In David Fleet, Tomas Pajdla, Bernt Schiele, Tinne Tuytelaars, eds., *Proceedings of 13th European Conference on Computer Vision,* Part V, 361–376. Zurich.

Training object class detectors from eye tracking data

Dim P. Papadopoulos, Alasdair D. F. Clarke, Frank Keller, Vittorio Ferrari

School of Informatics, University of Edinburgh

Abstract. Training an object class detector typically requires a large set of images annotated with bounding-boxes, which is expensive and time consuming to create. We propose novel approach to annotate object locations which can substantially reduce annotation time. We first track the eye movements of annotators instructed to find the object and then propose a technique for deriving object bounding-boxes from these fixations. To validate our idea, we collected eye tracking data for the trainval part of 10 object classes of Pascal VOC 2012 (6,270 images, 5 observers). Our technique correctly produces bounding-boxes in 50% of the images, while reducing the total annotation time by factor $6.8\times$ compared to drawing bounding-boxes. Any standard object class detector can be trained on the bounding-boxes predicted by our model. Our large scale eye tracking dataset is available at `groups.inf.ed.ac.uk/calvin/eyetrackdataset/`.

1 Introduction

Object class detection is the task of predicting a bounding-box around each instance of an object class in a test image. Traditionally, training an object class detector requires a large set of images in which objects are manually annotated with bounding-boxes [8, 14, 15, 47, 50] (Fig. 1). Bounding-box annotation is time consuming and expensive. The authors of [20] report a 26s median time to draw a bounding-box during a large-scale annotation effort for ImageNet by crowd-sourcing on Mechanical Turk. Additionally, detailed annotation guidelines, annotator training based on these guidelines, and manual checking of the annotation are typically required [14, 20]. Annotating large sets of images is therefore an enormous undertaking, which is typically supported by crowd sourcing [1, 13].

In this paper, we propose a novel approach to training object detectors which can substantially reduce the time required to annotate images. Instead of carefully marking every training image with accurate bounding-boxes, our annotators only need to find the target object in the image, look at it, and press a button. By tracking the eye movements of the annotators while they perform this task, we obtain valuable information about the position and size of the target object (Fig. 1). Unlike bounding-box annotation, our eye tracking task requires no annotation guidelines and can be carried out by completely naive viewers. Furthermore, the task can be performed in a fraction of the time it takes to draw a bounding-box (about one second per image, Sec. 3).

Eye movement behavior in scene viewing has been the subject of a large body of research in cognitive science [22]. Experimental results indicate that human observers prefer to fixate objects, rather than the background of an image [12]. This tendency is

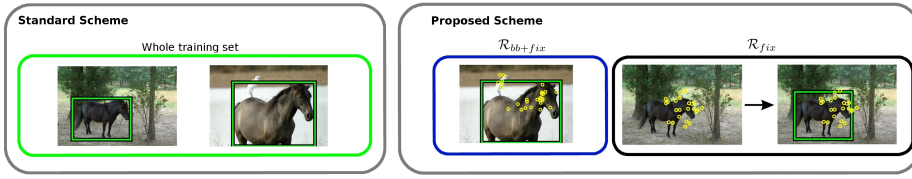


Fig. 1. *(Left) The standard approach to training an object detector, where all images in the training set are manually annotated with bounding-boxes. (Right) In our approach most of the images are annotated only with human eye fixations ($\mathcal{R}_{fix}$ set). The key idea is to automatically predict bounding-boxes for the images in $\mathcal{R}_{fix}$ set given only their fixations. For this we first train a model to infer the spatial support of objects from the fixations, on a small subset of images annotated by both fixations and bounding-boxes ($\mathcal{R}_{bb+fix}$, only 7% of the images in our experiments).*

particularly pronounced in visual search, as finding an object typically requires fixating it [51]. This strongly suggests that eye tracking data can be useful for training a system to automatically localize objects. However, fixation data only provides a rough indication of the spatial extent of an object: humans have a tendency to fixate the center of an object [33], and within animate objects (humans and animals), there is a strong preference for fixating faces [24]. Furthermore, the eye movement record will include fixations on salient non-target objects, and occasionally on the background.

Inspired by the above observations, we propose a technique for deriving a bounding-box covering the whole target object given the fixations on the image (Sec. 4). We cast bounding-box estimation as a figure-ground segmentation problem. We define a model that takes human eye movement data as input and infers the spatial support of the target object by labeling each pixel as either object or background. As the relation between fixations and bounding-boxes can be complex and vary between object classes, we train our model from a small subset of images annotated with bounding-boxes *and* fixations. The subset is only a small fraction of the dataset we want to annotate (7% of the images in our experiments). Once trained on this subset, we use our model to derive bounding-boxes from eye movement data for the whole dataset, which can then be used to train any standard object class detector, such as the Deformable Parts Model [15].

