



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Localizing Objects While Learning Their Appearance

Citation for published version:

Deselaers, T, Alexe, B & Ferrari, V 2010, Localizing Objects While Learning Their Appearance. in K Daniilidis, P Maragos & N Paragios (eds), Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 6314, Springer Berlin Heidelberg, pp. 452-466. DOI: 10.1007/978-3-642-15561-1_33

Digital Object Identifier (DOI):

[10.1007/978-3-642-15561-1_33](https://doi.org/10.1007/978-3-642-15561-1_33)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Early version, also known as pre-print

Published In:

Computer Vision – ECCV 2010

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Localizing Objects While Learning Their Appearance

Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari

Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland
{deselaers,bogdan,ferrari}@vision.ee.ethz.ch

Abstract. Learning a new object class from cluttered training images is very challenging when the location of object instances is unknown. Previous works generally require objects covering a large portion of the images. We present a novel approach that can cope with extensive clutter as well as large scale and appearance variations between object instances. To make this possible we propose a conditional random field that starts from generic knowledge and then progressively adapts to the new class. Our approach simultaneously localizes object instances while learning an appearance model specific for the class. We demonstrate this on the challenging PASCAL VOC 2007 dataset. Furthermore, our method enables to train any state-of-the-art object detector in a weakly supervised fashion, although it would normally require object location annotations.

1 Introduction

In weakly supervised learning (WSL) we are given a set of images, each containing one or more instances of an unknown object class. In contrast to the fully supervised scenario, the location of objects is *not* given. The task is to learn a model for this object class, which can then be used to determine whether a test image contains the class and possibly even to localize it (typically up to a bounding-box). In this case, the learned model is asked to do more than what the training examples teach.

WSL has become a major topic in recent years [1–7] to reduce the manual labeling effort to learn object classes. In the traditional paradigm, each new class is learned from scratch without any knowledge other than what was engineered into the system. In this paper, we explore a scenario where generic knowledge about object classes is first learned during a *meta-training stage* when images of many different classes are provided along with the location of objects. This generic knowledge is then used to support the learning of a new class *without* location annotation (fig. 1). Generic knowledge makes WSL easier as it rests on a stronger basis.

We propose a conditional random field (CRF) to simultaneously localize object instances and learn an appearance model for the new class. The CRF aims at selecting one window per image containing an instance of the new object class. We alternate between localizing the objects in the training images and learning class-specific models that are then incorporated into the next iteration. Initially

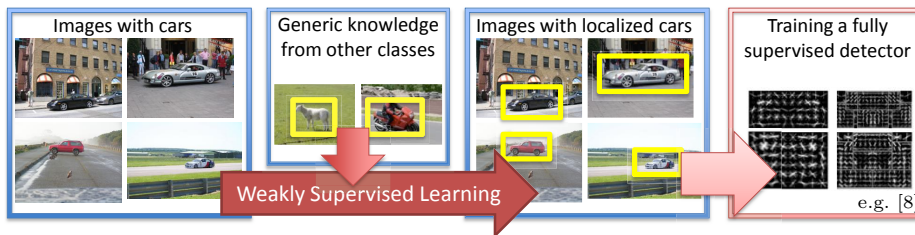


Fig. 1. Learning scenario. Starting from weakly supervised images we localize the object instances of a new class while learning an appearance model. Generic knowledge is used to start WSL on a stronger basis. Our method can be used as a pre-processing step to any fully supervised object detector.

the CRF employs generic knowledge to guide the selection process as it *reduces the location ambiguity*. Over the iterations the CRF progressively adapts to the new class, learning more and more about its appearance and shape. This strategy enables our method to learn from very cluttered images containing objects with large variations in appearance and scale, such as the PASCAL VOC 2007 [9] (fig. 4,5). To the best of our knowledge, no earlier method has been demonstrated capable of learning from PASCAL07 in a WSL scenario (but on easier datasets such as Caltech4 [3] or Weizmann horses [10]).

