This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Training object class detectors with click supervision

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

Training object class detectors typically requires a large set of images with objects annotated by bounding boxes. However, manually drawing bounding boxes is very time consuming. In this paper we greatly reduce annotation time by proposing center-click annotations: we ask annotators to click on the center of an imaginary bounding box which tightly encloses the object instance. We then incorporate these clicks into existing Multiple Instance Learning techniques for weakly supervised object localization, to jointly localize object bounding boxes over all training images. Extensive experiments on PASCAL VOC 2007 and MS COCO show that: (1) our scheme delivers high-quality detectors, performing substantially better than those produced by weakly supervised techniques, with a modest extra annotation effort; (2) these detectors in fact perform in a range close to those trained from manually drawn bounding boxes; (3) as the center-click task is very fast, our scheme reduces total annotation time by $9\times$ *to* $18\times$.

1. Introduction

How can we train high-quality computer vision models with minimal human annotation effort? Obtaining training data is especially costly for object class detection, the task of detecting all instances of a given object class in an image. Typically, detectors are trained under full supervision, which requires manually drawing tight object bounding boxes in a large number of training images. This takes time: annotating the popular ILSVRC dataset [52] required about 35s per bounding box, using a crowd-sourcing technique optimized for efficient bounding box annotation [66] (more details in Sec. 2).

Object detectors can also be trained under weak supervision using only image-level labels. While this is substantially cheaper, the resulting detectors typically deliver only about half the accuracy of their fully supervised counterparts [6, 7, 8, 11, 13, 29, 54, 61, 62, 63, 75]. In this paper, we aim to minimize human annotation effort while producing high-quality detectors. To this end we propose annotating objects by clicking on their center.

Clicking on an object can be seen as the human-computer-interaction equivalent of pointing to an object. Pointing is a natural way for humans to communicate that emerges early during cognitive development [69]. Human pointing behavior is well-understood in human-computer interaction, and can be modeled mathematically [65]. For the purpose of image annotation, clicking on an object is therefore a natural choice. Clicking offers several advantages over other ways to annotate bounding boxes: (1) is substantially faster than drawing bounding boxes [66], (2) requires little instructions or annotator training compared to drawing [66] or verifying bounding boxes [46, 53, 66], because it is a task that comes natural to humans, (3) can be performed using a simple annotation interface (unlike bounding box drawing [66]), and requires no specialized hardware (unlike eye-tracking [45]). Note that the scheme we propose does not require a human-in-the-loop setup [12, 46, 47, 72, 24]: clicks can be acquired separately, independently of the detector training framework used.

Given an image known to contain a certain object class, we ask annotators to click on the center of an imaginary bounding box enclosing the object (*center-click* annotations). These clicks provide reliable anchor points for the full bounding box, as they provide an estimate of its center. Moreover, we can also ask two different annotators to provide center-clicks on the same object. As their errors are independent, we can obtain a more accurate estimate of the object center by averaging their click positions. Interestingly, given the two clicks, we can even estimate the *size* of the object, by exploiting a correlation between the object size and the distance of the click to the true center (error). As the errors are independent, the distance between the two clicks increases with object size. This enables to estimate size based on the distance between the clicks. As a novel component of our crowd-sourcing protocol, we introduce a stage to train the annotators based on synthetic polygons. This enables generating an arbitrarily large set of training questions without using any manually drawn bounding box. Moreover, we derive models of the annotator error directly from this polygon stage, and use them later to estimate object size in real images.

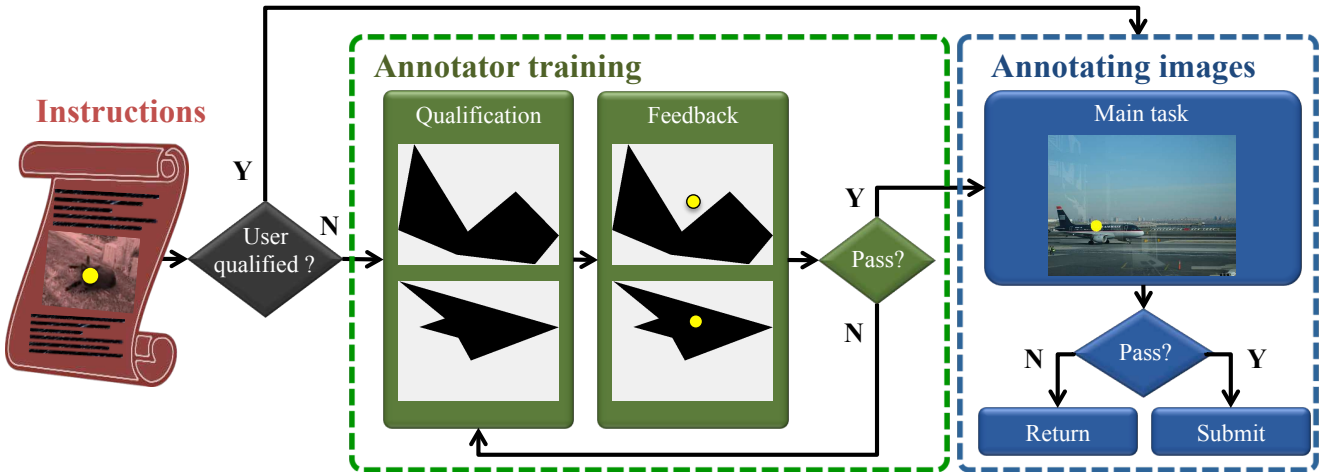

Figure 1. **The workflow of our crowd-sourcing framework for collecting click annotations.** The annotators read a set of instructions and then go through an interactive training stage that consists of a simple qualification test based on synthetic polygons. After completing it, they receive a detailed feedback on how well they performed. Annotators who successfully pass the qualification test can proceed to the annotation stage. In case of failure, they can repeat the test as many times as they want.