To test our hypothesis that eye tracking data can reduce the annotation effort required to train object class detectors, we collected eye tracking data for the complete training set of ten objects classes from Pascal VOC 2012 [13] (6,270 images in total). Each image is annotated with the eye movement record of five participants, whose task was to identify which object class was present in the image (Sec. 3). This dataset is publicly available at `groups.inf.ed.ac.uk/calvin/eyetrackdataset/`.

We demonstrate through extensive experiments on this dataset that our segmentation model can produce accurate bounding-boxes in about half the images, and that a modern object detector [15] can be trained from them (Sec. 5).

2 Related Work

Eye movement data in computer vision. Researchers in cognitive science have a long-standing interest in computational models that predict human eye movement behavior (e.g. [21, 23, 24]). Recently, however, a number of authors have started to use eye movement data for computer vision tasks. This includes work on image segmentation, which shows that using fixation data to help segmentation algorithms leads to improved performance [32, 37, 48]. Eye tracking data is also useful for face and text detection: [25] cluster fixations to find regions in which the targets are likely to be found, and then apply standard face and text detectors only there. Several authors have collected eye tracking data for video and shown that saliency maps computed from these data can improve action recognition [31, 46].

Yun et al. [52] collect eye movement data for a 1,000 image subset of Pascal VOC 2008; three observers performed a three second free-viewing task. This data is then used to to re-rank the output of an object class detector [15] on test images. Our dataset is substantially larger (6,270 images, 5 observers), and our observers perform a visual search task, which is faster (one second per image) and more likely to result in fixations on the target objects (Sec. 3). Most importantly, we present a method for using eye tracking data to *train* an object class detector, replacing ground-truth bounding-boxes, rather than using them for post-processing at test time.

Weakly supervised learning. Our research is related to work trying to reduce the annotation effort required to train object class detectors. Weakly supervised methods train from a set of images labeled only as containing a certain object class, without location annotation [7, 9, 16, 34, 40, 41, 45]. These methods try to locate the object by searching for patterns of appearance recurring across the training images. However, learning a detector without location annotation is very difficult and performance is still below fully supervised methods [9, 34, 39]. Recently a few authors have tackled learning object class detectors from video, where the temporal continuity facilitates the task [28, 36, 42]. Another line of work leverages text naturally occurring near an image as a weak annotation for its contents [3, 11, 19, 30], such as on web pages or newspapers. This form of supervision has been particularly successful for learning faces of specific people [3], in part because excellent generic face detectors are already available [47].

Our work provides a different way to reduce annotation effort which is complementary to those above. It could potentially be integrated with some of them for even greater savings.

3 Eye Tracking Dataset

3.1 Materials

The images in our dataset were taken from the 2012 edition of the Pascal VOC challenge [13]. We selected 10 of the 20 Pascal classes and included all trainval images for these classes in our dataset. We grouped our 10 classes into pairs as follows (see below for explanation): cat/dog, bicycle/motorbike, boat/aeroplane, horse/cow, sofa/diningtable.

(a) (b) (c)

Fig. 2. *Examples of eye tracking data for three images of class motorbike. We show the sequence of fixations (circles) for five participants (color coded). Note how the first fixation falls outside the image (see main text for details).*

For each pair of classes, we removed all images that contained objects of both classes (e.g. images containing both horses and cows). This eliminated 91 images, or 1.4% of total, and resulted in a dataset containing 6,270 images.

3.2 Procedure

As explained in Sec. 2, results in the visual cognition literature indicate that free viewing may not be the optimal task for collecting eye tracking data for training automatic object detectors. A visual search task, in contrast, increases the likelihood that participants fixate the target object, as this facilitates finding the target and help complete the task correctly.

However, traditional visual search tasks require a large number of target-absent trials. For example, if the task is to search for horses (the participants presses a "yes" button if a horse is present in the image, and "no" otherwise), then the set of images shown to the participant needs to contain images without horses (typically 50%, to minimize guessing). Such a setup would mean that eye tracking data is collected for a large number of target-absent images, which then cannot be used to train a detector.

We therefore used the related task of *two-alternative forced choice object discrimination*, in which each image contains instances of one of two object classes (e.g. cow or horse), and participants have to press one of two buttons to indicate which class is present. This way, visual search data can be collected for two classes at a time, without the need for target-absent images. In adopting this approach, we paired object classes carefully, such that the two sets of images were similar in size (to minimize guessing), and such that the objects were easily confusable (similar size and background, etc.), as otherwise the task is too easy for human observers.

3.3 Apparatus

The experiment was conducted in a sound-attenuated room; participants were seated 60 cm from a 22" LCD screen (Samsung SyncMaster 2233, 100 Hz refresh rate, 5 ms response time) while their eye movements were recorded using an Eyelink 2000 eye tracker (SR Research Ltd., Ottawa), which sampled both eyes at a rate of 1,000 Hz, with

a typical spatial resolution of 0.25° to 0.5°. A head rest was used to minimize participants head movements and improve recording accuracy. Button presses were recorded using a Logitech gamepad that offers millisecond accuracy.

