tool;
3. public release of the tool together with the camera-ready version of this paper;
4. the web-based structure of FreeLabel allows crowdsourcing and, when data privacy is of concern, private annotation using a local deployment.

## 2. Related work

### 2.1. Segmentation datasets and labeling tools

Introduced in 2005, the PASCAL VOC dataset [25] is the most widely-used dataset for visual object segmentation. Images within its 2012 *valtrain* set contain a total of 6929 segmented objects, distributed within 20 different semantic categories. As reported in [26], the process of annotating the images with pixel-level accuracy was extremely time-consuming, even though a 5-pixel wide tolerance margin was allowed around each object.

The ImageNet dataset [51] with its 15 million labeled images was crucial for the development of deep CNNs that revolutionized the state-of-the-art in image classification. Inspired by such success, the COCO dataset [35] was introduced in 2015 to foster advances in object recognition, localization and segmentation. It comprises 2.5 million objects instances in 328k images, labeled by AMT workers using an adapted version of the OpenSurfaces interface [5].

The OpenSurfaces interface resembles the LabelMe web-based annotation tool [41], which was introduced in 2008 and is still widely used for segmentation annotation. Users provide object segmentations by tracing polygons along its boundary, typing the object name after completing the polygon. However, as mentioned in both [41] and [35], quality control is an important concern with this scheme. High-quality segmentations of objects with complex boundary structures require large numbers of vertices, leading to a trade-off between quality versus time spent to label each object. For annotation of the COCO dataset, its authors opted to minimize costs by collecting only one annotation for each instance, which required on average 79 seconds per object. Yet, despite efforts such as quality verification steps, the dataset still contains some segmentation masks that poorly attach to the object boundaries [21].

The Cityscapes dataset for semantic urban scene understanding [17] was also annotated using layered polygons. To ensure that rich and high-quality pixel-level segmentation masks were obtained, its corresponding 5k images were annotated in-house. Over 1.5h were required on average for annotation and quality control of each image with a restricted pool of high-quality annotators.

Alternative labeling strategies exploit superpixels to facilitate the annotation process. The interface used for labeling the COCO-Stuff dataset [9] combines SLICO superpixels [2] with a size-adjustable paintbrush tool that enables labeling of large regions at once. As mentioned by Tang et al. in [45], superpixel errors can lead to significant annotation errors with this kind of interface. To minimize these artifacts, the authors described in [45] a interface that performs morphology-based boundary smoothing and allows the annotator to select the desired superpixel size to improve boundary adherence. Yet, this increases the complexity of the task, as the user has to try different configurations and label each superpixel individually.

### 2.2. Weak and unsupervised segmentation

**Graph cuts.** Energy minimization approaches using the graph cuts paradigm are suited to interactive segmentation in that hard constraints are specified via squiggles for background and foreground classes [6], [7], [27]. The popular GrabCut algorithm [40] improved over interactive tools such as Intelligent Scissors (Magnetic Lasso) [38], relaxing some of the labeling burden on the user. The

user selects a bounding box of background pixels and can further edit the generated segmentation by drawing firm background/foreground traces. Gaussian Mixture Models (GMMs) are used for color modeling and a Gibbs energy is iteratively minimized using minimum cut.

**Level sets.** The level set approach has been used in segmentation since the 1990s, and can also be formulated as an energy minimization problem. Given an initialization, a boundary is evolved in the direction of a local minimum found via front propagation by solving partial differential equations [19]. An issue with level set implementations in the 2000s was runtime, and interactive approaches focused on reducing runtime using GPU implementation [10] [18]. One approach allowed user input to adjust model parameters, in [10], while [18] reformulated energy functionals to incorporate user input. In [36], bounding-box initialization and the level set formalism were used for interactive segmentation. The TouchCut [51] interface exploits level-sets to grow segmentation masks from single points, which is effective when foreground and background colors are significantly different.

**Propagation by pixel-affinity.** In a similar fashion that superpixel algorithms segment input images into clusters [43], several matting and segmentation algorithms use low-level information such as texture, color affinity and spatial proximity to classify unlabeled regions based on sparse annotations [16, 13, 52]. Similar methods have been used to refine segmentation masks predicted by CNNs [32, 11, 12], as CNNs successfully exploit high-level context for semantic classification but fail to generate predictions with proper adherence to object boundaries. One such method is the *Region Growing Refinement* (RGR) [21], which combines Monte Carlo sampling of high-confidence samples with a region growing algorithm that is guided by spatial and color proximity between neighboring pixels. Selected as building-block for FreeLabel, we described more details of RGR in Section 3.

