Extreme clicking for efficient object annotation

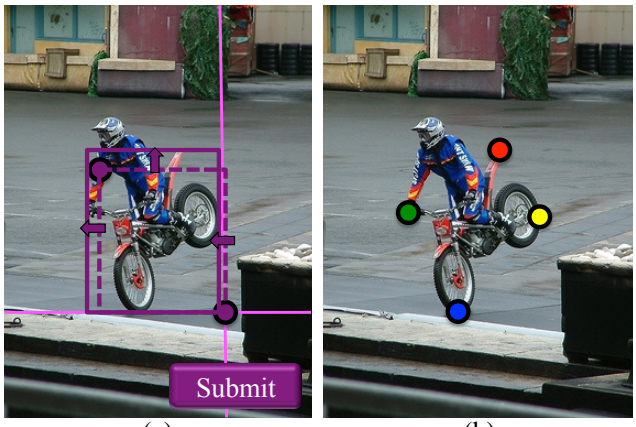
Dim P. Papadopoulos[1] Jasper R. R. Uijlings[2] Frank Keller[1] Vittorio Ferrari[1,2]
dim.papadopoulos@ed.ac.uk jrru@google.com keller@inf.ed.ac.uk vferrari@inf.ed.ac.uk
[1]University of Edinburgh [2]Google Research

Abstract

Manually annotating object bounding boxes is central to building computer vision datasets, and it is very time consuming (annotating ILSVRC [53] took 35s for one high-quality box [62]). It involves clicking on imaginary corners of a tight box around the object. This is difficult as these corners are often outside the actual object and several adjustments are required to obtain a tight box. We propose extreme clicking instead: we ask the annotator to click on four physical points on the object: the top, bottom, left- and right-most points. This task is more natural and these points are easy to find. We crowd-source extreme point annotations for PASCAL VOC 2007 and 2012 and show that (1) annotation time is only 7s per box, 5× faster than the traditional way of drawing boxes [62]; (2) the quality of the boxes is as good as the original ground-truth drawn the traditional way; (3) detectors trained on our annotations are as accurate as those trained on the original ground-truth. Moreover, our extreme clicking strategy not only yields box coordinates, but also four accurate boundary points. We show (4) how to incorporate them into GrabCut to obtain more accurate segmentations than those delivered when initializing it from bounding boxes; (5) semantic segmentations models trained on these segmentations outperform those trained on segmentations derived from bounding boxes.

1. Introduction

Drawing the bounding boxes traditionally used for object detection is very expensive. The PASCAL VOC bounding boxes were obtained by organizing an "annotation party" where expert annotators were gathered in one place to create high quality annotations [21]. But crowdsourcing is essential for creating larger datasets: Su et al. [62] developed an efficient protocol to annotate high-quality boxes using Amazon Mechanical Turk (AMT). They report 39% efficiency gains over consensus-based approaches (which collect multiple annotations to ensure quality) [13, 60]. However, even this efficient protocol requires 35s to annotate one box (more details in Sec. 2).



(a) (b)

Figure 1. **Annotating an instance of motorbike:** (a) The conventional way of drawing a bounding box. (b) Our proposed extreme clicking scheme.

Why does it take so long to draw a bounding box? Fig 1a shows the typical process [12, 21, 32, 54, 61, 62]. First the annotator clicks on a corner of an imaginary rectangle tightly enclosing the object (say the bottom-right corner). This is challenging, as these corners are typically not on the object. Hence the annotator needs to find the relevant extreme points of the object (the bottom point and the right-most point) and adjust the x- and y-coordinates of the corner to match them. After this, the annotator clicks and drags the mouse to the diagonally opposite corner. This involves the same process of x- and y-adjustment, but now based on a visible rectangle. After the rectangle is adjusted, the annotator clicks again. He/she can make further adjustments by clicking on the sides of the rectangle and dragging them until the box is tight on the object. Finally, the annotator clicks a "submit" button.

From a cognitive perspective, the above process is suboptimal. The three steps (clicking on the first corner, dragging to the second corner, adjusting the sides) effectively constitute three distinct tasks. Each task requires attention to different parts of the object and using the mouse differently. In effect, the annotator is constantly task-switching, a process that is cognitively demanding and is correlated with increased response times and errors rates [45, 52]. Further-

more, the process involves a substantial amount of mental imagery: the rectangle to be drawn is imaginary, and so are the corner points. Mental imagery also has a cognitive cost, e.g. in mental rotation experiments, response time is proportional to rotation angle [35, 57].

In this paper we propose an annotation scheme which avoids task switching and mental imagery, resulting in greatly improved efficiency. We call our scheme *extreme clicking*: we ask the annotator to click on four extreme points of the object, i.e. points belonging to the top, bottom, left-most, and right-most parts of the object (Fig 1b). This has several advantages: (1) Extreme points are not imaginary, but are well-defined physical points on the object, which makes them easy to locate. (2) No rectangle is involved, neither real nor imaginary. This further reduces mental imagery, and avoids the need for detailed instructions defining the notion of a bounding box. (3) Only a single task is performed by the annotator thus avoiding task switching. (4) No separate box adjustment step is required. (5) No "submit" button is necessary; annotation terminates after four clicks.

Additionally, extreme clicking provides *more information* than just box coordinates: we get four points on the actual object boundary. We demonstrate how to incorporate them into GrabCut [51], to deliver more accurate segmentations than when initializing it from bounding boxes [51]. In particular, GrabCut relies heavily on the initialization of the object appearance model (e.g. [39, 51, 68]) and on which pixels are clamped to be object/backgound. When using just a bounding box, the object appearance model is initialized from all pixels within the box (e.g. [23, 39, 51]). Moreover, it typically helps to clamp a smaller central region to be object [23]. Instead, we first expand our four object boundary points to an estimate of the whole contour of the object. We use this estimate to initialize the GrabCut object appearance model. Furthermore, we skeletonize the estimate and clamp the resulting pixels to be object.