The experiment was controlled using Psychophysics Toolbox Version 3 [6]. Data collection started with a standard nine-point calibration and validation. As explained below, participants viewed images blocked by pairs of classes; the images within a block were presented in random order (a new sequence was generated for each participant). Each trial started with a central fixation cross, displayed for 500 ms, after which the image was displayed. The participant's task was to press one of the two response buttons to indicate the class to which the object(s) in the image belonged, after which the next trial was started automatically. Drift correction was performed after every 20 images; re-calibration was performed if the correction showed that this was necessary (after approximately every 200 images). Participants were offered a five minute break every 30 minutes.

The images in the Pascal dataset differ in size, and they are all smaller than the screen resolution used for the experiment (1,680 $\times$ 1,050 pixels). Instead of re-scaling the images to this resolution (which would result in visual artifacts and make the task unnatural), we presented the images in their original size, but at random offset from the center of the screen. This has the advantage that participants cannot easily develop a viewing strategy (e.g. always looking at the center of the screen, looking in the upper half), thus ensuring that we obtain maximally informative eye movement data.

3.4 Participants

A total of 28 participants (11 male) took part in the data collection, all students at the University of Edinburgh. They gave informed consent and were paid £10 per hour. The materials were divided into seven blocks of around 1,000 images each, where each block contained of all the images in one pair of classes, except for one large pairs of classes (cat/dog), which was split across three blocks. Each participant saw all the images in one block, except for five participants, who saw between two and four blocks in multiple sessions. Blocks were distributed across participants such that every image in every block was seen by five distinct participants.

3.5 Results

A total of around 178,000 fixations were collected, corresponding a mean of 5.7 fixations per participant per image. For example images with fixations superimposed, see Fig. 2. Note that we removed the first fixation of each trial, as it almost always falls on the location of the fixation cross, and thus is uninformative (as well as often being outside the image, due to the off-center presentation).

The mean response time per image (the time from the onset of the image to the button press) was 889 ms, ranging from 786 ms for cat 1090 ms for cow. This indicates that the task can be performed very efficiently by human observers, especially compared to the 26 seconds reported in [20] as the time required to draw a bounding-box. Participants were nevertheless highly accurate at the task, with a mean discrimina-

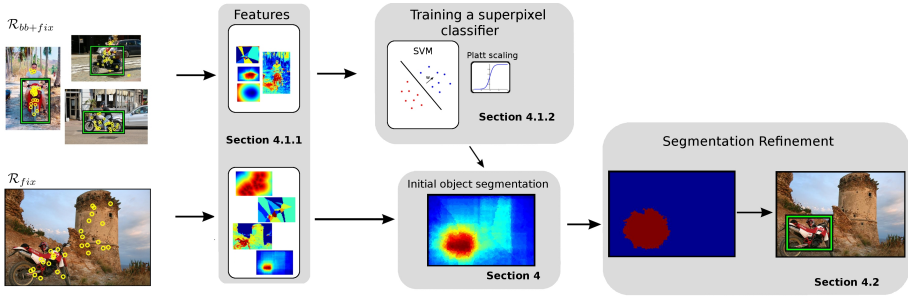


Fig. 3. *Illustration of our method for predicting bounding-boxes from fixations. See sec. 4.*

tion accuracy (percentage of correct button presses) of 95.2%, ranging from 92.9% for sofa/diningtable to 96.1% for boat/aeroplane.

We then compared the positions of the human fixations with the locations of the ground-truth bounding-boxes provided with the Pascal dataset. On average, 75.2% of all fixations on an image fell within one of the bounding-boxes for that image, ranging from 57.3% for boat to 89.6% for cat. This provides initial evidence of our claim that fixation data is useful for localizing objects, though it also indicates that there is considerable inter-class variability in fixation behavior.

4 From fixations to bounding-boxes

We present here our method for localizing objects in an image given fixations (Fig. 1). We model this problem as figure-ground segmentation. Our model takes as input human fixations Φ for an image I in $\mathcal{R}_{fix}$ and infers the spatial support of the object by labeling each pixel as either object or background. The method has two stages (Fig. 3):

I) **Initial object segmentation** (Sec. 4.1) The first stage predicts an initial estimate of the object position by labeling each (super-)pixel individually. This predictor captures the relation between fixations and object positions. It is trained on the image set $\mathcal{R}_{bb+fix}$, which is annotated with both fixations and manual bounding-boxes. The output of this stage is a values for each pixel of I that corresponds to the probability to be on the object.

II) **Segmentation refinement** (Sec. 4.2). The second stage refines the segmentation with an energy model similar to GrabCut [38]. This includes pairwise dependencies between neighboring superpixels, acting as a prior preferring spatially smooth segmentations.