We incorporate these clicks into a reference Multiple Instance Learning (MIL) framework which was originally designed for weakly supervised object detection [11]. It jointly localizes object bounding boxes over all training images of an object class. It iteratively alternates between re-training the detector and re-localizing objects. We use the center-clicks in the re-localization phase, to promote selecting bounding boxes compatible with the object center and size estimated based on the clicks.

Based on extensive experiments with crowd-sourced center-clicks on Amazon Mechanical Turk for PASCAL VOC 2007 and simulations on MS COCO, we demonstrate that: (1) our scheme incorporating center-click into MIL delivers better bounding boxes on the training set. In turn, this lead to high-quality detectors, performing substantially better than those produced by weakly supervised techniques, with a modest extra annotation effort (less than 4h on the entire PASCAL VOC 2007 trainval); (2) these detectors in fact perform in a range close to those trained from manually drawn bounding boxes; (3) as the center-click task is very fast, our scheme reduces total annotation time by $9\times$ (one click) to $18\times$ (two clicks); (4) given the same human annotation budget, our scheme outperforms the recent human verification scheme [46], which was already very efficient.

2. Related work

Time to draw a bounding box. The time required to draw a bounding box varies depending on several factors, including the desired quality of the boxes and the particular crowdsourcing protocol used. In this paper, as an authoritative reference we use the protocol of [66] which was used to annotate ILSVRC [52]. It was designed to produce high-quality bounding boxes with minimal human annotation time on Amazon Mechanical Turk, a popular crowd-sourcing platform. They report the following median times for annotating an object class in an image [66]: 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 (in which case the process repeats). Since we only consider localizing one object per class per image, we use $25.5s + 9.0s = 34.5s$ as the reference time for manually annotating a high-quality bounding box. This is a conservative estimate: when taking into account that some boxes are rejected in the second step and need to be re-drawn multiple times until they are correct, the median time increases to 55s. If we use average times instead of medians, the cost raises further to 117s.

We use 34.5s as reference both for PASCAL VOC [17], which has objects of comparable difficulty to ILSVRC [52], and for COCO [39], which is more difficult. Both datasets have high-quality bounding boxes, which we use as reference for comparisons to our method.

Weakly-supervised object localization (WSOL). These methods are trained from a set of images labeled only as containing a certain object class, without being given the location of the objects [6, 7, 8, 10, 13, 29, 54, 61, 62, 63, 75]. The goal is to localize the objects in these training images while learning an object detector for localizing instances in new test images. Recent work on WSOL [6, 7, 8, 10, 29, 62, 63, 75] has shown remarkable progress thanks to Convolutional Neural Nets (CNNs [20, 34]). However, learning a detector without location annotations is difficult and performance is generally about half that of their fully supervised counterparts [6, 7, 8, 10, 13, 29, 54, 61, 62, 63, 75].

WSOL is often addressed as a Multiple Instance Learning (MIL) problem [6, 10, 13, 14, 59, 61, 62, 63]. In this paper, we use MIL as our basis and augment it with center-click supervision.

Figure 2. **Instruction Examples**: (left) the desired box center may not be on the object, (middle) if the object instance is truncated, click on the center of the visible part and (right) if multiple instances are present, click on the center of any one of them.

Click supervision. Click annotation schemes have been used in part-based detection to annotate part locations of an object [9, 74], and in human pose estimation to annotate key-points of human body parts [26, 49, 56]. Click supervision has also been used to reduce the annotation time for semantic segmentation [4, 23, 5, 76]. Recently, Bearman et al. [4] collected clicks by asking the annotators to click anywhere on a target object. In Sec. 5.1, we show that our center-click annotations outperforms these click-anywhere annotations for object class detection. Finally, Mettes et al. [42] proposed to annotate actions in videos with click annotations. Our work also offers other new elements over the above works, e.g. estimating object area from two clicks and training annotators with synthetic polygons.

Other ways to reduce annotation cost. Researchers tried to learn object class detectors from videos, where the spatio-temporal continuity facilitates object localization [28, 38, 48, 36, 68]. An alternative direction is transfer learning, where an appearance model for a new class is learned from bounding box annotations on examples of related classes [3, 18, 21, 22, 35, 37, 50]. Eye-tracking data can be seen as another type of pointing to an object. Such data have been used as a weak supervisory signal to localize objects on images [41, 45] or videos [57, 40].

Recently, Papadopoulos et al. [46] proposed a very efficient framework for training object class detectors that only requires humans to verify bounding boxes produced by the learning algorithm. We compare with [46] in Sec. 5.

3. Crowd-sourcing clicks

We now describe the main components of our crowd-sourcing workflow, which is illustrated in Fig. 1.

3.1. Instructions

Our annotators are given an image and the name of the target class. Unlike [4] where annotators are asked to click anywhere on a target object, we want them to click on the center of an imaginary bounding box around the object (Fig. 2). This definition of center is crucial, as it provides a strong anchor point for the actual bounding box location. However, humans have a tendency to click on the center of mass of the object, which gives a less precise anchor point for the box location. We therefore carefully phrase our instructions as: "imagine a perfectly tight rectangular box around the object and then click as close as possible to the center of this imaginary box". For concave objects, the box center might even lie outside the object (Fig. 2-left).