We perform extensive experiments on PASCAL VOC 2007 and 2012 using crowd-sourced annotations which demonstrate: (1) extreme clicking only takes 7s seconds per box, $5\times$ faster than the traditional way of drawing boxes [62]; (2) extreme clicking leads to high-quality boxes on a par with the original ground-truth boxes drawn the traditional way; (3) detectors trained on boxes generated using extreme clicking perform as well as those trained on the original ground-truth; (4) incorporating extreme points into GrabCut [51] improve object segmentations by 2%-4% mIoU over initializing it from bounding boxes; (5) semantic segmentations models trained on segmentations derived from extreme clicking outperform those trained on segmentations generated from bounding boxes by 2.6% mIoU.

2. Related work

Time to draw a bounding box. The time required to draw a bounding box varies depending on several factors, including the quality of the boxes and the crowdsourcing protocol used. In this paper, as an authoritative reference we use the protocol of [62] which was used to annotate ILSVRC [53]. It was designed to produce high-quality bounding boxes with little human annotation time on Amazon Mechanical Turk. They report the following median times for annotating an object of a given class in an image [62]: 25.5s for drawing one box, 9.0s for verifying its quality, and 7.8s for checking whether there are other objects of the same class yet to be annotated. Since we only consider annotating one object per class per image, we use $25.5s+9.0s = 34.5s$ as the reference time. This is a conservative estimate: when taking into account that some boxes are rejected and need to be re-drawn, the median time increases to 55s. If we use average times instead of medians, the cost raises further to 117s.

Note how both PASCAL VOC and ILSVRC have images of comparable difficulty and come with ground-truth box annotations of similar high quality [53], justifying our choice of 35s reference time. Papers reporting faster timings [32, 54] aim for lower-quality boxes (e.g. the official annotator instructions of [32] show an example box which is not tight around the object). We compare to [54] in Sec. 5.

Reducing annotation time for training object detectors. Weakly-supervised object localization techniques (WSOL) can be used to train object detectors from image-level labels only (without bounding boxes) [5, 11, 14, 55, 59]. This setting is very cheap in terms of annotation time, but it produces lower quality object detectors, typically performing at only about half the level of accuracy achieved by training from bounding boxes [5, 11, 14, 55, 67].

Training object class detectors from videos could bypass the need for manual bounding boxes, as the motion of the objects facilitates their automatic localization [49, 40, 41]. However, because of the domain adaptation problem, these detectors are still quite weak compared to ones trained on manually annotated still images [34]. Alternative types of supervision information such as eye-tracking data [44, 46], text from news articles or web pages [17, 28], or even movie scripts [7] have also been explored. Papadopoulos et al. [47] propose a scheme for training object class detectors which only requires annotators to verify bounding boxes generated automatically by the learning algorithm. We compare our extreme clicking scheme to state-of-the-art WSOL [5], and to [47] in Sec. 5.

(Interactive) object segmentation. Object segmentations are significantly more expensive to obtain than bounding boxes. The creators of the SBD dataset [29] merged five annotations per instance, resulting in a total time of 315s per instance. For COCO [43], 79s per instance were

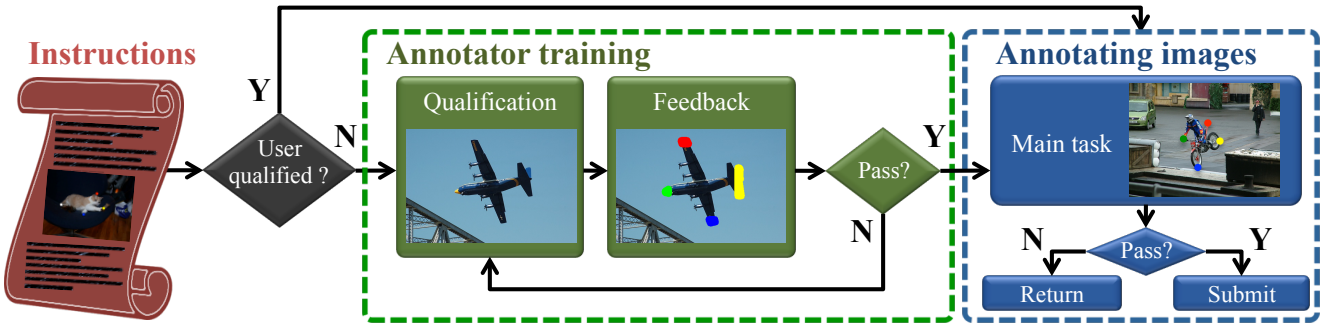


Figure 2. **The workflow of our crowd-sourcing protocol for collecting extreme click annotations on images.** The annotators read a set of instructions and then go through an interactive training stage that consists of a qualification test at the end of which they receive a detailed feedback on how well they performed. Annotators who successfully pass the test can proceed to the annotation stage. In case of failure, they are allowed to repeat the test as many times as they want until they succeed.

required for drawing object polygons, excluding verifying correctness and possibly redrawing. To reduce annotation time many interactive segmentation techniques have been proposed, which require the user to input either a bounding box around the object [42, 51, 71], or scribbles [3, 16, 24, 26, 27, 42, 50, 64, 65, 72], or clicks [31, 69]. Most of this work is based on the seminal GrabCut algorithm [51], which iteratively alternates between estimating appearance models (typically Gaussian Mixture Models [6]) and refining the segmentation using graph cuts [9]. The user input is typically used to initialize the appearance model and to clamp some pixels to background. In this paper, we incorporate extreme clicks into GrabCut [51], improving the appearance model initialization and automatically selecting good seed pixels to clamp to object.