4.1 Initial object segmentation.

We describe here the first stage of deriving object segmentations from fixations. We use an SVM to classify superpixels into object or background based on a diverse set of features computed from the fixations, such as the distance between a superpixel and

the nearest available fixation. Before the classifier is ready to label superpixels in a new image, we train it on the set $\mathcal{R}_{bb+fix}$. In this fashion it can learn a relation between fixations and object positions specific for that object class (i.e. the relation between these features and the fact that a superpixel is on the object or not). After training, the classifier is applied to each image $I \in \mathcal{R}_{fix}$, resulting in a soft segmentation mask M (Fig. 3). Each pixel value in M corresponds to the estimated probability for it to be on the object.

Features. We start by over-segmenting each image I into superpixels S using the Turbopixels method [29]. Operating at the superpixel level greatly reduces the computational cost and memory requirements of our technique.

Let Φ be the set of fixations in the image. Each fixation is determined by four values: the (x, y) position, the duration and the rank of the fixation in chronological order.

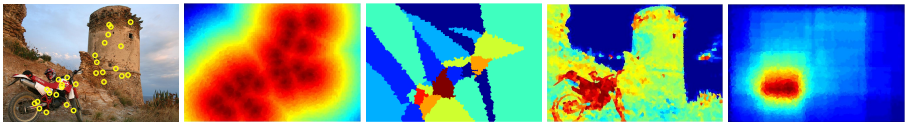
Fixation position features. As mentioned in Sec. 3, we acquired fixations under a visual search task instead of the usual free-viewing task to increase the proportion of fixations on (or near) the target object. Therefore, the position of the fixations directly relates to the position of the object. This motivates us to construct good features indicating whether a superpixel s lies inside the object based on its position relative to the fixations (Fig. 5a):

- *mean distance:* the distance between the center of s and the mean position of all fixations in the image.
- *near distance:* the distance between the center of s and the position of the fixation nearest to s (Fig. 4b).
- *mean offsets:* the vertical and horizontal difference between the center of s and the mean position of all fixations in the image.
- *near offsets:* the vertical and horizontal difference between the center of s and the position of the fixation nearest to s.

Fixation timing features. In addition to the position of each fixation, the eye tracker also delivers timing information, such as the duration and rank of each fixation. These properties carry valuable information. Intuitively, the longer a fixation lasts, the more significant it is. Moreover, in most images the first few fixations will not fall on the target object yet, as the annotator is searching for it, while later fixations are likely to be on the target (Fig. 4c).

- *duration:* the duration of the fixation nearest to s.
- *rank:* the rank of the fixation nearest to s.

Fixation appearance features. The features considered so far support learning a *fixed* spatial relation between the fixations and the superpixels on the object. For example, we could learn that superpixels within a certain distance of the nearest fixation are likely



(a) (b) (c) (d) (e)

Fig. 4. *(a) The positions of the fixations in an image. (b) The* near distance *feature. (c) The* rank *feature. (d) The* appearance *feature according to the distance of the nearest fixation. (e) The* objectness *feature.*

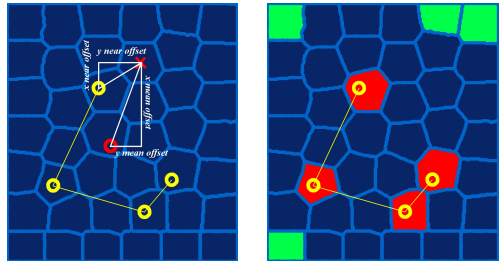
to be on the object. Or we could learn that most of the object mass is below the mean fixation position (e.g. for the person class, as faces receive most fixations).

However, in images of natural objects this spatial relation might change from image to image and therefore might be hard to recover when reasoning about image coordinates alone. For instance, animals can appear in a wide range of viewpoint and deformations, but viewers tend to fixate mostly their heads. This makes it difficult to guess the full extent of their body based purely on the fixation positions.

Here we consider another family of features based on the *appearance* of superpixels (i.e. their color distribution). The key idea is that, while the fixations do not cover the whole object, many of them are on the object, and so provide examples of how it looks. Hence, we can learn about object appearance from a few superpixels hit by fixations. Analogously, we can learn about background appearance from superpixels far from fixations. We can then transfer this knowledge to all superpixels in the image. For example we might find out that a horse is brown and the background is green, even from just a few fixations, and use this knowledge to segment out the whole horse (Fig. 1). Note that this idea works regardless of the spatial relation between the shape and size of the horse in the image and the fixation positions. It effectively creates a mapping from fixation to segmentations that *adapts* to the contents of the target image.

More precisely, we estimate two Gaussian Mixture Models (GMM), one for the object and one for the background. Each GMM has 5 components, each of which is a full-covariance Gaussian over the RGB color space. We estimate the object GMM A_{obj} from all pixels inside all superpixels hit by any fixation, as these are likely on the object (Fig. 5b). Selecting background sample pixels is harder as the converse relation does not hold: the fact that a superpixel is not hit by any fixation does not reveal much about it being on the background or the object (in fact nearly all superpixels are not hit by a fixation, as fixations are extremely sparse). Therefore, we sample superpixels according to three criteria, leading to three different background GMM models A_{bg}, and we leave it to the learner to decide how to best weight them: (1) sample proportionally to the distance to the mean fixation, resulting in many samples far from the mean fixation and less and less samples as we get nearer; (2) sample proportionally to the distance to the nearest fixation. This is simply an alternative way to express 'far from the expected object location'; (3) sample inversely proportionally to the objectness probability (see next paragraph).