The main contribution of this paper is a novel method to jointly localize and learn a new class from WS data. Therefore, in sec. 7 we directly evaluate performance as the percentage of instances of the new class which it localizes in their WS training images, and compare to two existing methods [11,12] and various baselines. Moreover, we also demonstrate an application of our method: we train the fully supervised model of Felzenszwalb et al. [8] from objects localized by our method, evaluate it on the PASCAL07 test set, and compare its performance to the original model trained from ground-truth bounding-boxes.

Related Work. We focus here on WSL methods to learn object classes (i.e. requiring no object locations). Many approaches are based on a *bag-of-words* for the entire image [13,14]. Although they have demonstrated impressive classification performance [9], they are usually unable to localize objects.

There are several WSL methods that achieve localization, such as part-based [2,3], segmentation-based [1,4–6,11,15], and others [7,12,16]. However, most methods have been demonstrated on datasets such as Caltech4 [1–4,6,7,16] and Weizmann horses [6,10,15], where objects are rather centered and occupy a large portion of the image, there is little scale/viewpoint variation, and limited background clutter. This is due to the difficulty of spotting the recurring object pattern in challenging imaging conditions.

There are a few exceptions [11,12,17]. [11] attempts to segment out regions similar across many images from the difficult LabelMe dataset [18], but reports that it is very hard to find small objects such as cars in it. [12] is related to our approach as it also finds one window per image. It iteratively refines windows initialized from the most discriminative local features. This fails when the objects

occupy a modest portion of the images and for classes such as horses, for which local texture features have little discriminative power. [17] clusters windows of similar appearance using link analysis techniques. Both [12] and [17] experiment on (part of) the PASCAL VOC 06 dataset. We quantitatively compare to [11, 12] in sec. 7.

Our use of generic knowledge is related to *transfer learning* [19, 20], where learning a new class is helped by labeled examples of other classes. There are relatively few works on transfer learning for visual recognition. Lando and Edelman [21] learn a new face from just one view, supported by images of other faces. Fei-Fei [22] sequentially updates a part-based classifier trained on previous object classes to fit a new class from very few examples. Stark et al. [23] transfer shape knowledge between related classes in a manually controlled manner. Tommasi et al. [24] use the parameters of the SVM for a known class as a prior for a new, related class. These works reduce the number of images necessary to learn a new class, improving generalization from only a few examples [20]. In this paper instead, we reduce the *degree of supervision* (i.e. no object locations). As another difference, the works above transfer knowledge from one class to another, whereas our generic knowledge provides a background against which it is easier to learn *any* new class. Our generic knowledge conveys how to localize new classes. Automatically *localizing* instances of the new class in their training images is a central objective of our work.

Plan of the Paper. Our new CRF model is described in sec. 2. In sec. 3 and 4 we explain how it is used to localize instances of a new object class in WS training images while learning a model of the new class. Sec. 5 details the generic knowledge that is incorporated into the process and how it is obtained. Sec. 6 describes the image cues we use and in sec. 7 we experimentally evaluate the method.

2 The CRF Model to Localize a New Class

The goal of this paper is to simultaneously localize objects of a new target class in a set of training images and learn an appearance model of the class. As we make no assumption about object locations, scales, or overall shape (aspect-ratio), any image window can potentially contain an object of the target class. We select one window per image by optimizing an energy function defined globally over all training images. Ideally the energy is minimal when all selected windows contain an object of the same class.