3. Collecting extreme clicks

In this section, we describe our crowd-sourcing framework for collecting extreme click annotations (Fig. 2). Annotators read a simple set of instructions (sec. 3.1) and then go through an interactive training stage (sec. 3.2). Those who successfully pass the training stage can proceed to the annotation stage (sec. 3.3).

3.1. Instructions

The annotators are given an image and the name of a target object class. They are instructed to click on four extreme points (top, bottom, left-most, right-most) on the visible part of any object of this class. They can click the points in any order. In order to let annotators know approximately how long the task will take, we suggest a total time of 10s for all four clicks. This is an upper bound on the expected annotation time that we estimated from a small pilot study.

Note that our instructions are extremely simple, much simpler than those necessary to explain how to draw a bounding box in the traditional way (e.g. [54, 62]). They are also simpler than instructions required for verifying whether a displayed bounding box is correct [47, 54, 62]. That requires the annotator to imagine a perfect box on the object, and to mentally compare it to the displayed one.

3.2. Annotator training

After reading the instructions, the annotators go through the training stage. They have to complete a qualification test, at the end of which they receive detailed feedback on how well they performed. Annotators who successfully pass this test can proceed to the annotation stage. In case of failure, annotators can repeat the test until they succeed.

Qualification test. A qualification test is a good mechanism for enhancing the quality of crowd-sourcing data and for filtering out bad annotators and spammers [1, 19, 33, 37]. Some annotators do not pay attention to the instructions or do not even read them. Qualification tests have been successfully used to collect image labels, object bounding boxes, and segmentations for some of the most popular datasets (e.g., COCO [43] and Imagenet [53, 62]).

The qualification test is designed to mimic our main task of clicking on the extreme points of objects. We show the annotator a sequence of 5 different images with the same object class and ask them to carry out the extreme clicking task.

Feedback. The qualification test uses a small pool of images with ground-truth segmentation masks for the objects, which we employ to automatically validate the annotator's clicks and to provide feedback (Fig. 2, middle part). We take a small set of qualification images from a different dataset than the one that we annotate.

In the following, we explain the validation procedure for the top click (the other three cases are analogous). We ask the annotator to click on a top point on the object, but this point is not necessarily uniquely defined. Depending on the object shape, there may be multiple points that are equivalent, up to some tolerance margin (e.g. the top of the dog's head in fig. 3, top row). Clearly, clicking on any of these points is correct. The area in which we accept the annotator's click is derived from the segmentation mask. First, we find the pixels with the highest y-coordinate in it (there might be multiple such pixels). Then, we select all pixels in

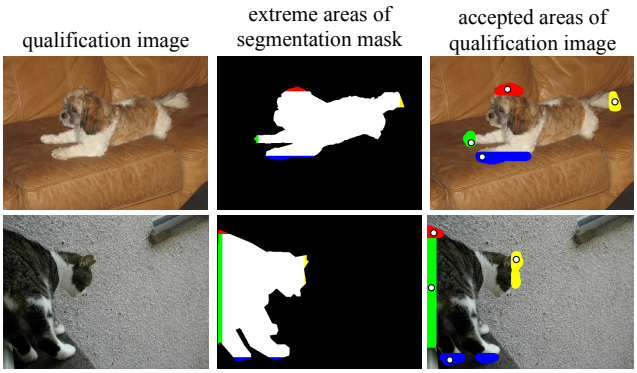


Figure 3. **Qualification test.** (Left) Qualification test examples of the dog and cat class. (Middle) The figure-ground segmentation masks we use to evaluate annotators' extreme clicks during the training stage. The pixels of the four extreme areas of the mask are marked with colors. (Right) The accepted areas for each extreme click and the click positions as we display them to the annotators as feedback.

the mask with y-coordinates within 10 pixels of any of these top pixels (red area in Fig. 3, middle column). Finally, we also include in the accepted area all image pixels within 10 pixels of any of the selected pixels in the segmentation mask (Fig. 3, right column). Thus the accepted area includes all top pixels in the mask, plus a tolerance region around them, both inside and outside the mask.

After the annotators finish the qualification test, they receive a feedback page with all the examples they annotated. For each image, we display the annotator's four clicks, and the accepted areas for each click (Fig. 3 right column).

Success or failure. The annotators pass the qualification test if all their clicks on all 5 qualification images are inside the accepted areas. Those that pass the test are recorded as qualified annotators and can proceed to the main annotation stage. A qualified annotator never has to retake the qualification test. In case of failure, annotators are allowed to repeat the test as many times as they want. The combination of automatically providing rich feedback and allowing annotators to repeat the test makes the training stage interactive and highly effective. Annotators that have reached the desired level of quality can be expected to keep it throughout the annotation [30].

3.3. Annotating images

In the annotation stage, annotators are asked to annotate small batches of 10 consecutive images. To increase annotation efficiency, the target class for all the images within a batch is the same. This means annotators do not have to re-read the class name for every image and can use their prior knowledge of the class to find it rapidly in the image [63]. More generally, it avoids task-switching which is well-known to increase response time and decrease accuracy [52, 45].

Quality control. Quality control is a common process when crowd-sourcing image annotations [4, 36, 43, 53, 56, 60, 62, 66, 70]. We control the quality of the annotation by hiding one evaluation image for which we have a ground-truth segmentation inside a 10-image batch, and monitor the annotator's accuracy on it (golden question). Annotators that fail to click inside the accepted areas on this evaluation image are not able to submit the task. We do not do any post-processing rejection of the submitting data.

4. Object segmentation from extreme clicks

Extreme clicking results not only in high-quality bounding box annotations, but also in four accurate object boundary points. In this section we explain how we use these boundary points to improve the creation of segmentation masks from bounding boxes.