**Joint propagation and CNN training.** Recent approaches aiming at interactive or weakly-supervised semantic segmentation focus on architectures in which the propagation of sparse annotations and the optimization of network parameters are performed jointly. Different works combine Fully Convolutional Networks (FCNs) with: GrabCut [39]; superpixels and graphical modeling [31, 34]; novel loss functions and training strategies for weakly-supervised and interactive learning [31, 44, 37]. In [3], the idea of Laplacian matting matrices is combined with superpixels and a Deeplab-ResNet [12] to identify layers (soft segments) that are semantically meaningful. For annotation of video sequences, in [15] a FCN is used to map input pixels onto an embedded space where pixels belonging to the same instance are close together, followed by a nearest-neighbor approach that classifies pixels based on

reference masks provided at the first frame and on sparse user inputs.

### 2.3. Good practices for design of annotation tools

Vondrick et al. in [50] provide a set of best practices for crowdsourced video annotation, based on a three-year large scale study costing thousands of dollars for image annotation. A critical observation is that annotating platforms must aim at minimizing the cognitive load of the user. As backed by psychology studies [42, 4], minimizing interruptions and choices help to reduce user anxiety and increase efficiency. Moreover, they observed that providing motivational feedback increases the workers' confidence that their work will not be rejected, which encourages workers to continue annotating.

Games With A Purpose (GWAP) exploit the idea that adding game-like elements to interfaces additionally motivates users to perform tasks of interest. The ESP Game [46] for image labeling is a widely known example: an image is shown to two players (users) and, without external communication, both enter possible words until a word is agreed upon. The common word becomes a label for the image. Other examples are the Peekaboom game for object localization [49], Verbosity to collect commonsense facts about words [48], and Phylo for multiple sequence analysis [30].

Users play for the desire of being entertained, rather than for money or altruism [47]. Timed response, score keeping, and randomness are important features for designing challenging and hence enjoyable games [47], as players are driven to play by the desire of increasing their skill level or to score higher than others. Compared to subjective and verbal instructions, scores are a more intuitive form of feedback to the user as they combine multiple aspects into a single performance metric.

## 3. Method

Our object is to develop a web-based labeling interface that: i) is intuitive to use; ii) allows users to quickly provide high-quality annotations; iii) can be easily adapted for different datasets and categories.

As observed in Section 2.3, a good user interface should minimize the cognitive load on the user. Thus, instead of using propagation techniques that require supervised training or manual tuning of different sets of parameters, our tool exploits the RGR algorithm for unsupervised region growing. Based on related works, limitations of current tools and previous experiences with image annotation, we opted for designing a tool where the user input consists in simply drawing scribbles (freehand traces) or straight line segments on the images.

By keeping all the parameterizations of the RGR algorithm fixed, we avoid any non-intuitive burden on the users.