After estimating the appearance models A_{obj}, A_{bg}, we use them to evaluate every superpixel s in the image, resulting in per-pixel likelihoods of object/background



(a) (b)

Fig. 5. *Feature extraction.* (a) *Fixation position features. The yellow circles indicate the positions of the fixations in the image, while the red circle indicates the mean position of all fixations. The figure shows the distances from a superpixel center (red cross) that are required to compute all the feature position features.* (b) *Selecting foreground and background superpixels to estimate appearance models for the fixation appearance features. The selected foreground superpixels are indicated in red, while the sampled background superpixels according to the distance from the nearest fixation are in green.*

$p(s|obj) = A_{obj}(s), p(s|bg) = A_{bg}(s)$. We combine these likelihoods in a per-pixel posterior probability of being on the object with Bayes' formula under a uniform prior: $p(obj|s) = p(s|obj)/(p(s|obj)+p(s|bg))$ (Fig. 4d). We compute three different posteriors, by using each of the three background models in turn, resulting in three appearance features for each super-pixel.

Objectness feature. As an additional feature, we incorporate the *objectness* measure [2]. Objectness estimates the probability that an image window contains an object of *any* class. It is designed to distinguish windows containing an object with a well defined boundary and center, such as cows and telephones, from amorphous background windows, such as grass and road. It measures distinctive characteristics of objects, such as appearing different from their surroundings, having a closed boundary, and sometimes being unique within the image.

We sample 1,000 windows using [2] according to their probability of containing an object. Then we convert this measure to per-superpixel probabilities by averaging the objectness value over all windows covering a super-pixel. This objectness feature helps by pushing the segmentation method towards objects and away from backgrounds (Fig. 4e).

Training a superpixel classifier. We train a separate classifier for each object class, as the relation between fixations and objects can be complex and vary between classes. As training samples we use the feature vectors of all superpixels from the $\mathcal{R}_{bb+fix}$ images of a class (Fig. 3). Each superpixel is labeled according to whether it is inside a ground-truth bounding-box or not. After whitening the features, we train a linear SVM with the very efficient implementation of [43] on a random 80% subset of the training data. We set the regularization parameter C by validation on a held-out 10% of the data, and then re-train the SVM on the total 90% of the data. In order to get a smooth, probabilistic

output we apply Platt scaling [35] and fit a sigmoid to the output of the SVM on the remaining 10% of the training data.

Applying the classifier to a new image $I \in \mathcal{R}_{fix}$ yields a soft-segmentation mask M where each pixel value corresponds to the estimated probability of being on the target object. This is the output of the first stage of our method.

4.2 Segmentation refinement

We describe here the second stage of deriving object segmentations from fixations. This refines the soft-segmentation M output by the first stage by taking into account pairwise dependencies between neighboring superpixels and by improving the appearance models.

Let $l_s \in \{0, 1\}$ be the label for superpixel s and L be the labeling of all l_s in the image. We employ a binary pairwise energy function E defined over the superpixels and their labels, analog to [26, 38]

$$E(L) = \sum_s M_s(l_s) + \sum_s A_s(l_s) + \sum_{s,r} V(l_s, l_r) \quad (1)$$

As in [26, 38], the pairwise potential V encourages smoothness by penalizing neighboring pixels taking different labels. The penalty depends on the color contrast between the pixels, being smaller in regions of high contrast (image edges). The summation over (s, r) is defined on an eight-connected pixel grid.

Because of the probabilistic nature of the soft segmentation mask M, we can use $M_s(l_s) = M(s)^{l_s}(1 - M(s))^{1-l_s}$ as a unary potential (with M_s the value of the mask at superpixel s). As M_s estimates the probability that superpixel s is on the object, this potential encourages the final segmentation to be close to M (see [26]). This anchors the segmentation to image regions likely to contain the target object, while letting this second stage refine its exact delineation.

The second unary potential A_s evaluates how likely a superpixel s is to take label l_s, according to object and background appearance models. As in the classic GrabCut [38], an appearance model consists of two GMMs, one for the object (used when $l_s = 1$) and one for the background (used when $l_s = 0$). Each GMM has five components, each a full-covariance Gaussian over the RGB color space.

In traditional work using similar energy models [4, 38, 49], estimating the appearance models requires user interaction to indicate the image region containing the object (typically a manually drawn bounding-box or scribbles). Recently, [26] proposed to automatically estimate the appearance models from a soft segmentation mask produced by transferring segmentations from manually annotated images in a training set. Inspired by that idea, we estimate our appearance models from the mask M obtained from fixations in Sec. 4.1 (so our method does not require training segmentations).