We cast the problem of segmenting an object instance in image I as a pixel labeling problem. Each pixel $p \in I$ should be labeled as either object ($l_p = 1$) or background ($l_p = 0$). A labeling L of all pixels represents the segmented object. Similar to [51], we employ a binary pairwise energy function E defined over the pixels and their labels.

$$E(L) = \sum_p U(l_p) + \sum_{p,q} V(l_p, l_q) \qquad (1)$$

U is a unary potential that evaluates how likely a pixel p is to take label l_p according to the object and background appearance models, while the pairwise potential V encourages smoothness by penalizing neighboring pixels taking different labels.

Initial object surface estimate from extreme clicks. For GrabCut to work well, it is important to have a good initial estimate of the object surface to initialize the appearance model. Additionally, it helps to clamp certain pixels to object [39]. We show how the four collected object boundary points can be exploited to do both.

In particular, for each pair of consecutive extreme clicks (e.g. leftmost-to-top, or top-to-rightmost) we find the path connecting them which is most likely to belong to the object boundary. For this purpose we first apply a strong edge detector [15] to obtain a boundary probability $e_p \in [0, 1]$ for every pixel p of the image (second row of Fig. 4). We then define the best boundary path between two consecutive extreme clicks as the shortest path whose minimum edge-response is the highest (third row of Fig. 4, magenta). We found this objective function to work better than others, such as minimizing $\sum_p (1 - e_p)$ for pixels p on the path. The resulting object boundary paths yield an initial estimate of the object outlines.

We use the surface within the boundary estimates (shown in green in the third row of Fig. 4) to initialize the object appearance model used for U in Eq. (1). Furthermore, from

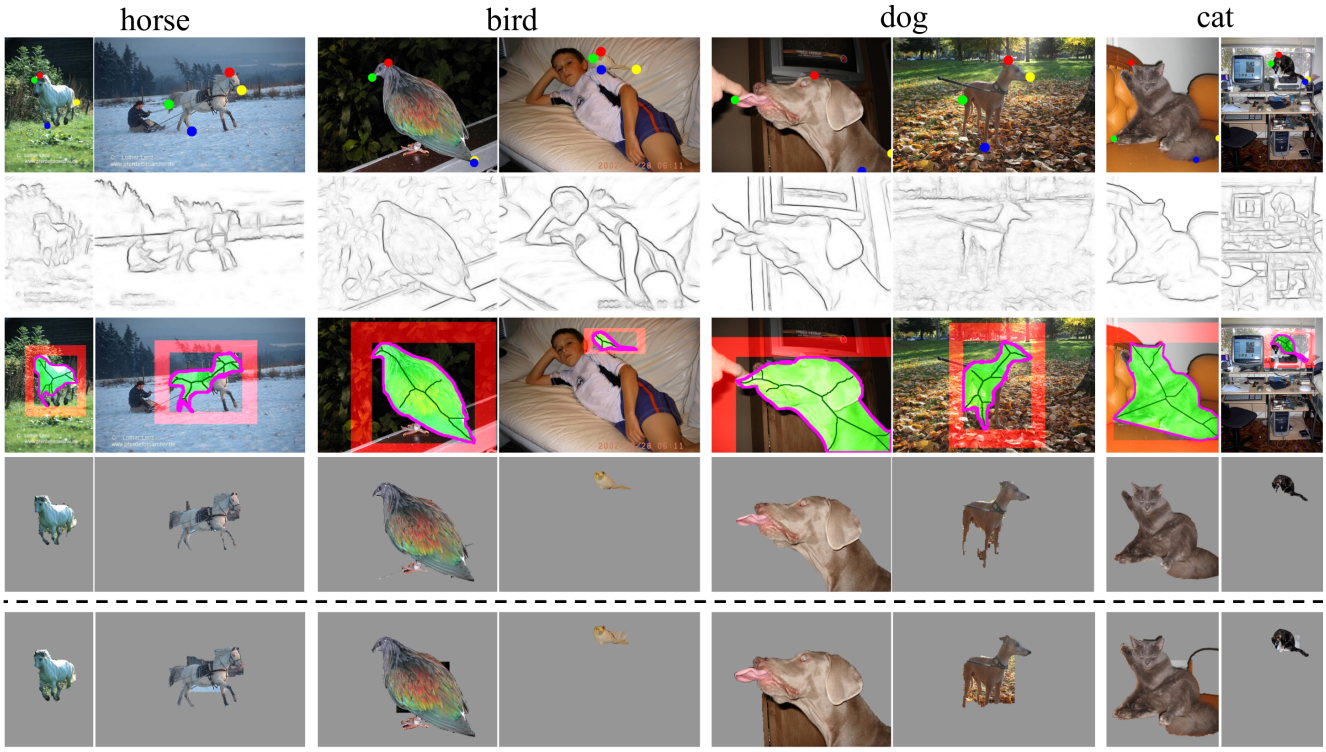


Figure 4. **Visualization of input cues and output of GrabCut.** First row shows input with annotator's extreme clicks. Second row shows output of edge detector [15]. Third row shows our inputs for GrabCut: the pixels used to create background appearance model (red), the pixels used to create the object appearance model (bright green), the initial boundary estimate (magenta), and the skeleton pixels which we clamp to have the object label (dark green). Fourth row shows the output of GrabCut when using our new inputs, while the last row shows the output when using only a bounding box.

this surface we obtain a skeleton using standard morphology (shown in dark green in third row of Fig. 4). This skeleton is very likely to be object, so we clamp its pixel-labels to be object ($l_s = 1$ for all pixels s on the skeleton).