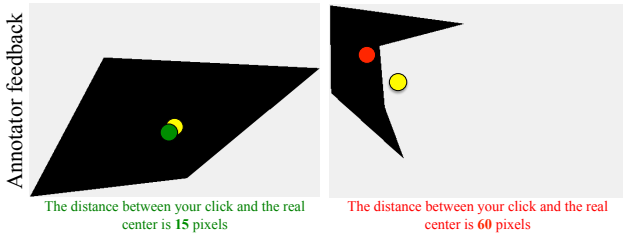

Figure 3. Examples that the annotators receive as feedback. For each example, we provide the real center of the polygon (yellow dot), their click (green or red dot) and the Euclidean distance between the two.

We also include explanations for special cases: If an object is truncated (i.e. only part of it is visible), the annotator should click on the center of the visible part (Fig. 2-middle). If there are multiple instances of the target class, one should click on the center of only one of them (Fig. 2-right).

In order to let annotators know approximately how long the task will take, we suggest a time of 3s per click. This is an upper bound on the expected annotation time that we estimated from a small pilot study.

3.2. Annotator training

After reading the instructions, the annotators go through the training stage. They complete a simple qualification test, at the end of which we provide 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. Qualification tests have been successfully used for enhancing the quality of the crowd-sourced data and filtering out bad annotators and spammers [2, 16, 27, 32, 52, 66]. This happens because some annotators pay little to no attention to the task instructions.

During a qualification test, the annotator is asked to respond on some questions for which the answers are known. This typically requires experts to annotate a batch of examples (in our case draw object bounding boxes). Instead, we use an annotation-free qualification test in which the annotators need to click on the center of 20 synthetically generated polygons, like the ones in Fig. 1. Using synthetic polygons allows us to generate an arbitrarily large set of qualification questions with zero human annotation cost. Additionally, annotators cannot overfit to qualification questions or cheat by sharing answers, which is possible when the number of qualification questions is small.

Why polygons? We use polygons instead of axis-aligned rectangles in order to train the annotators on the difference between the center of mass of an object and the center of the imaginary box enclosing the object. Moreover, polygons provide a more realistic level of difficulty for the qualification test. Finding the center of an axis-aligned rectangle is

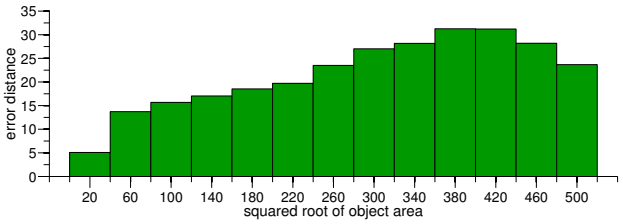

Figure 4. The error distance of the annotators as a function of the square root of the object area.

trivial, whereas finding the center of a polygon is analogous to finding the center of a real object. And yet, polygons are abstractions of real objects, thus reducing the cognitive load on the annotators, potentially making the training stage more efficient.

Feedback. After the annotators finish the qualification test, they receive a feedback page with all polygon examples they annotated (Fig. 3). For each polygon, we display (a) the position of the real center, (b) the position of the annotator's click, and (c) the Euclidean distance in pixels between the two (error distance).

Success or failure. The annotator needs to click close to the real centers of the polygons in order to pass the test. The exact criterion to pass the test is to have an error distance below 20 pixels, on average over all polygons in the test.

The annotators that pass the qualification test are flagged as *qualified annotators* and can proceed to the main annotation task where they work on real images. 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 until they pass it successfully.

The combination of providing rich feedback and allowing annotators to repeat the test results in an interactive and highly effective training stage.

3.3. Annotating images

In the annotation stage, annotators are presented small batches of 20 consecutive images to annotate. For increased efficiency, our batches consist of a single object class. Thanks to this, annotators do not have to re-read the class name for every image, and can keep their mind focused on their prior knowledge about the class to find it rapidly in the image [70]. More generally, it avoids task-switching which is well-known to increase response time and decrease accuracy [51].

Quality control. Quality control is a common process when crowd-sourcing image annotations [4, 31, 39, 52, 55, 64, 66, 73, 77]. We control the quality of click annotation by hiding two evaluation images for which we have ground-truth bounding boxes inside a 20-image batch, and monitor the annotator's accuracy on them (golden questions). Annotators that fail to achieve an accuracy above the threshold set in the qualification test are not able to submit the task. We do not do any post-processing of the submitted data.

Qualification test	Quality control	Error distance
No	No	43.8
images	No	29.4
polygons	No	29.3
polygons	Yes	21.2

Table 1. The influence of the two main elements of our crowdsourcing protocol on click accuracy.

We point out that we use extremely few different golden questions, and add them repeatedly to many batches. On PASCAL VOC 2007, we used only 40, which amounts to 0.5% of the dataset. This is a negligible overhead.

3.4. Data collection

We implemented our annotation scheme on Amazon Mechanical Turk (AMT) and we collected click annotations for all 20 classes of the whole trainval set of PASCAL VOC 2007 [17]. Each image was annotated with a click by two different annotators for each class present in the image. This results in 14,612 clicks in total for the 5,011 trainval images.

Annotation time. During the annotation stage we measure the annotator's response time from the moment the image appears until they click. The mean response time was 1.87s. This indicates that the task can be performed very efficiently by annotators. Note that we are able to annotate the whole PASCAL VOC 2007 trainval set with one click per object class per image in only 3.8 hours.

Interestingly, the response time we measured is comparable to image-level annotation time (1.5s in [33]) indicating that most of the time is spent on the visual search to find the object and not on clicking on it. Also, our requirement to click on the center of the object does not slow down the annotators: our response time is comparable to the time reported in [4] for click-anywhere annotations.