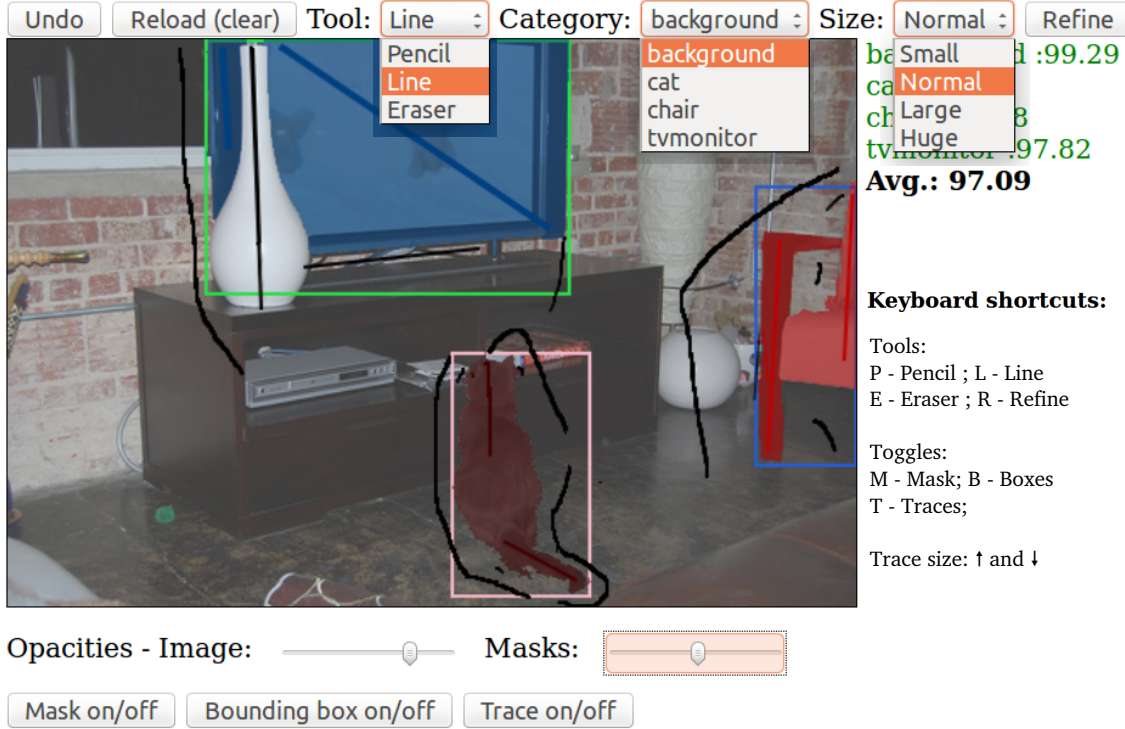


Figure 2. Diagram summarizing how the different modules of our tool, FreeLabel, interact with each other. Users can draw with a freehand pencil, or line segments. An eraser allows undoing small errors. Dialog boxes allow the user to select the object categories associated to the current trace, as well as adjust tool sizes. Other options, to help with visibility while labeling, such as opacity and masks, are available via slider bars.

The quality of the segmentation provided by RGR is proportional to the amount and quality of initial seeds available. In this way, the user interaction to guide the growing process becomes quite intuitive, with simple guidelines: traces are grown based on color similarity and must be provided within the boundaries of the corresponding objects; thicker traces act as enforcement for the growing algorithm, since more seeds are available than for thin traces; if any region is incorrectly labeled by RGR, the user can easily correct it by adding a new trace of the correct category.

In addition to its simple formulation, we found the RGR implementation to be very suitable for multi-class segmentation annotation. Its growing process is class agnostic, propagating initial seeds into clusters regardless of seed label. This is advantageous in terms of running time, as the growing process has the same computational complexity regardless of the number of classes present in the image (average runtime lower than 1 second for PASCAL images [21]). After clusters are formed for each set of seeds, they are classified into semantic categories by means of simple majority voting. Figure 3 shows an example of this process, where each cluster is assigned to the class for which it contains most labeled pixels.

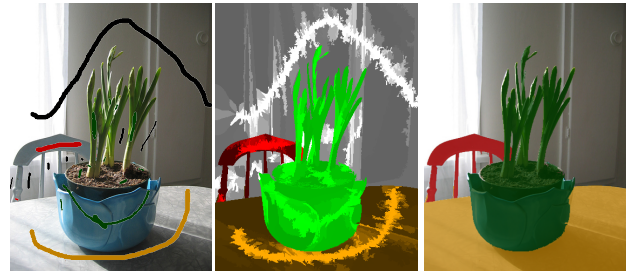





Figure 3. Illustration of how traces are propagated to neighboring pixels. *Left*: input traces drawn by the user. *Center*: the brightness (intensity) of the color in each pixel is proportional to the score computed for its most likely category. For better visualization, *background* traces are shown in black, while the *background* likelihood is in grayscale from black (lowest) to white (highest). *Right*: final segmentation obtained using maximum category likelihood per-pixel, with transparent *background*.

### 3.1. FreeLabel Functionality

Figure 2 shows a screenshot illustrating the functionality of our interface, together with an example of high-quality segmentation masks obtained from only a few user interactions. Three tools are available for drawing and adjusting traces using the mouse:

- *Pencil* : used for quickly tracing freehand scribbles. Once the user holds down the mouse’s left-button, traces corresponding to the mouse trajectory are drawn. It is especially useful for regions that do not require high precision;
- *Line* : traces straight lines connecting the point where the user clicked the mouse button to the point where it was released. It is especially helpful for straight and thin structures, such as chairs’ legs and animals’ limbs.
- *Eraser* : used to correct imprecisions in provided scribbles, such as small portions protruding outside the corresponding object’s boundary.

Each tool can be configured with four different thicknesses: *small* (1px thick), *normal* (2px), *large* (4px) or *huge* (8px). After tracing scribbles over the image, the user can invoke the RGR algorithm by simply clicking the *Refine* button, which automatically grows segmentation masks from the provided traces. To annotate smaller objects, the user can zoom in/out using the mouse scroll, as in any modern web-browser. Finally, keyboard shortcuts are available for all the commands to facilitate the annotation process.

In addition to intuitive commands, visualization is another key factor that impacts the labeling experience and final annotation quality. Similar to the PASCAL, COCO and other datasets, a specific color is associated to the traces and masks of each category. For the background, traces are shown in black, while the masks are invisible. However, there are scenarios where the image is too dark or contain colors with poor contrast to traces and/or masks. To handle these cases, our interface allows the user to control the brightness (opacity) of both image and segmentation masks using the sliders under the canvas. Moreover, masks and traces can be hidden/shown with the click of the corresponding toggle buttons.

### 3.2. Implementation

Our FreeLabel tool for segmentation annotation relies on three main building blocks: a graphical user interface (GUI), the Django framework, and the RGR algorithm. Figure 4 summarizes the relationships.

An important criterion for our design choices concerned how easy the user’s inputs and the RGR algorithm could be combined for the computation of segmentation masks. As a starting point, we had access to the original RGR implementation in MATLAB. Aiming at a open-source, platform-independent web interface, we adapted RGR’s implementation to Python and opted for the Django platform as the web framework.