Appearance model. As in classic GrabCut [51], the appearance model consists of two GMMs, one for the object (used when $l_p = 1$) and one for the background (used when $l_p = 0$). Each GMM has five components, where each is a full-covariance Gaussian over the RGB color space.

Traditional interactive segmentation techniques [42, 51, 71] start from a manually drawn bounding box and estimate the initial appearance models from all pixels inside the box (object model) and all pixels outside it (background model). However, this may be suboptimal: since we are trying to segment the object within the box, intuitively only the immediate background is relevant, not the whole image. Indeed, we improved results by using a small ring around the bounding box for initializing the background model (see third row Fig. 4 in red). Furthermore, not all pixels within the box belong to the object. But given only a bounding box as input, the best is to still use the whole box to initialize the object model. Therefore, in our baseline GrabCut implementation, the background model is initialized from the immediate background and the object model is initialized from all pixels within the box.

However, because we have extreme clicks we can do better. We use them to obtain an initial object surface estimate (described above) from which we initialize the object appearance model. Fig. 5 illustrates how this improves the unary potentials U resulting from the appearance models.

Clamping pixels. GrabCut sometimes decides to label all pixels either as object or background. To prevent this, one can clamp some pixels to a certain label. For the background, all pixels outside the bounding box are typically clamped to background. For the object, one possible approach is to clamp a small area in the center of the box [23]. However, there is no guarantee that the center of the box is on the object, as many objects are not convex. Moreover, the size of the area to be clamped is not easy to set.

In this paper, we estimate the pixels to be clamped by skeletonizing the object surface estimate derived from our extreme clicks (described above). In Sec. 6 we show how our proposed object appearance model initialization and clamping scheme affect the final segmentation quality.

Pairwise potential V. The summation over (p, q) in (1) is defined on an eight-connected pixel grid. Usually, this penalty depends on the RGB difference between pixels, being smaller in regions of high contrast [8, 6, 27, 42, 51, 64]. In this paper, we instead use the sum of the edge responses of the two pixels given by the edge detector [15]. In Sec. 6

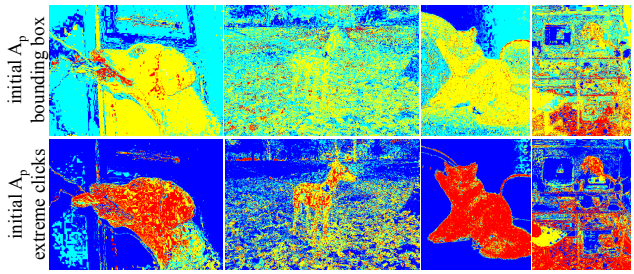


Figure 5. Posterior probability of pixels belonging to object. For both rows the background appearance model is created by using an area outside the initial box (see Fig. 4). In the first row the object model is created using the area inside the initial box. In the second row the object model is created from the object surface estimated using extreme clicks (Fig. 4, third row in light-green). Predictions from the appearance model using extreme clicks are visibly better.

we evaluate both pairwise potentials and show how they affect the final segmentation.

Optimization. After the initial estimation of appearance models, we follow [51] and alternate between finding the optimal segmentation L given the appearance models, and updating the appearance models given the segmentation. The first step is solved globally optimally by minimizing (1) using graph-cuts [9], as our pairwise potentials are submodular. The second step simply fits GMMs to labeled pixels.

5. Extreme Clicking Results

We implement our annotation scheme on Amazon Mechanical Turk (AMT) and collect extreme click annotations for both the trainval set of PASCAL VOC 2007 [20] (5011 images) and the training set of PASCAL VOC 2012 [22] (5717 images), which contain 20 object categories. For every image we annotate a single instance per class (if present in the image), which enables direct comparison to other methods described below. We compare methods both in terms of efficiency and quality.

Compared methods. Our main comparisons are to the existing ground-truth bounding boxes of PASCAL VOC. As discussed in Sec. 2, we use 34.5s as the reference time necessary to produce one such high quality bounding box by drawing it the traditional way [62].

At the other extreme, it is possible to obtain lower quality bounding boxes automatically at zero extra costs by using weakly supervised methods, which only input image-level labels. We compare to the recent method of [5].

We also compare to two methods which strike a trade-off between accuracy and efficiency [54, 47]. In [54], manual box drawing is part of a complex computer-assisted annotation system. Papadopoulos et al. [47] propose an annotation scheme that only requires annotators to verify boxes automatically generated by a learning algorithm. Importantly, both [47, 54] report both annotation time and quality, enabling proper comparisons.

Evaluation measures. For evaluating efficiency we report time measurements, both in terms of annotating the whole dataset and per instance.

We evaluate the quality of bounding boxes with respect to the PASCAL VOC ground-truth. We do this with respect to the ground-truth bounding boxes (*GT Boxes*), but also with respect to bounding boxes which we fit to the ground-truth *segmentations* (*GT SegBoxes*). We quantify quality by intersection-over-union (IoU) [21], where we measure the percentage of bounding boxes we annotated per object class with IoU greater than 0.5 and 0.7, and then take the mean over all classes (IoU>0.5, IoU>0.7). In addition, we calculate the average IoU for all instances of a class and take the mean over all classes (mIoU).

As an additional measure of accuracy we measure detector performance using Fast-RCNN [25], trained either on our extreme click boxes or on the PASCAL GT Boxes.

5.1. Results on quality and efficiency

PASCAL ground-truth boxes vs. extreme clicks. Table 1 reports the results. Having two sets of ground-truth boxes enables us to measure the agreement among the expert annotators that created PASCAL. Comparing GT Boxes and GT SegBoxes reveals this agreement to be at 88% mIoU on VOC 2007. Moreover, 93% of all GT Boxes have IoU > 0.7 with their corresponding GT SegBox. This shows that the ground-truth annotations are highly consistent, and these metrics represent the quality of the ground-truth itself. Similar findings apply to VOC 2012.