We examined the response time as a function of the area of the target object and we observed an interesting phenomenon. Response time does not increase when the object becomes smaller, ranging from 1.7s for very small objects to 2.2s for object as big as the whole image. We hypothesize that while small objects are more difficult to find, estimating their center is easier than for large objects.

Error analysis. We evaluate the accuracy of the collected clicks by measuring their distance from the true centers of the ground-truth object bounding boxes. In Fig. 4 we show this error distance as a function of the square root of the object area. As expected, the error distance in pixels increases as the object area increases. However, it slightly drops as the object occupies the whole image. This is likely because such images have truncated instances, which means the annotator needs to click in the center of the image rather than the center of the object, an easier task. In general, the error distances are quite low: 19.5 pixels on average with a median of 13.1 pixels (the images are 300x500 on average).

Next, we want to understand the influence of using a qualification test, using quality control, and using polygons

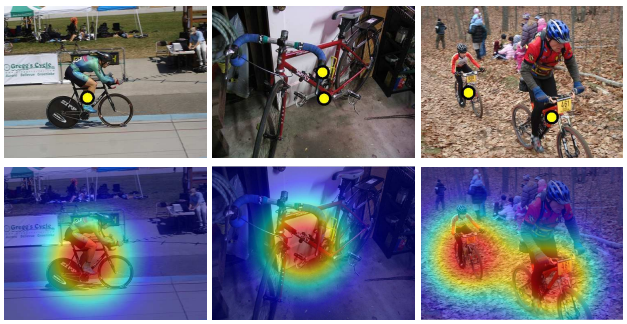

Figure 5. **Box center score** S_{bc} on bicycle examples. (left): One-click annotation. (middle): Two-click annotation on the same instance. (right): Two-click annotation on different instances. The values of each pixel in the heatmaps give the S_{bc} of an object proposal centered at that pixel.

or real examples during the qualification test. Therefore we conducted a series of smaller-scale crowd-sourcing experiments on 400 images of PASCAL VOC 2007 trainval. As Tab. 1 shows, using a qualification test reduces average error substantially, from 43.8 to 29.4 pixels. Interestingly, using polygons instead of real examples does not influence the error at all, demonstrating that our proposed qualification test is well-suited to train annotators. Quality control, hiding two evaluation images inside the task of annotating images, brings the error further down to 21.2 pixels (on the full dataset we measure 19.5 pixels error). Finally, we note that all four variants in Tab. 1 resulted in similar annotation time. Hence qualification tests or quality control has no significant influence on the speed of the annotators.

Cost. We paid annotators $0.10 to annotate a batch of 20 images. Based on their mean response time this results in a wage of about $9 per hour. The total cost for annotating the whole trainval set of PASCAL VOC 2007 with two click annotations was $75.40 (or $37.70 for one click annotation).

4. Incorporating clicks into WSOL

We now present how we incorporate our click supervision into a reference Multiple Instance Learning (MIL) framework, which is typically used in weakly supervised object detection (WSOL). All explanations in this section consider working with one object class at a time, as we treat them essentially independently.

4.1. Reference Multiple Instance Learning (MIL)

The input to MIL is a training set with positive images, which contain the target class, and negative images, which do not. We represent each image as a bag of object proposals extracted using Edge-Boxes [15]. Following [20, 11, 6, 7, 62, 75], we describe each object proposals with a 4096-dimensional feature vector using the Caffe implementation [25] of the AlexNet CNN [34]. We pre-trained the CNN on the ILSVRC [52] dataset using only image-level labels (no bounding box annotations).

A negative image contains only negative proposals, while a positive image contains at least one positive proposal, mixed in with a majority of negative ones. The goal is to find the true positive proposals from which to learn an appearance model for the object class. We iteratively build an SVM appearance model $\mathcal{A}$ by alternating between two steps:

(I) *re-localization*: in each positive image, we select the proposal with the highest score given by the current appearance model $\mathcal{A}$.

(II) *re-training*: we re-train the SVM using the current selection of proposals from the positive images, and all proposals from negative images.

As initialization, in the first iteration we train the classifier using complete images as positive training examples [10, 11, 44, 54, 43, 30].

Refinements. In order to obtain a competitive baseline, we apply two refinements to the standard MIL framework. First, we use multi-folding [11], which helps escaping local optima. Second, we combine the score given by the appearance model $\mathcal{A}$ with a general measure of "objectness" [1] $\mathcal{O}$, which measures how likely it is for a proposal to tightly enclose an *object* of any class (e.g. bird, car, sheep), as opposed to background (e.g. sky, water, grass). Objectness was used in WSOL before, to steer the localization process towards objects and away from background [11, 13, 21, 48, 58, 61, 59, 67, 75]. In this paper we use the recent objectness measure of [15].

Formally, at step (I) we linearly combine the scores $\mathcal{A}$ and $\mathcal{O}$ under the assumption of equal weights. The score of each proposal p is given by $S_{ap}(p) = \frac{1}{2} \cdot \mathcal{A}(p) + \frac{1}{2} \cdot \mathcal{O}(p)$.

Deep MIL. After MIL converges (typically within 10 iterations), we perform two additional iterations where during the step (II) we deeply re-train the whole CNN network, instead of just an SVM on top of a fixed feature representation. During these iterations we use Fast RCNN [19] as the appearance model $\mathcal{A}$.