After this initial estimation, we follow [38] 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 [5] as our pairwise potentials are submodular. The second step fits GMMs to labeled superpixels. Defining the energy over superpixels [18, 27, 44] instead

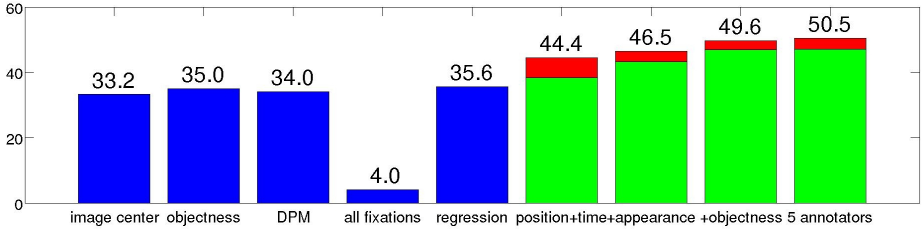


Fig. 6. *CorLoc performance for the baselines (blue bars) and several stripped down versions of our model (green+red) bars. The height of the green bars indicate the performance of the initial segmentation method (i) using only position and timing features, (ii) adding appearance features, (iii) adding the objectness feature, and (iv) using all features and fixations from 5 annotators (the previous three bars use only 2 annotators). The height of the red bars show the improvement brought by the refinement stage.*

of pixels brings great memory savings and reduces the cost to optimize over L. As demonstrated in [18], the accuracy of the superpixel model is essentially identical to the corresponding pixel model.

The final output of our method is a bounding-box enclosing the largest connected component in the segmentation (Fig. 3).

5 Experiments

We carry out experiments on the challenging Pascal VOC 2012 benchmark [13]. We first evaluate the ability of our method to derive object bounding-boxes from fixations on the trainval part of the dataset (Sec. 5.1). Next, we train the object class detector of [15] from these bounding-boxes and compare its performance on the test set to the original model trained from ground-truth bounding-boxes (Sec. 5.2).

5.1 From fixations to bounding-boxes

Data. We consider 10 classes from Pascal VOC 2012, for a total of 6,270 images in the trainval part (see Sec. 3.1 for a list of classes). We split each class into two subsets $\mathcal{R}_{bb+fix}$ and $\mathcal{R}_{fix}$. We use both ground-truth bounding-boxes and human fixations from the set $\mathcal{R}_{bb+fix}$ to train our segmentation model. This subset contains only 7% of all trainval images. This fraction was chosen so that each class has ≥ 20 images. On average, each class has only 44 images. After training, we use our model to predict bounding-boxes for images in the $\mathcal{R}_{fix}$ set, which are annotated only by fixations.

Evaluation measure. We report performance as CorLoc [9, 34, 36, 40], i.e. the percentage of images in $\mathcal{R}_{fix}$ where a method correctly localizes an object of the target class according to the Pascal criterion [14] (intersection-over-union ≥ 0.5).

Baselines. In Fig. 6 we evaluate several informative baselines: **(1) image center:** a window in the image center with area set to the average over the object bounding-boxes in the $\mathcal{R}_{bb+fix}$ set. This baseline provides an indication of the difficulty of the dataset; **(2) all fixations:** a bounding-box around all fixations; **(3) objectness:** the window with the highest objectness probability, out of 1000 sampled using [2]; **(4) DPM:** the highest scored detection returned by a Deformable Parts Model detector [15] trained on the $\mathcal{R}_{bb+fix}$ set; **(5) regression:** linear regression from the mean and variance of the fixation positions to a bounding-box. This models the spatial relation between the cloud of fixations and the object with a single translation and anisotropic scaling transformation, constant across all images of a class.

As Fig. 6 shows, the *image center* baseline achieves 33.2% CorLoc, confirming the dataset contains mostly images with smaller, often off-center objects. Interestingly, *all fixations* fails entirely, demonstrating that the task of deriving bounding-boxes from fixations is far from trivial. *Regression* does much better, finding the object in 35.6% of the images. This highlights the need for learning the relation between fixations and bounding-boxes. Note how *Objectness* also performs quite well (35.0%). It can find some objects even when used on its own, confirming observations made by earlier research [17]. As *regression* and *objectness* incorporate elements of our full method, they set the standards to beat. Finally, the *DPM* baseline reaches only 34.0% CorLoc, showing that the problem cannot be solved simply by training an object detector on the small fully annotated set $\mathcal{R}_{bb+fix}$.

Results. Fig. 6 shows the CorLoc achieved by several stripped down version of our model. We study the impact of using increasingly more features (Sec. 4.1), and the difference between using only the initial segmentation stage (Sec. 4.1) or also the segmentation refinement stage (Sec. 4.2). To quantify the performance of the initial segmentation stage, we threshold the soft-segmentation mask M and fit a bounding-box around the largest segment. The threshold is optimized on the training set $\mathcal{R}_{bb+fix}$.