Interestingly, the boxes derived from our extreme clicks achieve equally high metrics, when compared to the PASCAL ground-truth annotations. Therefore our extreme click annotations yield boxes with a quality within the agreement among expert-annotators using the traditional way of drawing. To get a better feeling for such quality, if we perturb each of the four coordinates of the GT Boxes by 4 pixels, the resulting boxes also have 88% mIoU with the unperturbed annotations.

To further demonstrate the quality of extreme clicking, we train Fast-RCNN [25] using either PASCAL GT Boxes or extreme click boxes. We train on PASCAL VOC 2007s trainval set and test on its test set, then we train on VOC 2012s train and test on its val set. We experiment using AlexNet [38] and VGG16 [58]. Performance when training from GT Boxes or from our boxes is identical on both datasets and using both base networks.

Annotation efficieny. In terms of annotation efficiency, extreme clicks are $5\times$ cheaper: 7.0s instead of 34.5s. This demonstrates that extreme clicking costs only a fraction of the annotation time of the widely used box-drawing protocol [12, 21, 54, 61, 62], without any compromise on quality.

Human verification [47] vs. extreme clicks. Table 2 compares extreme clicks to human verification [47] on VOC

		Annotation quality w.r.t. GT SegBoxes			Detector performance (mAP)		Annotation time	
Dataset	Annotation approach	mIoU	IoU>0.7	IoU>0.5	AlexNet	VGG16	dataset (h)	instance (s)
PASCAL VOC 2007	Extreme clicks	88	92	98	56	66	14.3	7.0
	PASCAL GT Boxes	88	93	98	56	66	70.0	34.5
PASCAL VOC 2012	Extreme clicks	87	91	95	52	62	16.8	7.2
	PASCAL GT Boxes	87	90	96	52	62	79.8	34.5

Table 1. Comparison of extreme clicking and PASCAL VOC ground-truth.

		Annotation quality w.r.t. GT Boxes			Detector performance (mAP)		Annotation time	
Dataset	Annotation approach	mIoU	IoU>0.7	IoU>0.5	AlexNet	VGG16	dataset (h)	instance (s)
PASCAL VOC 2007	Extreme clicks	88	94	97	56	66	14.3	7.0
	Human verification [47]	–	–	81	50	58	9.2	4.5
	WSOL: Bilen and Vedaldi [5]	–	–	54	35	35	0	0
ILSVRC (subset)	box drawing in [54]	–	71	–	–	–	–	12.3

Table 2. Comparison of extreme clicking and alternative fast annotation approaches.

2007. While verification is 1.6× faster, our bounding boxes are much more accurate (97% correct at IoU>0.5, compared to 81% for [47]). Additionally, detector performance at test time is 6%-8% mAP higher for extreme clicking.

Weak supervision vs. extreme clicks. Weakly supervised methods are extremely cheap in human supervision time. However, the recent work [5] reports 35% mAP using VGG16, which is only about half the result brought by extreme clicking (66% mAP, Table 2).

Box drawing [54] vs. extreme clicks. Finally, we compare to [54] in Table 2. This is an approximate comparison as measurements of their box-drawing component are done on an unspecified subset of ILSVRC 2014. However, as ILSVRC and PASCAL VOC are comparable in both quality of annotations and difficulty of the dataset [53], this comparison is representative. In [54] they report 12.3s for drawing a bounding box, where 71% of the drawn boxes have an IoU>0.7 with the ground-truth box. This suggests that bounding boxes can be drawn faster than reported in [62] but this comes with a significant drop in quality. In contrast, extreme clicking costs 7s per box and 91%-94% of those boxes have IoU>0.7. Hence our protocol to annotate bounding boxes is both faster and more accurate.

5.2. Additional analysis

Per-click response-time. We examine the mean response time per click during extreme clicking. Interestingly, the first click on an object takes about 2.5s, while subsequent clicks take about 1.5s. This is because the annotator needs to find the object in the image before they can make the first click. Interestingly, 1s visual search is consistent with earlier findings [18, 46].

Influence of qualification test and quality control. We conducted three crowd-sourcing experiments on 200 trainval images of PASCAL VOC 2007 to test the influence of using a qualification test and quality control. We report the quality of the bounding boxes derived from extreme clicks in Tab. 3. Using a qualification test vastly improves annotation quality (from 75.4% to 85.7% mIoU). The quality control brings a smaller further improvement to 87.1% mIoU.

Actual Cost. We paid the annotators $0.15 to annotate a batch of 10 images which, based on our timings, is about $7.7 per hour. The total cost for annotating the whole trainval set of PASCAL VOC 2007 and the training set of PASCAL VOC 2012 was $147 and $167, respectively.

Qualification test	Quality control	mIoU	IoU>0.7
		75.4	68.0
✓		85.7	91.0
✓	✓	87.1	92.5

Table 3. Influence of the qualification test and quality control on the accuracy of extreme click annotations (on 200 images from PASCAL VOC 2007).

6. Results on Object Segmentation

This section demonstrates that one can improve segmentation from a bounding box by using also the boundary points which we obtain from extreme clicking.

6.1. Results on PASCAL VOC

Datasets and Evaluation. We perform experiments on VOC 2007 and VOC 2012. The trainval set of the segmentation task of VOC 2007 consists of 422 images with ground-truth segmentation masks of 20 classes. For VOC 2012, we evaluate on the training set, using as reference ground-truth the augmented masks set by [29] (5623 images).

To evaluate the output object segmentations, for every class we compute the intersection over union (IoU) between the predicted and ground-truth segmentation mask, and report the mean IoU over all object classes (mIoU). Some pixels in VOC 2007 are labeled as 'unknown' and are excluded from evalutation. For these experiments we use structured edge forests [15] to predict object boundaries, which is trained on BSD500 [2].