Configuration of Windows L . The set of training images $\mathcal{I} = (I_1, \dots, I_N)$ is represented as a fully connected CRF. Each image I_n is a node which can take on a state from a discrete set corresponding to all image windows. The posterior probability for a configuration of windows $L = (l_1, \dots, l_N)$ can be written as

$$p(L|\mathcal{I}, \theta) \propto \exp \left(\sum_n \rho_n \Phi(l_n|I_n, \theta) + \sum_{n,m} \rho_n \rho_m \Psi(l_n, l_m|I_n, I_m, \theta) \right) \quad (1)$$

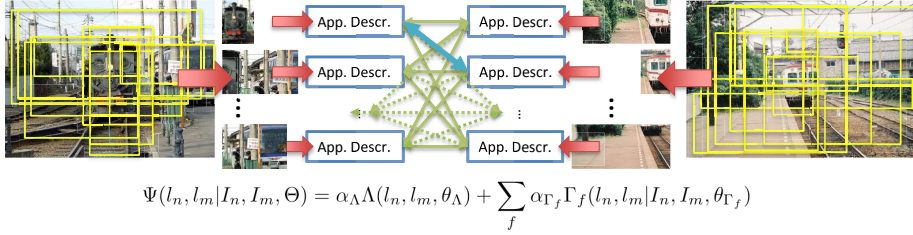


Fig. 2. The pairwise potential. Two images with candidate windows (yellow). Appearance descriptors are extracted for each window (arrows). The pairwise potential Ψ is computed for every pair of windows between the two images, as a linear combination of appearance similarity cues Γ_f and the aspect-ratio similarity Λ .

where each l_n is a window in image I_n ; Θ are the parameters of the CRF; ρ_n is the responsibility of image I_n , weighting its impact on the overall energy (sec. 4.3).

The Unary Potential Φ measures how likely an image window l_n is to contain an object of the target class

$$\Phi(l_n; I_n) = \alpha_\Omega \Omega(l_n | I_n, \theta_\Omega) + \alpha_\Pi \Pi(l_n | I_n, \theta_\Pi) + \sum_f \alpha_{\Upsilon_f} \Upsilon_f(l_n | I_n, \theta_{\Upsilon_f}) \quad (2)$$

It is a linear combination of: (a) Ω , the likelihood [25] that l_n contains an object of *any* class, rather than background (sec. 5.1); (b) Π , a model of the overall shape of the windows, specific to the target class (sec. 4.2); (c) Υ_f , appearance models, one for each image cue f , specific to the target class (sec. 4.1). The scalars α weight the terms.

The Pairwise Potential Ψ measures the similarity between two windows, assessing how likely they are to contain objects of the same class (fig. 2).

$$\Psi(l_n, l_m | I_n, I_m, \Theta) = \alpha_\Lambda \Lambda(l_n, l_m, \theta_\Lambda) + \sum_f \alpha_{\Gamma_f} \Gamma_f(l_n, l_m | I_n, I_m) \quad (3)$$

It is a linear combination of: (a) Λ , a prior on the shape similarity between two windows l_n, l_m , depending only on states l_n, l_m (sec. 5.2); (b) a term Γ_f measuring the appearance similarity between l_n and l_m according to multiple cues f that depends on the image content (sec. 5.3). The scalars α weight the terms. Fig. 2 illustrates the computation of the pairwise potential for every pair of windows between two images.

The Parameters $\theta_\Omega, \theta_\Lambda, \theta_{\Gamma_f}$ and the weights α are learned from meta-training data (sec. 5). The class-specific models Π and Υ and the image responsibilities ρ_n are initially unknown and set to uniform. Over the learning iterations they are progressively adapted to the target class (sec. 4).

Note that our model connects nodes (windows) *between* images, rather than elements *within* an image as typically done for CRFs in other domains (e.g. pixels in segmentation [26], body parts in human pose estimation [27]).