Django [1] is a free, open source Python framework that follows the Model-View-Template architectural pattern. The *Model* layer allows users to access database information without requiring any knowledge of the intricacies

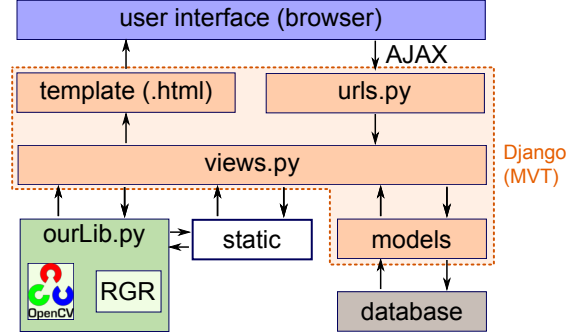


Figure 4. Diagram summarizing how the different modules of FreeLabel interact with each other.

of database rules. The *View* logic layer of Django contains functions that handle the communication between the *Model* and the *Templates*, which correspond to the exhibition layers that defines what is shown to users through the browser.

Using Figure 4 as guidance, a top-down walk-through of our tool’s implementation starts with the graphical interface displayed by the web browser to the user. The design and functionality described in Section 3.1 and exemplified in Figure 2 are implemented as customized Django templates, using HTML/Javascript. For actions requiring the execution of Python commands, the template (.html) file will trigger an AJAX call that is mapped to a corresponding function in *views* (.py). This layer mediates the access to the database (through the *Model* layer), static files or any customized Python function.

Aiming at a modular implementation that can be easily tailored for different datasets or configurations, we package the implementation of RGR and other custom functions into a separate Python library (*ourLib.py*). This includes functions using the OpenCV [8] library, which are responsible for image loading and converting the outputs of RGR from mathematical arrays to images for visualization.

RGR is used as the core component of FreeLabel, and adapted in two minor aspects to compose the annotation tool. The original algorithm described in [21] focuses on the refinement of a CNN’s semantic segmentation predictions, a scenario with coarse segmentation masks as input. While for that case sampling fewer seeds is beneficial to filter out false-positives, in our scenario we aim at minimizing the required number of user interactions. Since the user inputs tend to be sparse but highly-accurate, we increase the percentage of seeds sampled in each Monte Carlo iteration to 75%, with 8 iterations per run. Moreover, we remove RGR’s constraint that automatically classifies as background any pixel significantly distant from labeled neighbors in terms of appearance and spatial position. By removing this constraint, RGR will assign to each unlabeled pixel the category provided for its nearest neighbor, regard-

less of how far they might be. If the propagated label is incorrect, the user can easily improve the segmentation by tracing an additional scribble to the corresponding region.

#### 4. Experiments and Results

We evaluate our tool in terms of: i) quality of the obtained segmentation masks; and ii) time required by users to annotate images using FreeLabel. To that end, we defined first a task where users were asked to annotate images from the PASCAL VOC 2012 dataset. We opted for this dataset as it contains good quality segmentations of multiple object categories and is widely used by the computer vision community, such that it represents a good reference standard for anyone searching for a suitable annotation tool.

Inspired by the idea of GWAP, we designed a game-like version of FreeLabel for the annotation of PASCAL images. Ideally, users must provide high-quality segmentation but also be as quick as possible, which represents a trade-off for which it is difficult to provide the annotators with clear guidelines. We therefore employ a game with a simple unified score metric that combines both annotation time and mean average precision ( $mAP$ ) between the obtained masks and corresponding ground-truth annotations, which is computed according to the official PASCAL metrics. The quality of the segmentation must be the main priority, while the time spent on each image is a secondary concern. Thus, as summarized in Figure 5, we select accuracy ( $mAP$ ) as the base factor for the score computation, with a “bonus” multiplying factor that is proportional to time spent on each image.

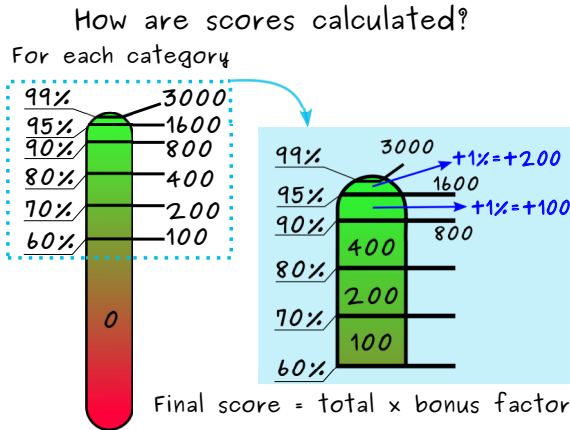


Figure 5. Score chart presented to the users as reference for the game where they are asked to label PASCAL images in an accurate and timely manner.