GrabCut from PASCAL VOC GT Boxes. We start with establishing our baseline by using GrabCut on the original GT Boxes of VOC (for which no boundary points are available). Since applying [51] directly leads to rather poor performance on VOC 2007 (37.3% mIoU), we first optimize GrabCut on this dataset using methods discussed in Sec. 4. Our optimized model has the following properties: the object appearance model is initialized from all pixels within the box. The background appearance model is initialized from a small ring around the box which has twice the area

of the bounding box. A small rectangular core centered within the box whose area is a quarter of the area of the box is clamped to be object. All pixels outside the box are clamped to be background. As pairwise potential, instead of using standard RGB differences, we use the summed edge responses of [15] of the corresponding pixels. All modifications together substantially improve results to 74.4% mIoU on VOC 2007. We then run GrabCut again on VOC 2012 using the exact same settings optimized for VOC 2007, obtaining 71.0% mIoU.

GrabCut from extreme clicking. Thanks to our extreme clicking annotations, we also have object boundary points. Starting from the optimized GrabCut settings established in the previous paragraph, we make use of these boundary points to (1) initialize a better object appearance model, and (2) choose better pixels to clamp to object. As described in Sec. 4, we use the extreme clicks to estimate an initial contour of the object by following predicted object boundaries [15]. We use the surface bounded by this contour estimate to initialize the appearance model. We also skeletonize this surface and clamp the resulting pixels to be object. The resulting model yields 78.1% mIoU on VOC 2007 and 72.7% on VOC 2012. This is an improvement of 3.7% (VOC 2007) and 1.7% (VOC 2012) over the strong baseline we built. Fig. 4 shows qualitative results comparing GrabCut segmentations starting from GT Boxes (last row) and those based on our extreme clicking annotations (second-last row).

6.2. Results on the GrabCut dataset

We also conducted an experiment on the Grabcut dataset [51], consisting of only 50 images. The standard evaluation measure is the error rate in terms of the percentage of mislabelled pixels. For this experiment, we simulate the extreme click annotation by using the extreme points of the ground-truth segmentation masks of the images.

When we perform GrabCut from bounding boxes, we obtain an error rate of 8%. When using additionally the boundary points from simulated extreme clicking, we obtain 5.5% error, an improvement of 2.5%. This again demonstrates that boundary points contain useful information over bounding boxes alone for this task.

For completeness, we note that the state-of-the-art method on this dataset has 3.6% error [71]. This method uses a framework of superpixels and Multiple Instance Learning to turn a bounding box into a segmentation mask. In this paper we build on a much simpler segmentation framework (GrabCut). We believe that incorporating our extreme clicks into [71] would bring further improvements.

6.3. Training a semantic segmentation model

We now explore training a modern deep learning system for semantic segmentation from the segmentations derived from extreme clicking. We train DeepLab [10, 48] based on VGG-16 [58] on the VOC 2012 train set (5,623 images) and then we test on its val set (1,449 images). We measure performance using the standard mIoU measure (Tab. 4). We compare our approach to full supervision by training on the same images but using the ground-truth, manually drawn object segmentations (one instance per class per image, for fair comparison). We also compare to training on segmentations generated from GT Boxes.

	Full supervision	Segments from GT Boxes	Segments from extreme clicks
mIoU	59.9	55.8	58.4

Table 4. Segmentation performance on the val set of PASCAL VOC 2012 dataset using different types of annotations.

Full supervision yields 59.9% mIoU, which is our upper bound. As a reference, training on manual segmentations for all instances in the dataset yields 63.8% mIoU. This is 3.8% lower than in [48] since they train from train+val using the extra annotations by [29] (10.3k images).

Segments from GT Boxes result in 55.8% mIoU.

Segments from extreme clicks lead to 58.4% mIoU. This means our extreme clicking segmentations lead to a +2.6% mIoU improvement over those generated from bounding boxes. Moreover, our result is only -1.5% mIoU below the fully supervised case (given the same total number of training samples).

7. Conclusions

We presented an alternative to the common way of drawing bounding boxes, which involves clicking on imaginary corners of an imaginary box. Our alternative is extreme clicking: we ask annotators to click on the top, bottom, left- and right-most points of an object, which are well-defined physical points. We demonstrate that our method delivers bounding boxes that are as good as traditional drawing, while taking just 7s per annotation. To achieve this same level of quality, traditional drawing needs 34.5s [62]. Hence our method cuts annotation costs by a factor $5\times$ without any compromise on quality.

In addition, extreme clicking leads to more than just a box: we also obtain accurate object boundary points. To demonstrate their usefulness we incorporate them into GrabCut, and show that they leads to better object segmentations than when initializing it from the bounding box alone. Finally, we have shown that semantic segmentation models trained on these segmentations perform close to those trained with manually drawn segmentations (when given the same total number of samples).

Acknowledgement. This work was supported by the ERC Starting Grant "VisCul".