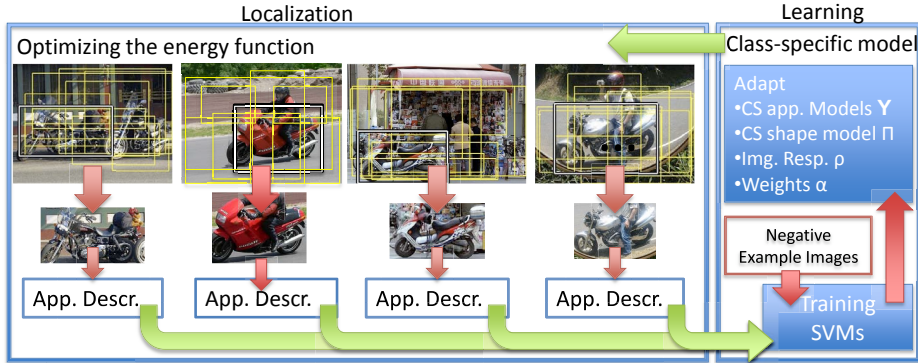


Fig. 3. Localization and learning. The localization and learning stages are alternated. *Localization*: one window (black/white) is selected among the candidate windows (yellow) for each image. *Learning*: a model \mathcal{Y} specific to the target class is (re)-trained from the appearance descriptors of the selected windows and a set of negative training windows. Other CRF components are adapted to the class and the CRF is updated.

3 Localization and Learning

When given a set of images \mathcal{I} of a target class the goal is to localize its object instances. The *localization* and *learning* stages are alternated, optimizing one while keeping the other fixed (fig. 3).

3.1 Localization. Localizing objects corresponds to finding the configuration L^* that maximizes eq. (1):

$$L^* = \arg \max_L \{p(L|\mathcal{I}, \Theta)\} \quad (4)$$

The selected windows L^* are the most likely to contain instances of the same object class (according to our model).

Optimizing our fully connected model is NP-hard. We approximate the global optimum using the tree-reweighted message passing algorithm TRW-S [28]. TRW-S also returns a lower bound on the energy. When this coincides with the returned solution, we know it found the global optimum. In our experiments, TRW-S finds it in 93% of the cases, and in the others the lower bound is only 0.06% smaller on average than the returned energy. Thus we know that the obtained configurations L^* are very close to the global optimum.

3.2 Learning. Based on the selected windows L^* , we adapt several characteristics of the CRF to the target class: (a) the class-specific appearance models \mathcal{X}_f , (b) the class-specific shape model Π , (c) the image responsibilities ρ_n , and (d) the weights α of the cues (details in sec. 4).

The localization and learning stages *help each other*, as better localizations lead to better class-specific models, which in turn sharpen localization. Similar EM-like optimization schemes [8] are commonly used to learn in the presence of latent variables (in our case L^*).

4 Adaptation

During the learning stage (sec. 3.2), the CRF is progressively adapted from generic to class-specific. For this adaptation, an additional negative image set \mathcal{N} is used, which does not contain any object of the target class.

4.1 Class-specific Appearance Models Υ_f . Any model trainable from annotated object windows could be used here (e.g. [8, 14, 29, 30]). We train a separate SVM θ_{Υ_f} for each appearance cue f .

Since usually not all selected windows L^* contain an object of the target class, these SVMs are iteratively trained. First, the SVM θ_f is trained to separate all windows L^* from windows randomly sampled from \mathcal{N} . Then, the SVM θ_f is used to score each training window $l_n^* \in L^*$. The top scored $\kappa\%$ windows are then used to retrain θ_f . This is repeated ten times.

4.2 Class-specific Shape Model II. The parameters θ_{II} are learned as the distribution of the aspect-ratio of the selected windows L^* .

4.3 Image Responsibilities ρ_n emphasize images where the model is confident of having localized an object of the target class. We set ρ_n proportional to the score of the class-specific appearance model: $\rho_n \propto \sum_f \alpha_{\Upsilon_f} \Upsilon_f(l_n | I_n, \theta_{\Upsilon_f})$. This reduces the impact of particularly difficult images and makes the model more robust to outliers.