4.2. One-click supervision

Motivation. Click annotations on object centers derived using our crowdsourcing method of Sec. 3 provide a powerful cue about object position. In this section, we improve the reference MIL framework by using the position of one single click c in each image of the target class.

Box center score S_{bc}**.** Intuitively, simply selecting the object proposal whose center is closest to the click would fail since annotators are not perfectly accurate. Instead, we introduce a score function S_{bc}, which represents the likelihood of a proposal p covering the object according to its center point c_p and the click c

$$S_{bc}(p; c, \sigma_{bc}) = \mathrm{e}^{-\frac{\|c_p - c\|^2}{2\sigma_{bc}^2}} \quad (1)$$

Figure 6. **Box area score** S_{ba}. All windows used here have fixed aspect ratio and are centered on the center of the object.

where $\|c_p - c\|$ indicates the Euclidean distance in pixels between c and c_p. The standard deviation σ_{bc} controls how quickly the S_{bc} decreases as c_p gets farther from c (Fig. 5).

Use in re-localization. We use the box center cue S_{bc} in the re-localization step (I) of MIL (Sec. 4.1). Instead of selecting the proposal with the highest score according to the score function S_{ap} alone, we combine it with S_{bc} with a product: $S_{ap}(p) \cdot S_{bc}(p; c, \sigma_{bc})$. In Sec. 5.1 we show that this results in improved re-localization, which in turn leads to better appearance models in the next re-training iteration, and ultimately improves the final MIL outcome.

Use in initialization. We also use the click position to improve the MIL initialization. Instead of initializing the positive training samples from the complete images, we now construct windows centered on the click while at the same time having maximum size without exceeding the image borders. This greatly improves MIL initialization, especially in cases where the position of the click is close to the image borders.

4.3. Two-click supervision

Motivation. While using two annotator clicks doubles the total annotation time compared to one click, it allows us to estimate the object center even more accurately. Moreover, we can estimate the object area based on the distance between the two clicks.

Box center score S_{bc}. By averaging the positions of two clicks we can estimate the object center more accurately. We simply replace c in Eq. (1) with the average of the two clicks c_1 and c_2.

However, in images containing multiple instances of the target class, the two annotators might click on different instances (Fig. 5, right). To address this, we introduce a distance threshold d_{max} beyond which the clicks are considered to target different instances. In that case, we keep both clicks and use them both in Eq. (1). Formally, if $\|c_1 - c_2\| > d_{max}$, then for each proposal p we use the nearest of the two click to its center c_p.

Box area score S_{ba}. There is a clear correlation between the area of the object and the click's error distance (Fig. 4). As errors made by two annotators are independent, the distance between their two clicks increases as the object area increases (on average). Therefore we estimate the object area based on the distance between the two clicks c_1 and c_2.

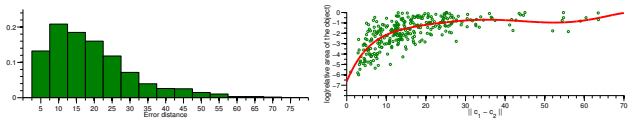

Figure 7. (left) The distribution of errors that the annotators made during our qualification test. (right) The relative area of the synthetic polygons (log scale) as a function of the distance between two clicks. The red line shows the regressed function μ.

Let $\mu(\|c_1 - c_2\|)$ be a function that estimates the logarithm of the object area (we explain how we learn this function in Sec. 4.4). Based on this, for each proposal p we introduce a box area score S_{ba} that represents the likelihood of p covering the object according to the ratio between the proposal area and the estimated object area:

$$S_{ba}(p; c_1, c_2, \sigma_{ba}) = \mathrm{e}^{-\frac{(a_p - \mu(\|c_1 - c_2\|))^2}{2\sigma_{ba}^2}} \quad (2)$$

Here a_p is the logarithm of the proposal's area, and $(a_p - \mu)$ indicates the log ratio between the two areas. The standard deviation σ_{ba} controls how quickly S_{ba} decreases as a_p grows different from μ.

Fig. 6 shows an example of the effect of the S_{ba} score on proposals of various areas. For illustration purposes, all proposals used here have a fixed aspect-ratio and are centered on the object. The score is maximal when the area of the proposal matches the estimated object area.

Use in re-localization. We now use all cues in the final score function for a proposal p during the re-localization step (I) of MIL step:

$$S(p) = S_{ap}(p) \cdot S_{bc}(p; c_1, c_2, \sigma_{bc}) \cdot S_{ba}(p; c_1, c_2, \sigma_{ba}) \quad (3)$$

4.4. Learning score parameters

We exploit the clicks obtained from our qualification task on synthetic polygons to estimate the hyper-parameters of our model: σ_{bc} (Eq. (1)), d_{max} (Sec. 4.3), σ_{ba} (Eq. (2)) and the function μ (Eq. (2)).

Fig. 7-left shows the distribution of the annotators' error distances during our qualification test. We estimate σ_{bc} from this distribution. Also, in the same figure we see that the maximum error distance is 70 pixels, hence we set $d_{max} = 70$. Fig. 7-right shows the logarithm of the relative area of the synthetic polygons as a function of the distance between the two clicks. We learn the function $\mu(\|c_1 - c_2\|)$ as a polynomial regressor fit to this data (red line in Fig. 7-right). Finally, we set σ_{ba} simply as the average error of the area estimation made by the regressor on the polygons.