The main goal of this metric is to constitute feedback that tells the user how well he/she is performing the task, such that we do not focus on a more rigorous formulation for score computation. Instead, we aim at motivating the user to obtain the highest accuracy as possible by increasing the

base score progressively as the  $mAP$  approaches 100%. To motivate users to be quick, we multiply the base score with a factor that decays over time: for an image containing a single object, the bonus decays linearly according to Eq. 1, where  $t$  is the time elapsed in the image,  $T$  is the expected time for the image and  $N$  is its number of objects.

$$bonus = \max(2 + \frac{T - t}{T}, 1) \quad (1)$$

$$T = 60 + 30 \times (N - 1)[sec].$$

Based on the performance of expert labelers, we roughly estimated an expected time of 60 seconds for an image with a single object, plus an extra 30 seconds per object in the presence of two or more objects. With this formulation, the bonus will be  $2\times$  if the user annotates the image in the expected time  $T$ , linearly decaying to  $1\times$  if the annotation time takes more than twice the expected duration.

After showing the participants a training video, we asked seven different users to label an average of 25 images each, in a task expected to take 1 hour. We followed the official PASCAL annotation guidelines [26]<sup>2</sup>, indicating with bounding boxes the objects to be annotated by the users.

Figure 6 summarizes the average accuracy ( $mAP$ ) and average time necessary to annotate the different objects in the images. Overall, users provided segmentations with 92.8% average overlap with the ground-truth masks, at a mean pace of 61.3 seconds per object. As a reference, this is significantly quicker than the average 79 sec/object required for annotating the COCO dataset using the OpenSurfaces tool [35].

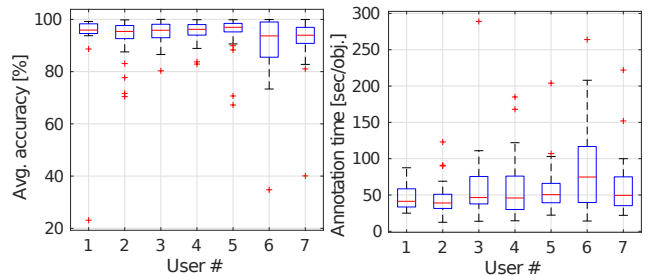


Figure 6. Distribution of the accuracies and annotation times obtained by users annotating images from the PASCAL dataset.

A more detailed analysis further highlights the qualities of FreeLabel for image annotation. Median values of 95.5% overall accuracy and 50.1 seconds per object suggest the presence of outliers, which is corroborated by an analysis per object category. As depicted in Figure 8, objects from the categories *bicycle*, *chair* and *pottedplant* are significantly harder to label than instances from classes such as *airplane*, *cows* and *trains*, which present fewer enclosed regions or thin structures. However, despite requiring longer

<sup>2</sup><http://host.robots.ox.ac.uk/pascal/VOC/voc2011/guidelines.html>



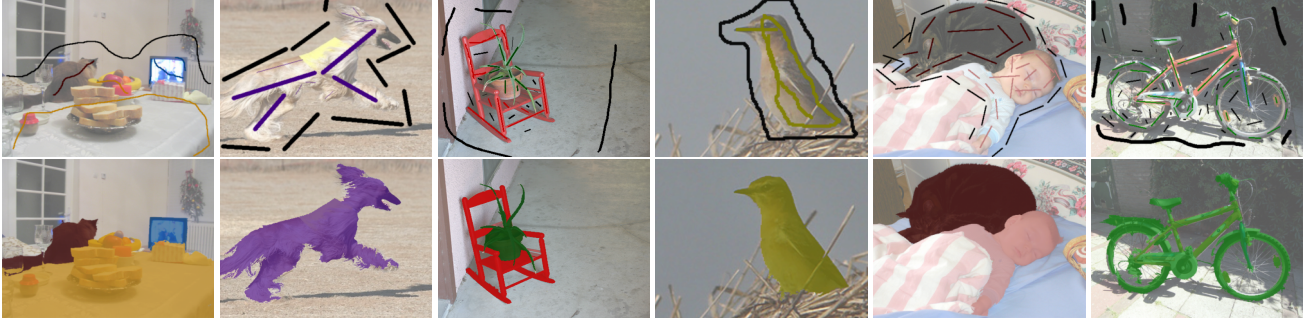


Figure 7. Examples of annotations provided by users for the PASCAL dataset using FreeLabel. *Top*: user annotations. *Bottom*: final grown mask generated by FreeLabel from the corresponding inputs.

annotation times, high-quality segmentations can still be obtained for such harder categories. Figure 7 is a compilation of annotation examples provided by the users, with the rightmost *bicycle* example illustrating the quality of segmentation that can be obtained even for harder cases.