4.4 Unary Appearance Cue Weights α_{Υ_f} . Not all classes can be discriminated equally well using the same cues (e.g. motorbikes can be recognized well using texture patches, mugs using shape/gradient features, and sheep using color). We adapt the weights α_{Υ_f} of the class-specific appearance models Υ_f for the cues f . We use the top-scored $\kappa\%$ selected windows to train a linear SVM w to combine their appearance scores $\Upsilon_f(l_n | I_n, \theta_{\Upsilon_f})$. Then, we update $\alpha_{\Upsilon_f} \leftarrow \alpha_{\Upsilon_f} + \lambda w_f$. The scalar λ controls the adaptation rate.

4.5 Pairwise Appearance Cue Weights α_{Γ_f} . We proceed analogously to the previous paragraph. The SVM w is trained to combine the scores $\Gamma_f(l_n, l_m | I_n, I_m)$ between all pairs of the top $\kappa\%$ selected windows.

The objectness Ω , the shape similarity Λ , and the appearance similarity Γ_f are not explicitly adapted to the target class but only implicitly through weights $\alpha_{\Upsilon_f}, \alpha_{\Gamma_f}$ and image responsibilities ρ_n .

5 Generic Knowledge: Initializing Θ

Initially the model parameters Θ carry only generic knowledge. They are learned in a meta-training stage to maximize the localization performance on a set of meta-training images \mathcal{M} containing objects of known classes annotated with bounding-boxes.

5.1 Objectness Ω . We use the objectness measure $\Omega(l|I, \theta_\Omega)$ of [25], which quantifies how likely it is for a window l to contain an object of *any* class. Objectness is trained 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. Objectness combines several image cues measuring distinctive characteristics of objects, such as appearing different from their surroundings, having a closed boundary, and sometimes being unique within the image.

We use objectness as a location prior in our CRF, by evaluating it for all windows in an image I and then sampling 100 windows according to their scores $\Omega(l|I, \theta_\Omega)$. These form the set of states for node I (i.e. the candidate windows the CRF can choose from).

This procedure brings two advantages. First, it greatly reduces the computational complexity of CRF inference, which grows with the square of the number of states (there are $\simeq 10^8$ windows in an image [30]). Second, the sampled windows and their scores Ω attract the CRF toward selecting objects rather than background windows. In a WSL setup this avoids trivial solutions, e.g. where all selected windows cover a chunk of sky in airplane training images [7]. In sec. 7 we evaluate objectness quantitatively. For more details about objectness see [25].

5.2 Pairwise Shape Similarity Λ . θ_Λ is learned as the Bayesian posterior $\Lambda(l_n, l_m, \theta_\Lambda) = p(l_n \stackrel{c}{=} l_m | \text{SS}(l_n, l_m))$ from many window pairs containing the same ($l_n \stackrel{c}{=} l_m$) and different classes. $\text{SS}(l_n, l_m)$ measures the aspect ratio similarity of the windows l_n and l_m . In practice this learns that instances of the same class have similar aspect-ratios.

5.3 Pairwise Appearance Similarity Γ_f . We compute the similarity between two windows l_n, l_m in images I_n, I_m as the SSD $\|l_n^f(I_n) - l_m^f(I_m)\|^2$ between their appearance descriptors $l_n^f(I_n)$ and $l_m^f(I_m)$. This measures how likely they are to contain instances of the same class, according to cue f . The pairwise potentials Γ_f are directly defined as $\Gamma_f(l_n, l_m | I_n, I_m) = \|l_n^f(I_n) - l_m^f(I_m)\|^2$.

5.4 Weights α . To learn the weights α between the various terms of our model, we perform a multi-stage grid search.

First, we learn the weights α_Ω , α_Λ , and α_{Γ_f} for objectness Ω , shape similarity Λ , and appearance similarity Γ_f so that the windows L^* returned by the localization stage (sec. 3.1) best cover the meta-training bounding-boxes \mathcal{M} (according to the criterion in sec. 7.1). These weights are determined using only the localization stage, not the adaptation stage, as they contain no class-specific knowledge.