5. Experimental results

5.1. Results on PASCAL VOC 2007

Dataset. We perform experiments on PASCAL VOC 2007 [17], which has 20 classes, 5,011 training images (trainval), and 4,952 test images. During training we only

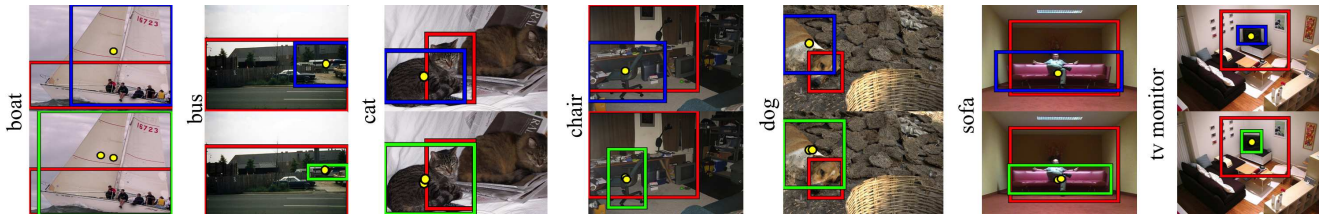

Figure 8. Examples of objects localized on the trainval set of PASCAL VOC 2007 using our one-click (blue) and two-click (green) supervision models. For each example, we also show the localization produced by the reference MIL (red).

use image-level labels. Unlike some previous WSOL work which removes images with truncated and difficult objects [10, 11, 13, 54, 75], we use the complete trainval set.

Object detection model. As object detector we use Fast R-CNN [19]. Instead of Selective Search [71] we use Edge-Boxes [15] as proposals, as they come with an objectness measure [1] which we use inside MIL. Unless stated otherwise, we use AlexNet [34] as the underlying CNN architecture for our method and for all compared methods.

Evaluation. Given a training set with image-level labels (and possibly click annotations), our goal is to localize the object instances in this set and to train good object detectors. We quantify localization performance on the training set with Correct Localization (CorLoc), enabling direct comparison with WSOL methods [6, 7, 8, 11, 13, 29, 54, 61, 75]. CorLoc is the percentage of images in which the bounding-box returned by the algorithm correctly localizes an object of the target class (i.e., IoU ≥ 0.5). We measure the performance of the trained object detector on the test set using mean average precision (mAP). We quantify annotation effort in terms of actual human time measurements.

Compared methods. We compare our approach to the fully supervised alternative by training the same object detector [19] on the same training images, but with manually annotated bounding boxes (one per class per image, for fair comparison). We also compare to a modern MIL-based WSOL technique (Sec. 4.1) run on the same training images, but without click supervision.

For MIL WSOL, the effort to draw bounding boxes is zero. For fully supervised learning we use the actual annotation times for ILSVRC from [66]: 35 seconds for drawing a single bounding box and verifying its quality (Sec. 2). These timings are representative for PASCAL VOC, since their images are of comparable difficulty and quality [52].

We also compare to the human verification scheme [46], using their reported timings, and to various baselines.

Reference MIL. We run the reference MIL WSOL with $k = 10$ folds for 10 iterations, after which it converges. It achieves 43.4% CorLoc on the training set. Applying two deep MIL iterations (Sec. 4.1) on top of this improves to 44.5% CorLoc. The detectors produced by this approach achieve 29.6% mAP on the test set (red dot in Fig. 9).

One-click supervision yields 73.3% CorLoc. The resulting object detector yields 45.9% mAP (yellow dot in Fig. 9). Hence, at a modest extra annotation cost of only 3.8 hours we achieve an absolute improvement of +28.8% CorLoc and +16.3% mAP over the reference MIL.

Two-click supervision doubles the annotation time but it improves our model in two ways: (1) we can estimate the object center more accurately, and (2) we can estimate the object area based on the distance between the two clicks. Using the two-click supervision only to improve the box center estimate S_{bc} brings +0.8% CorLoc and +0.9% mAP over using one-click. Including also the box area estimate S_{ba} leads to a total improvement of +5.2% CorLoc and +3.2% mAP over one-click (78.5% CorLoc and 49.1% mAP, orange dot in Fig. 9). This shows that the box area estimate contributes the most to the improvement brought by two-click over one-click supervision.

State-of-the-art WSOL approaches based on AlexNet architecture [34] perform as follows. Wang et al. [75]: 48.5% CorLoc and 31.6% mAP. Cinbis et al. [11]: 52.0% CorLoc and 30.2% mAP. Bilen et al. [8]: 54.2% CorLoc and 34.5% mAP. Our two-click supervision outperforms all these methods with 78.5% CorLoc and 49.1% mAP, at a modest extra annotation cost.

Full supervision achieves 55.5% mAP. Our two-click supervision comes remarkably close (49.1% mAP). Importantly, full supervision requires 71 hours of annotation time. Instead, our two-click approach requires only 7.6 hours, a reduction of $9\times$ (or $18\times$ for our one-click approach).

Human verification [46] is shown as the blue line in Fig. 9. Given the same total annotation time, our one-click method delivers higher CorLoc and mAP. When we use two-click annotations, given the same annotation effort we match their mAP and get slightly higher CorLoc.

Deeper CNN. When using VGG16 [60] instead of AlexNet, the fully supervised training leads to 65.9% mAP. Our two-click model achieves 57.5% mAP, while the reference MIL WSOL delivers 32.4% mAP.