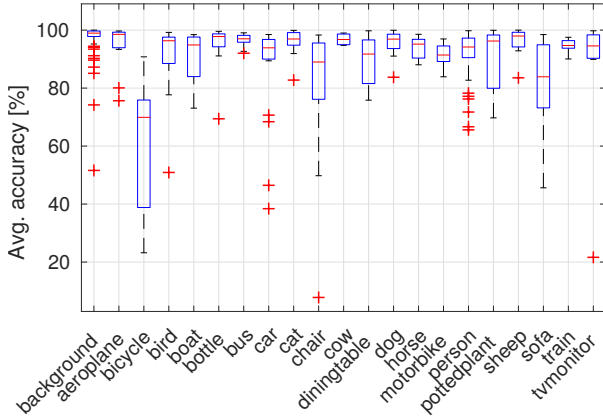


Figure 8. Distribution of average accuracy for objects of the different categories in the PASCAL dataset.

We also observed which strategies were adopted by the most successful users. The left plot in Figure 9 summarizes the frequency of usage of the *Refine* button by each user, while in the right is the average image area covered by each user’s scribbles. User #4 exemplifies the usefulness of interactivity using RGR: by frequently using the *Refine* option, this user obtained one of the highest accuracy averages, with fewer low-quality outliers. This user also drew fewer traces and thus finished the task faster than others who provided annotations of similar quality.

#### 4.1. Annotation of unlabeled images

To demonstrate the suitability of FreeLabel for the realistic scenario of annotating unlabeled datasets, we performed a second round of experiments where 8 users were asked to annotate images of a significantly different dataset. We chose the dataset made publicly available in [24, 23], which contains images of multiple species of fruit-flowers

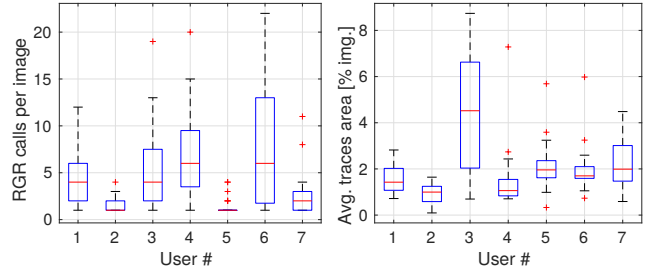


Figure 9. Number of *Refine* calls and average image area covered by user traces for annotating images of the PASCAL dataset.

that were acquired under varied conditions. Since the images comprising this dataset are of high-resolution ( $2704 \times 1520$ )px and contain dozens of small flowers, we decided to split each image into 16 blocks of  $676 \times 380$ px.

With the lessons learned from the PASCAL experiments, we designed a new training sequence (video available as Supplementary Material) that emphasizes good strategies for efficient labeling with FreeLabel. Before annotating the flowers, all users were required to annotate 10 PASCAL images with a minimum accuracy of 90% per category. Our rationale behind this strategy is that annotating the PASCAL images in a game-format works as a training session in which the users become familiar with the interface and grasp the main guidelines to keep in mind for annotating any type of image segmentation dataset.

Preliminary experiments demonstrated that the lack of performance feedback harms the motivation of the users to perform the task and, as consequence, the quality of segmentation obtained. For this reason, we structured the annotation sessions such that each user was required to label 9 blocks of different flower images, in batches of 3 blocks each. Each batch contained 2 non-annotated blocks and 1 block for which ground-truth was available. We used the ground-truth image blocks as checkpoints: if the segmentation provided by the user did not meet a certain accuracy threshold, the user would have to redo the entire batch of 3 images. The ground-truth annotations are never shown



Figure 10. Examples of flower annotations provided by users using FreeLabel. The colormap boundaries illustrate how many users labeled the enclosed regions as flower. Colors proportionally range from dark blue (one user) to dark red (all users labeled it as flower).

to the users, such that while only every third image is actually used to compute the average accuracy, we “deceive” the users to believe that all images are verified and must thus be accurately labeled. Moreover, we used a rather lower accuracy threshold of 70%, as the main intent is just to avoid very poor annotations.

Results demonstrate the effectiveness of this strategy for annotation of unlabeled images. In Figure 10, the colormap progressively ranging from blue to red illustrates for each enclosed region how many users labeled it as flower. This representation qualitatively demonstrates how the annotations provided by the different users for the three different datasets converge to ideal segmentation masks. Such convergence suggests that majority voting can be used to approximate ideal segmentation masks, which we then use to statistically evaluate the variability of the annotations provided for images without ground-truth.

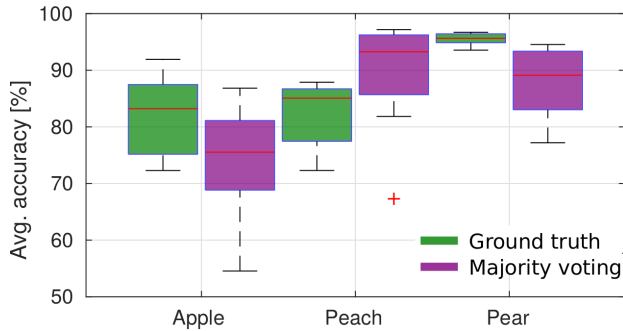


Figure 11. Distribution of the average accuracy obtained by the users for annotation of flower datasets.