With these weights fixed, we proceed to determine the remaining weights α_Π and α_{Υ_f} for the class-specific shape model Π and the class-specific appearance models Υ_f . These are learned using the full method (sec. 3.1 and 3.2).

5.5 Kernel of the SVMs Υ_f . We evaluated linear and intersection kernels for the class-specific appearance models Υ_f and found the latter to perform slightly better.

5.6 Percentage κ of images. With weights α and the SVM kernels fixed, we determine the optimal percentage κ of selected windows to use for the iterative training in sec. 4.1.

The remaining parameters, the class-specific appearance models \mathcal{Y}_f , the class-specific shape model Π , and the image responsibilities ρ_n are not learned from meta-training data. They are initially unknown and set uniformly.

6 Appearance cues

We extract 4 appearance descriptors f from each candidate window and use them to calculate the appearance similarity I_f and the class-specific appearance score \mathcal{Y}_f .

GIST [31] is based on localized histograms of gradient orientations. It captures the rough spatial arrangement of image structures, and has been shown to work well for describing the overall appearance of a scene. Here instead, we extract GIST from each candidate *window*.

Color Histograms (CH) provide complementary information to gradients. We describe a window with a single histogram in the LAB color space.

Bag of Visual Words (BOW) are de-facto standard for many object recognition tasks [12–14, 30]. We use the SURF descriptors [30, 32] and quantize them into 2000 words using k -means. A window is described by a BOW of SURF.

Histograms of Oriented Gradients (HOG) also are an established descriptor for object class recognition [8, 29].

7 Experiments: WS localization and learning

We evaluate the central ability of our method: localizing objects in weakly supervised training images. We experiment on datasets of varying difficulty.

Caltech4 [3]. We use 100 random images for each of the four classes in this popular dataset (airplanes, cars, faces, motorbikes). The images contain large, centered objects, and there is limited scale variation and background clutter.

As meta-training data \mathcal{M} we use 444 train+val images from 6 PASCAL07 classes (bicycle, bird, boat, bus, dog, sheep) with bounding-box annotations. \mathcal{M} is used to learn the parameters for initializing our CRF (sec. 5). This is done only once. The same parameters are then reused in all experiments.

Pascal06 [12, 33]. For comparison, we run our method on the training subsets used by [12]¹. These include images for each aspect of 6 classes: car, bicycle, bus, motorbike, cow, and sheep. Up to four aspects are considered per class, totaling 14 training sets (see [12] for details). Although PASCAL VOC06 images are challenging in general, these subsets are easier and contain many large objects. As meta-training data \mathcal{M} we use 471 train+val images from 6 PASCAL07 classes (aeroplane, bird, boat, cat, dog, horse).

¹ Provided to us by the authors of [12]

Pascal07-6x2 [9]. For the detailed evaluation of the components of our method below, we use all images from 6 classes (aeroplane, bicycle, boat, bus, horse, and motorbike) of the PASCAL VOC 2007 train+val dataset from the left and right aspect each. Each of the 12 class/aspect combination contains between 28 and 67 images for a total of 538 images. As negative set \mathcal{N} we use 2000 random images taken from train+val not containing any instance of the target class. This dataset is very challenging, as objects vary greatly in location, scale, and appearance. Moreover, there is significant viewpoint variation within an aspect (fig. 4, 5). We report in detail on these classes because they represent compact objects on which fully supervised methods perform reasonably well [9] (as opposed to classes such as ‘potted plant’ where even fully supervised methods fail). As meta-training data \mathcal{M} we use 799 train+val images from 6 other PASCAL07 classes (bird, car, cat, cow, dog, sheep).

Pascal07-all [9]. For completeness, we also report results for *all* class/aspect combinations in PASCAL07 with more than 25 images (our method, as well as the competitors and baselines to which we compare, fails when given fewer images). We use the same meta-training data as for PASCAL07-6x2. In total, the PASCAL07-all set contains 42 class/aspect combinations, covering all 14 classes not used for meta-training.