Effect of click accuracy. We compare the center-click annotations we collected (Sec. 3) to three alternatives: *(oracle clicks)*: use the centers of the ground-truth boxes as clicks; *(random clicks)*: uniformly sample a pixel inside a ground-truth box; *(click-anywhere)*: we simulate a scenario where humans are instructed to click anywhere on the object, by mimicking the distribution of the publicly available click annotations of [4] on PASCAL VOC 2012. We measure the distances from the centers of the ground-truth boxes to their clicks. Then we build a regressor to predict this distance

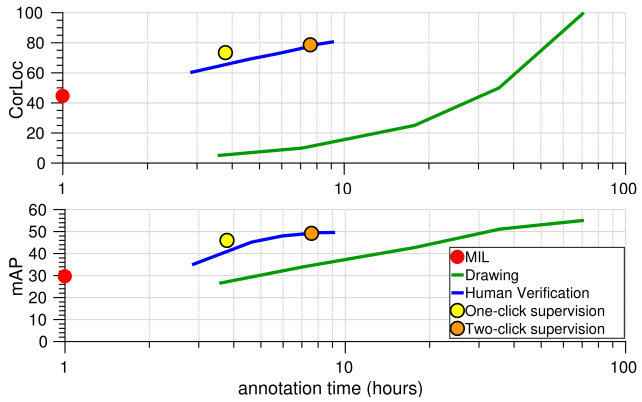

Figure 9. **Evaluation on PASCAL VOC 2007.** CorLoc and mAP performance against human annotation time in hours(log-scale).

based on the area of the object. Finally, we apply this regressor on VOC 2007 and displace the ground-truth object centers by the predicted distance.

For simplicity we use the alternative clicks into our one-click supervision model (Sec. 4.2) in one additional re-localization iteration at the end of the reference MIL (as opposed to using it in every iteration). For each of the three alternatives, we use the oracle best value of the parameter σ_{bc}, while for our center-click annotations we use the one learned on the synthetic polygons (sec. 4.4). As a reference, when used on top of MIL this way, our center-clicks lead to 67.2% CorLoc. Oracle clicks give an upper bound of 73.7% CorLoc, while random clicks on the object do not improve over MIL (43.4% CorLoc). Finally, the click-anywhere scenario achieves 55.5% CorLoc. Interestingly, using our center-clicks leads to +11.7% CorLoc, which shows that they convey more information.

5.2. Results on MS COCO

Dataset. The MS COCO dataset [39] is more difficult than PASCAL VOC, as demonstrated in [39], featuring smaller objects on average, and also more object classes (80). We use exactly the same evaluation setup as for PASCAL VOC 2007 and evaluate CorLoc on the training set (82,783 images) and mAP on the val set (40,504 images).

Reference MIL. The reference MIL WSOL achieves 24.2% CorLoc and 8.9% mAP (red dot in Fig. 10). This is considerably lower than its performance on PASCAL VOC 2007.

Click supervision. We did not collect real click annotations for COCO, but instead simulated them. As we want to create a realistic scenario close to real annotators clicks, we did not use the centers of the available ground-truth boxes as simulated clicks. Assuming the annotator's error distance only depends on the object area, we use the findings of our error analysis on PASCAL VOC 2007 (Fig. 4) to generate realistic noisy simulated clicks for COCO.

Our simulated one-click supervision approach achieves double the performance of reference MIL, reaching 51.8% CorLoc and 18.3% mAP (yellow dot in Fig. 10). Our simulated two-click supervision approach goes even beyond that, with 58.6% CorLoc and 19.3% mAP (orange dot in Fig. 10). Assuming the same annotation time per click as in PASCAL VOC 2007, the total annotation time for one-click is 125 hours.

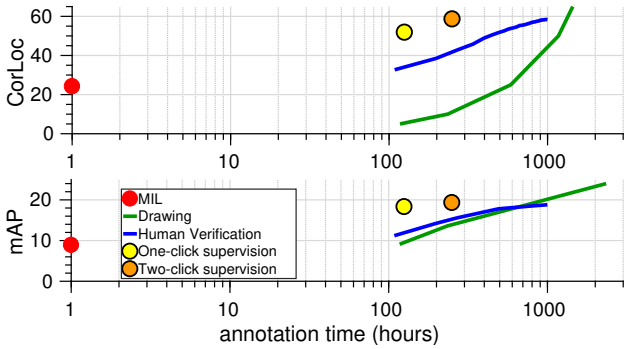

Figure 10. **Evaluation on MSCOCO.** CorLoc and mAP performance against human annotation time in hours(log-scale).

Full supervision. Training with full supervision requires 2,343 hours of annotation time and leads to 24.0% mAP.

Human verification [46]. As [46] do not perform experiments on COCO, we simulate their verification responses by sampling them according to the error distribution of actual humans they report on VOC. This creates a realistic simulation. The CorLoc and mAP of this scheme can be seen in Fig. 10 (blue lines). Our two-click supervision approach reaches about the same CorLoc as the simulated [46] (58.3%) and it performs a bit better in terms of mAP (19.3% vs 18.8%). Importantly, it takes about 3.5× less total annotation time. From another perspective, given the same annotation time (250 hours), our two-click supervision approach outperforms the human verification one by +16% CorLoc and +4% mAP. Hence, on difficult datasets with small objects our method has an edge, as the efficiency of [46] degrades, while the benefits of click supervision remain.

6. Conclusions

We proposed center-click annotation as a way of training object class detectors and showed that crowd-sourced annotators can perform this task accurately and fast (1.9s per object). In extensive experiments on PASCAL VOC and MS COCO we have shown that our center-click scheme dramatically improves over weakly supervised learning of object detectors, at a modest additional annotation cost. Moreover, we have shown that it reduces total annotation time by 9×-18× compared to manually drawing bounding boxes, while still delivering high-quality detectors. Finally, we have shown that our scheme compares favorably against a recent method where annotators verify automatically proposed bounding boxes [46].