The results reveal several interesting observations: (1) all feature types we propose contribute to the overall performance of the full model, as adding each type in turn progressively leads to higher CorLoc; (2) with 49.6% CorLoc, our full model significantly outperforms all baselines, including *regression* and *objectness*. This shows that it can learn better, more complex relations between the fixations and the object's spatial extent than the regression baseline. It also shows that, while objectness is a valuable cue to the position of objects, our model goes well beyond it; (3) the segmentation refinement stage always helps, adding between 3% and 6% CorLoc depending on the features used in the initial segmentation stage.

As a reference, the recent weakly supervised method [39], which produces bounding-boxes on a set of images labelled only as containing a class, achieves 32% in a similar setting on Pascal VOC 2007 (which is a dataset of similar difficulty). This provides a ballpark for what can be expected when using such alternative methods.

Our results discussed above used fixations from just *two* annotators. This is an especially economical annotation scenario, as it takes only a total of about *two seconds* to annotate an image, substantially less than the 26 seconds it takes to draw a bounding-box [20]. However, as we collected fixations from five annotators per image (Sec. 3), for

Fig. 7. *Qualitative results of our method for localizing objects given fixations. The fixations are indicated with yellow circles while the predicted bounding-boxes are in green. Successful examples are shown in the first four rows for various classes, while some failure cases are shown in the last row. Note how our method nicely outputs a bounding-box covering the whole object even when the fixations are concentrated on only part of it, often the center, or the head.*

completeness we report also the performance of our model when using all annotators (rightmost bar in Fig. 6). As performance improves only marginally, we conclude that our model efficiently exploits the information provided by two annotators, and keep this as the default setting in the next experiment.

5.2 Training object class detectors from fixations

Settings. In the previous subsection we automatically derived bounding-boxes from fixations for the $\mathcal{R}_{fix}$ part of the Pascal VOC 2012 trainval set (i.e. 93% of the total).

Here we use these predicted bounding-boxes, along with the 7% images in $\mathcal{R}_{bb+fix}$ with ground-truth bounding-boxes, to train a DPM detector per class [15]. Following the standard Pascal VOC protocol, the negative training set for a class contains all trainval images not containing that class. After training, the detectors are applied to the Pascal VOC 2012 test set (10,991 images). Performance is quantified by the mean average precision (mAP) over all 10 classes, as returned by the Pascal VOC evaluation server. We compare performance to DPM detectors trained from exactly the same images, but *all* annotated with ground-truth bounding-boxes.

Results. The mAP of detectors trained from bounding-boxes derived by our method is 12.5%, compared to 25.5% for the detectors trained from ground-truth bounding-boxes. We consider this an encouraging result, given that our scenario enables to train these detectors in 6.8× less total annotation time compared to drawing bounding-boxes on all images. This estimate takes into account all relevant factors, i.e. two annotators per image at one second per image, the time to set up and calibrate the eye tracker, breaks between blocks of images, and the time to draw bounding-boxes on the 7% images in $\mathcal{R}_{bb+fix}$. Interestingly, training DPMs from ground-truth bounding-boxes for a 6.8× smaller training set leads to comparable performance as our method (13.7%).

6 Conclusions

We have presented a novel approach to train object detectors. Instead of the traditional, time consuming manual bounding-box annotation protocol, we proposed to learn the detector from eye movement data recorded while annotators simply look for the object in the training images. We proposed a technique to successfully derive object bounding-boxes from such eye movement data and demonstrated that they can be used to train a modern object detector [15].

In its current form, when given equal total annotation time, our scheme leads to detectors that are about as good as those trained from manual bounding-box annotations. In future work we plan to further improve the quality of our segmentation model, so as to localize more objects and more accurately, which should lead to training better detectors. Moreover, the bounding-boxes predicted by our method could be sent to Amazon Mechanical Turk for verification, leading to cleaner training sets at a small additional cost. As another direction, we plan to extend the work to connect it to weakly supervised models that look for repeated appearance patterns in multiple training images [10, 40]. Finally, we would like to extend the work to video, ideally moving towards the dream paradigm of 'learning object detectors while watching TV'.

Acknowledgements F. Keller was supported by the ERC Starting Grant "Synchronous Linguistic and Visual Processing". V. Ferrari was supported by the ERC Starting Grant "Visual Culture for Image Understanding". We are grateful to Mukta Prasad for suggesting this research direction.