Figure 11 summarizes the average accuracy and deviations observed for the images with and without ground-truth available (in green and purple, respectively). The average overlap between the segmentations provided by the users and the available ground-truth masks were higher than 80% for the three different datasets, reaching 95.5% for the *Pear* image. The higher deviations for the *Apple* and *Peach* datasets are mostly associated with the annotation of small flower buds and mistakes related to bright leaves on the apple images. Such mistakes are visible as well in the examples in Figure 10. Finally, the deviations observed for

ground-truth images are similar to the ones observed for the images without ground-truth, which indicates a somewhat consistent performance of users for both groups of images.

## 5. Conclusion

We introduced FreeLabel, an interactive interface for fast and high-quality annotation of image segmentation datasets. In contrast to annotation tools that require drawing polygons fully enclosing objects to be segmented, FreeLabel simplifies the user interactions to freehand scribbles and straight lines. By means of the unsupervised algorithm known as RGR, such inputs are grown into segmentation masks that tightly adhere to actual object boundaries.

FreeLabel has a modular design and relies solely on open-source libraries, as we aim at a publicly available tool that can be easily adapted for annotation of a wide range of datasets. Its web-based arrangement can be deployed both locally or in external servers, allowing annotations through both private (confidential) or crowdsourced strategies.

Our experiments demonstrate that segmentations with high overlap to ground-truth annotations of the PASCAL dataset can be obtained in a matter of seconds. Through short tutorial videos and a game-like version of FreeLabel, users quickly learned how to use the tool and were capable of properly annotating significantly different datasets.

As future work, we intend to accelerate the RGR algorithm and evolve FreeLabel into a interactive tool that automatically grows the user scribbles in real-time. With minor adjustments, we believe FreeLabel could be also efficiently used with tablets and mobile devices. Moreover, we consider combining ideas such as majority voting and GWAP for annotation of unlabeled datasets. With a multiplayer design, cooperative and antagonistic roles could be exploited for both user motivation and annotation quality control.

Finally, we aim at hiring AMT workers for larger scale image annotation using FreeLabel. Feedback received from 5 AMT workers hired as a preliminary experiment included encouraging comments such as “*I was surprised how well the bounding tools worked. They seemed to accurately pick up my responses*”, and “*the interface was easy to understand for anyone mildly familiar with MS paint*”.



## Acknowledgement

We acknowledge the support of USDA ARS agreement #584080-5-020, and of NVIDIA Corporation with the donation of the GPU used for this research. We thank our colleagues Abubakar Siddique, Enrico Prampolini, Reza Mozhdehi, Jamir Jyoti and others who collaborated with data collection for this study.

## References

- [1] Django (version 2.0). <https://djangoproject.com/>. Accessed: 2018-09-13. **5**
- [2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2281, 2012. **2**
- [3] Y. Aksoy, T.-H. Oh, S. Paris, M. Pollefeys, and W. Matusik. Semantic soft segmentation. *ACM Transactions on Graphics*, 37(4):72, 2018. **3**
- [4] B. P. Bailey and J. A. Konstan. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior*, 22(4):685 – 708, 2006. Attention aware systems. **3**
- [5] S. Bell, P. Upchurch, N. Snively, and K. Bala. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Transactions on Graphics*, 32(4):111, 2013. **2**
- [6] Y. Boykov and G. Funka-Lea. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision*, 70(2):109–131, Nov. 2006. **2**
- [7] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary amp; region segmentation of objects in N-D images. In *IEEE International Conference on Computer Vision*, volume 1, pages 105–112 vol.1, 2001. **2**
- [8] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000. **5**
- [9] H. Caesar, J. Uijlings, and V. Ferrari. COCO-Stuff: Thing and Stuff Classes in Context. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 10, 2018. **1, 2**
- [10] J. Cates, A. Lefohn, and R. Whitaker. GIST: an interactive, GPU-based level set segmentation tool for 3d medical images. *Medical Image Analysis*, 8(3):217–231, Sept. 2004. **3**
- [11] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using CNNs and a discriminatively trained domain transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4545–4554, 2016. **3**
- [12] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, Apr. 2018. **1, 3**
- [13] Q. Chen, D. Li, and C.-K. Tang. Knn matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2175–2188, 2013. **3**
- [14] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar. Counting Apples and Oranges With Deep Learning: A Data-Driven Approach. *IEEE Robotics and Automation Letters*, 2(2):781–788, Apr. 2017. **1**
- [15] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1189–1198, 2018. **3**
- [16] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–271. IEEE, 2001. **3**
- [17] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. **2**
- [18] D. Cremers, O. Fluck, M. Rousson, and S. Aharon. A probabilistic level set formulation for interactive organ segmentation. In *Medical Imaging 2007: Image Processing*, volume 6512, page 65120V. International Society for Optics and Photonics, Mar. 2007. **3**
- [19] D. Cremers, M. Rousson, and R. Deriche. A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape. *International Journal of Computer Vision*, 72(2):195–215, Apr. 2007. **3**
- [20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. **1**
- [21] P. A. Dias and H. Medeiros. Semantic Segmentation Refinement by Monte Carlo Region Growing of High Confidence Detections. *arXiv:1802.07789 [cs]*, Feb. 2018. arXiv: 1802.07789. **2, 3, 4, 5**
- [22] P. A. Dias, A. Tabb, and H. Medeiros. Apple flower detection using deep convolutional networks. *Computers in Industry*, 99:17–28, Aug. 2018. **2**
- [23] P. A. Dias, A. Tabb, and H. Medeiros. Data from: Multi-species fruit flower detection using a refined semantic segmentation network, 2018. **7**
- [24] P. A. Dias, A. Tabb, and H. Medeiros. Multispecies Fruit Flower Detection Using a Refined Semantic Segmentation Network. *IEEE Robotics and Automation Letters*, 3(4):3003–3010, Oct. 2018. **2, 7**
- [25] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. **2**
- [26] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. **1, 2, 6**
- [27] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 755–762 vol. 1, June 2005. **2**