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 3

[2] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. on PAMI*, 2011. 7

[3] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV*, 2009. 3

[4] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 4

[5] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 2, 6, 7

[6] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, 2004. 3, 5

[7] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding actors and actions in movies. In *ICCV*, 2013. 2

[8] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, 2001. 5

[9] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on PAMI*, 26(9):1124–1137, 2004. 3, 6

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015. 8

[11] R. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE Trans. on PAMI*, 2016. 2

[12] CrowdFlower: https://www.crowdflower.com/. Crowdflower bounding box annotation tool. https://www.youtube.com/watch?v=lUIU2_HW4Ic, 2016. 1, 6

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[14] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *IJCV*, 2012. 2

[15] P. Dollar and C. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013. 4, 5, 7, 8

[16] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, and F. Ségonne. Segmentation by transduction. In *CVPR*, 2008. 3

[17] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002. 2

[18] K. A. Ehinger, B. Hidalgo-Sotelo, A. Torralba, and A. Olivia. Modelling search for people in 900 scenes: A combined source model of eye guidance. *Visual Cognition*, 2009. 7

[19] I. Endres, A. Farhadi, D. Hoiem, and D. A. Forsyth. The benefits and challenges of collecting richer object annotations. In *DeepVision workshop at CVPR*, 2010. 3

[20] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results, 2007. 6

[21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 1, 6

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012. 6

[23] V. Ferrari, M. Marin, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008. 2, 5

[24] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, 2005. 3

[25] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 6

[26] L. Grady. Random walks for image segmentation. *IEEE Trans. on PAMI*, 28(11):1768–1783, 2006. 3

[27] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, 2010. 3, 5

[28] A. Gupta and L. Davis. Beyond nouns: Exploiting prepositions and comparators for learning visual classifiers. In *ECCV*, 2008. 2

[29] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 2, 7, 8

[30] K. Hata, R. Krishna, L. Fei-Fei, and M. Bernstain. A glimpse far into the future: Understanding long-term crowd worker accuracy. In *CSCW*, 2017. 4

[31] S. Jain and K. Grauman. Click carving: Segmenting objects in video with point clicks. In *Proceedings of the Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016. 3

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

[33] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, 2011. 3

[34] V. Kalogeiton, V. Ferrari, and C. Schmid. Analysing domain shift factors between videos and images for object detection. *IEEE Trans. on PAMI*, 2016. 2

[35] S. M. Kosslyn, W. L. Thompson, I. J. Kim, and N. M. Alpert. Topographic representations of mental images in primary visual cortex. *Nature*, 378(6556):496–498, 1995. 2

[36] A. Kovashka and K. Grauman. Discovering attribute shades of meaning with the crowd. *IJCV*, 2015. 4

[37] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshop on 3D Representation and Recognition*, 2013. 3

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 6

[39] D. Kuettel and V. Ferrari. Figure-ground segmentation by transferring window masks. In *CVPR*, 2012. 2, 4

[40] K. Kumar Singh, F. Xiao, and Y. Jae Lee. Track and transfer: Watching videos to simulate strong human supervision for weakly-supervised object detection. In *CVPR*, 2016. 2

[41] A. Kuznetsova, S. J. Hwang, B. Rosenhahn, and L. Sigal. Expanding object detector's horizon: Incremental learning framework for object detection in videos. In *CVPR*, 2015. 2

[42] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009. 3, 5

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 2, 3, 4

[44] S. Mathe and C. Sminchisescu. Action from still image dataset and inverse optimal control to learn task specific visual scanpaths. In *NIPS*, 2013. 2

[45] S. Monsell. Task switching. *Trends in Cognitive Sciences*, 7(3):134–140, 2003. 1, 4

[46] D. P. Papadopoulos, A. D. F. Clarke, F. Keller, and V. Ferrari. Training object class detectors from eye tracking data. In *ECCV*, 2014. 2, 7

[47] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari. We don't need no bounding-boxes: Training object class detectors using only human verification. In *CVPR*, 2016. 2, 3, 6, 7

[48] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015. 8

[49] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 2

[50] B. L. Price, B. Morse, and S. Cohen. Geodesic graph cut for interactive image segmentation. In *CVPR*, 2010. 3

[51] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 2, 3, 4, 5, 6, 7, 8

[52] J. S. Rubinstein, D. E. Meyer, and J. E. Evans. Executive control of cognitive processes in task switching. *Journal of Experimental Psycholgy: Human Perception and Performance*, 27(4):763–797, 2001. 1, 4

[53] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 1, 2, 3, 4, 7

[54] O. Russakovsky, L.-J. Li, and L. Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In *CVPR*, 2015. 1, 2, 3, 6, 7

[55] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *ECCV*, 2012. 2

[56] B. C. Russell, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *IJCV*, 2008. 4

[57] R. N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. 2

[58] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6, 8

[59] P. Siva and T. Xiang. Weakly supervised object detector learning with model drift detection. In *ICCV*, 2011. 2

[60] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *Workshop at CVPR*, 2008. 1, 4

[61] Spare5/Mighty AI: https://app.spare5.com/fives. Bounding box drawing instruction video. https://www.youtube.com/watch?v=3SZyFJiMGOw, 2017. 1, 6

[62] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. In *AAAI Human Computation Workshop*, 2012. 1, 2, 3, 4, 6, 7, 8

[63] A. Torralba, A. Oliva, M. Castelhano, and J. M. Henderson. Contextual guidance of attention in natural scenes: The role of global features on object search. *Psychological Review*, 113(4):766–786, 2006. 4

[64] O. Veksler. Star shape prior for graph-cut image segmentation. In *ECCV*, 2008. 3, 5

[65] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008. 3

[66] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *IJCV*, 2013. 4

[67] C. Wang, W. Ren, J. Zhang, K. Huang, and S. Maybank. Large-scale weakly supervised object localization via latent category learning. *IEEE Transactions on Image Processing*, 24(4):1371–1385, 2015. 2

[68] J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV*, 2005. 2

[69] T. Wang, B. Han, and J. Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *CVIU*, 2014. 3

[70] P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In *NIPS*, 2010. 4

[71] J. Wu, Y. Zhao, J.-Y. Zhu, S. Luo, and Z. Tu. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *CVPR*, 2014. 3, 5, 8

[72] W. Yang, J. Cai, J. Zheng, and J. Luo. User-friendly interactive image segmentation through unified combinatorial user inputs. *IEEE Transactions on Image Processing*, 2010. 3