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

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 5, 7

[2] 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

[3] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011. 3

[4] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 3, 4, 7

[5] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. In *CVPR*, 2015. 3

[6] H. Bilen, M. Pedersoli, and T. Tuytelaars. Weakly supervised object detection with posterior regularization. In *BMVC*, 2014. 1, 2, 5, 7

[7] H. Bilen, M. Pedersoli, and T. Tuytelaars. Weakly supervised object detection with convex clustering. In *CVPR*, 2015. 1, 2, 5, 7

[8] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 1, 2, 7

[9] S. Branson, P. Perona, and S. Belongie. Strong supervision from weak annotation: Interactive training of deformable part models. In *ICCV*, 2011. 3

[10] R. Cinbis, J. Verbeek, and C. Schmid. Multi-fold mil training for weakly supervised object localization. In *CVPR*, 2014. 2, 5, 7

[11] R. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE Trans. on PAMI*, 2016. 1, 2, 5, 7

[12] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, 2013. 1

[13] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, 2010. 1, 2, 5, 7

[14] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997. 2

[15] P. Dollar and C. Zitnick. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 5, 7

[16] 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

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 2, 4, 6

[18] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *cviu*, 2007. 3

[19] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 5, 7

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2, 5

[21] M. Guillaumin and V. Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *CVPR*, 2012. 3, 5

[22] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, and J. Donahue. LSDA: Large scale detection through adaptation. In *NIPS*, 2014. 3

[23] 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

[24] S. D. Jain and K. Grauman. Active image segmentation propagation. In *CVPR*, 2016. 1

[25] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013. 5

[26] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 3

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

[28] V. Kalogeiton, V. Ferrari, and C. Schmid. Analysing domain shift factors between videos and images for object detection. *IEEE Trans. on PAMI*, 2016. 3

[29] V. Kantorov, M. Oquab, M. Cho, and I. Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 2016. 1, 2, 7

[30] G. Kim and A. Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *NIPS*, 2009. 5

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

[32] 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

[33] R. A. Krishna, K. Hata, S. Chen, J. Kravitz, D. A. Shamma, L. Fei-Fei, and M. S. Bernstein. Embracing error to enable rapid crowdsourcing. In *CHI*, 2016. 4

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2, 5, 7

[35] D. Kuettel, M. Guillaumin, and V. Ferrari. Segmentation Propagation in ImageNet. In *ECCV*, 2012. 3

[36] 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. 3

[37] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 3

[38] C. Leistner, M. Godec, S. Schulter, A. Saffari, and H. Bischof. Improving classifiers with weakly-related videos. In *CVPR*, 2011. 3

[39] 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, 4, 8

[40] S. Mathe and C. Sminchisescu. Dynamic eye movement datasets and learnt saliency models for visual action recognition. In *ECCV*, 2012. 3

[41] S. Mathe and C. Sminchisescu. Action from still image dataset and inverse optimal control to learn task specific visual scanpaths. In *NIPS*, 2013. 3

[42] P. Mettes, J. C. van Gemert, and C. G. Snoek. Spot on: Action localization from pointly-supervised proposals. In *ECCV*, 2016. 3

[43] M. Nguyen, L. Torresani, F. de la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *ICCV*, 2009. 5

[44] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 5

[45] D. P. Papadopoulos, A. D. F. Clarke, F. Keller, and V. Ferrari. Training object class detectors from eye tracking data. In *ECCV*, 2014. 1, 3

[46] 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. 1, 2, 3, 7, 8

[47] A. Parkash and D. Parikh. Attributes for classifier feedback. In *ECCV*, 2012. 1

[48] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 3, 5

[49] D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, 2006. 3

[50] M. Rochan and Y. Wang. Weakly supervised localization of novel objects using appearance transfer. In *CVPR*, 2015. 3

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

[52] 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, 5, 7

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

[54] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *ECCV*, 2012. 1, 2, 5, 7

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

[56] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *CVPR*, 2013. 3

[57] K. Shanmuga Vadivel, T. Ngo, M. Eckstein, and B. Manjunath. Eye tracking assisted extraction of attentionally important objects from videos. In *CVPR*, 2015. 3

[58] N. Shapovalova, A. Vahdat, K. Cannons, T. Lan, and G. Mori. Similarity constrained latent support vector machine: An application to weakly supervised action classification. In *ECCV*, 2012. 5

[59] Z. Shi, P. Siva, and T. Xiang. Transfer learning by ranking for weakly supervised object annotation. In *BMVC*, 2012. 2, 5

[60] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 7

[61] P. Siva and T. Xiang. Weakly supervised object detector learning with model drift detection. In *ICCV*, 2011. 1, 2, 5, 7

[62] H. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darell. On learning to localize objects with minimal supervision. In *ICML*, 2014. 1, 2, 5

[63] H. Song, Y. Lee, S. Jegelka, and T. Darell. Weakly-supervised discovery of visual pattern configurations. In *NIPS*, 2014. 1, 2

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

[65] R. W. Soukoreff and I. S. MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies*, 61(6):751–789, 2004. 1

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

[67] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014. 5

[68] K. Tang, R. Sukthankar, J. Yagnik, and L. Fei-Fei. Discriminative segment annotation in weakly labeled video. In *CVPR*, 2013. 3

[69] M. Tomasello, M. Carpenter, and U. Liszkowski. A new look at infant pointing. *Child development*, 2007. 1

[70] 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

[71] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013. 7

[72] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *IJCV*, 108(1-2):97–114, 2014. 1

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

[74] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011. 3

[75] 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. 1, 2, 5, 7

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

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