- [28] M. V. Giuffrida, F. Chen, H. Scharr, and S. A. Tsafaris. Citizen crowds and experts: observer variability in image-based plant phenotyping. *Plant Methods*, 14:12, Feb. 2018. 1
- [29] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1
- [30] A. Kawrykow, G. Roumanis, A. Kam, D. Kwak, C. Leung, C. Wu, E. Zarour, L. Sarmenta, M. Blanchette, J. Waldspühl, et al. Phylo: a citizen science approach for improving multiple sequence alignment. *PloS one*, 7(3):e31362, 2012. 3
- [31] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, pages 695–711. Springer, 2016. 3
- [32] P. Krähenbühl and V. Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011. 3
- [33] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully Convolutional Instance-aware Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2367, 2017. 1
- [34] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, June 2016. 1, 3
- [35] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 1, 2, 6
- [36] Y. Liu and Y. Yu. Interactive Image Segmentation Based on Level Sets of Probabilities. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):202–213, Feb. 2012. 3
- [37] S. Mahadevan, P. Voigtlaender, and B. Leibe. Iteratively trained interactive segmentation. In *British Machine Vision Conference*, 2018. 3
- [38] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Computer Graphics and Interactive Techniques*, pages 191–198. ACM, 1995. 2
- [39] M. Rajchl, M. C. H. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Damodaram, M. A. Rutherford, J. V. Hajnal, B. Kainz, and D. Rueckert. DeepCut: Object Segmentation From Bounding Box Annotations Using Convolutional Neural Networks. *IEEE Transactions on Medical Imaging*, 36(2):674–683, Feb. 2017. 3
- [40] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 309–314, New York, NY, USA, 2004. ACM. 1, 2
- [41] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1-3):157–173, May 2008. 2
- [42] B. Schwartz. *The Paradox of Choice: Why More Is Less*. Harper Perennial. HarperCollins, 2003. 3
- [43] D. Stutz, A. Hermans, and B. Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 2017. 3
- [44] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers. Normalized Cut Loss for Weakly-Supervised CNN Segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition:10, 2018. 3
- [45] P. Tangseng, Z. Wu, and K. Yamaguchi. Looking at Outfit to Parse Clothing. *arXiv:1703.01386 [cs]*, Mar. 2017. arXiv: 1703.01386. 1, 2
- [46] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004. 3
- [47] L. Von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, 2008. 3
- [48] L. Von Ahn, M. Kedia, and M. Blum. Verbosity: a game for collecting common-sense facts. In *SIGCHI conference on Human Factors in computing systems*, pages 75–78. ACM, 2006. 3
- [49] L. Von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *SIGCHI conference on Human Factors in computing systems*, pages 55–64. ACM, 2006. 3
- [50] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently Scaling up Crowdsourced Video Annotation: A Set of Best Practices for High Quality, Economical Video Labeling. *International Journal of Computer Vision*, 101(1):184–204, Jan. 2013. 3
- [51] T. Wang, B. Han, and J. Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *Computer Vision and Image Understanding*, 120:14–30, 2014. 1, 2, 3
- [52] Q. Zhu, L. Shao, X. Li, and L. Wang. Targeting accurate object extraction from an image: A comprehensive study of natural image matting. *IEEE Transactions on Neural networks and Learning Systems*, 26(2):185–207, 2015. 3