References

1. Imagenet large scale visual recognition challenge (ILSVRC). http://www.image-net.org/challenges/LSVRC/2011/index (2011)
2. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: CVPR (2010)
3. Berg, T., Berg, A., Edwards, J., Mair, M., White, R., Teh, Y., Learned-Miller, E., Forsyth, D.: Names and Faces in the News. In: CVPR (2004)
4. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive GMMRF model. In: ECCV (2004)
5. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. on PAMI 26(9), 1124–1137 (2004)
6. Brainard, D.H.: The Psychophysics Toolbox. Spatial Vision 10, 433–436 (1997)
7. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: CVPR (2007)
8. Dalal, N., Triggs, B.: Histogram of Oriented Gradients for human detection. In: CVPR (2005)
9. Deselaers, T., Alexe, B., Ferrari, V.: Weakly supervised localization and learning with generic knowledge. IJCV (2012)
10. Deselaers, T., Ferrari, V.: Global and efficient self-similarity for object classification and detection. In: CVPR (2010)
11. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: ECCV (2002)
12. Einhäuser, W., Spain, M., Perona, P.: Objects predict fixations better than early saliency. Journal of Vision 8, 1–26 (2008)
13. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
14. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
15. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. on PAMI 32(9) (2010)
16. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)
17. Guillaumin, M., Ferrari, V.: Large-scale knowledge transfer for object localization in imagenet. In: CVPR (2012)
18. Guillaumin, M., Küttel, D., Ferrari, V.: ImageNet auto-annotation with segmentation propagation. IJCV (2014)
19. Gupta, A., Davis, L.: Beyond nouns: Exploiting prepositions and comparators for learning visual classifiers. In: ECCV (2008)
20. Hao, S., Deng, J., Fei-Fei, L.: Crowdsourcing annotations for visual object detection. In: AAAI (2012)
21. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: NIPS (2007)
22. Henderson, J.: Human gaze control in real-world scene perception. Trends in Cognitive Sciences 7, 498–504 (2003)
23. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. on PAMI 20(11), 1254–1259 (1998)
24. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: IEEE International Conference on Computer Vision (ICCV) (2009)
25. Karthikeyan, S., Jagadeesh, V., Shenoy, R., Eckstein, M., Manjunath, B.: From where and how to whatwe see. In: ICCV (2013)
26. Kuettel, D., Ferrari, V.: Figure-ground segmentation by transferring window masks. In: CVPR (2012)

27. Ladicky, L., Russell, C., Kohli, P.: Associative hierarchical crfs for object class image segmentation. In: ICCV (2009)
28. Leistner, C., Godec, M., Schulter, S., Saffari, A., Bischof, H.: Improving classifiers with weakly-related videos. In: CVPR (2011)
29. Levinshtein, A., Stere, A., Kutulakos, K., Fleed, D., Dickinson, S.: Turbopixels: Fast superpixels using geometric flows. In: IEEE Trans. on PAMI (2009)
30. Luo, J., Caputo, B., Ferrari, V.: Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In: NIPS (2009)
31. Mathe, S., Sminchisescu, C.: Dynamic eye movement datasets and learnt saliency models for visual action recognition. In: ECCV (2012)
32. Mishra, A., Aloimonos, Y., Fah, C.L.: Active segmentation with fixation. In: ICCV (2009)
33. Nuthmann, A., Henderson, J.M.: Object-based attentional selection in scene viewing. Journal of Vision 10(8), 1–19 (2010)
34. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: ICCV (2011)
35. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers (1999)
36. Prest, A., Leistner, C., Civera, J., Schmid, C., Ferrari, V.: Learning object class detectors from weakly annotated video. In: CVPR (2012)
37. Ramanathan, S., Katti, H., Sebe, N., Kankanhalli, M., Chua, T.S.: An eye fixation database for saliency detection in images. In: ECCV (2010)
38. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. SIGGRAPH (2004)
39. Siva, P., Russell, C., Xiang, T., Agapito, L.: Looking beyond the image: Unsupervised learning for object saliency and detection. In: CVPR (2013)
40. Siva, P., Xiang, T.: Weakly supervised object detector learning with model drift detection. In: ICCV (2011)
41. Siva, P., Xiang, T., Russell, C.: In defence of negative mining for annotating weakly labeled data. In: ECCV (2012)
42. Tang, K., Sukthankar, R., Yagnik, J., Fei-Fei, L.: Discriminative segment annotation in weakly labeled video. In: CVPR (2013)
43. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008)
44. Veksler, O., Boykov, Y., Mehrani, P.: Superpixels and supervoxels in an energy optimization framework. In: ECCV. pp. 211–224 (2010)
45. Vicente, S., Rother, C., Kolmogorov, V.: Object cosegmentation. In: CVPR. pp. 2217–2224 (2011)
46. Vig, E., Dorr, M., Cox, D.: Saliency-based space-variant descriptor sampling for action recognition. In: ECCV (2012)
47. Viola, P.A., Platt, J., Zhang, C.: Multiple instance boosting for object detection. In: NIPS (2005)
48. Walber, T., Scherp, A., Staab, S.: Can you see it? two novel eye-tracking-based measures for assigning tags to image regions. In: MMM (2013)
49. Wang, J., Cohen, M.: An iterative optimization approach for unified image segmentation and matting. In: ICCV (2005)
50. Wang, X., Yang, M., Zhu, S., Lin, Y.: Regionlets for generic object detection. In: ICCV. pp. 17–24. IEEE (2013)
51. Wolfe, J., Horowitz, T.S.: Visual search. Scholarpedia 3(7), 3325 (2008)
52. Yun, K., Peng, Y., Samaras, D., Zelinsky, G.J., Berg, T.L.: Studying relationships between human gaze, description, and computer vision. In: CVPR (2013)