7.1 Localizing Objects in their Weakly Supervised Training Images

We *directly* evaluate the ability of our method to localize objects in a set of training images \mathcal{I} only known to contain a target class (sec. 7). Tab. 1 shows results for two baselines, two competing methods [11, 12] and for several variants of our method. We report as CorLoc the percentage of images in which a method correctly localizes an object of the target class according to the PASCAL-criterion (window intersection-over-union > 0.5). No location of any object in \mathcal{I} is given to any method beforehand. The detailed analysis in the following paragraphs focuses on the Caltech4, PASCAL06, and PASCAL07-6x2 datasets. The last paragraph discusses results on the PASCAL07-all dataset.

Baselines. The ‘image center’ baseline simply picks a window in the image center by chopping 10% off the width/height from the image borders. This is useful to assess the difficulty of a dataset. The ‘ESS’ baseline is based on bag-of-visual-words. We extract SURF features [32] from all images of a dataset, cluster them into 2000 words, and weight each word by the log of the relative frequency of occurrence in positive vs negative images of a class (as done by [12, 13, 30]). Hence, these feature weights are class-specific. For localization, we use Efficient Subwindow Search (ESS) [30] to find the window with the highest sum of weights in an image².

The image center baseline confirms our impressions about the difficulty of the datasets. It reaches about 70% CorLoc on Caltech4 and PASCAL06-[12], but fails on PASCAL07. The trend is confirmed by ESS.

² Baseline suggested by C. Lampert in personal communications.

Table 1. Results. The first block reports results for the baselines and the second for the competitors [11,12]. Rows (a)-(c): results for our method using only the localization stage. Rows (d)-(e): results for the full method using the localization and learning stages. All results until row (e) are given in CorLoc. Rows (f)-(g) report the performance of the objectness measure Ω (see main text). Column (Color) shows the colors used for visualization in figs. 4, 5.

Method	Caltech4	PASCAL07			Color
		PASCAL06-[12]	6×2	all	
image center	66	76	23	14	
ESS	43	33	23	10	
Russel et al. [11]	40	58	20	13	■
Chum et al. [12]	57	67	29	15	■
this paper – localization only					
(a) random windows	0	0	0	0	
(b) single cue (GIST)	72	64	35	21	
(c) all cues	70	77	39	22	■
this paper – localization and learning					
(d) learning \mathcal{Y}_f	85	84	48	22	
(e) full adaptation	87	82	50	26	■
objectness measure Ω					
(f) hit-rate	100	99	89	85	
(g) signal-to-noise	29	31	16	14	

Competitors. We compare to the method from [11] using their implementation³. This method does not directly return one window per image. It determines 30 topics roughly corresponding to object classes. A topic consists of a group of superpixels in each training image. For each topic, we put a bounding-box around its superpixels in every image, and then evaluate its CorLoc performance. We report the performance of the topic with the highest CorLoc. This method achieves a modest CorLoc on the challenging PASCAL07-6x2, but found the object in about half the images of the easier Caltech4 and PASCAL06-[12].

As a second competitor we reimplemented the method from [12], which directly returns one window per image. It works quite well on Caltech4 and on their PASCAL06-[12] subset, where the objects occupy a large portion of the images. On the much harder PASCAL07-6x2 it performs considerably worse since its initialization stage does not lock onto objects⁴. Overall, this method performed better than [11] on all three datasets.

Localization Only (a)-(c). Here we stop our method after the localization stage (sec. 3.1), without running the learning stage (sec. 3.2). In order to in-

³ http://www.di.ens.fr/~russell/projects/mult_seg_discovery/index.html

⁴ Unfortunately, we could not obtain the source code from the authors of [12]. We asked them to process our PASCAL07-6x2 training sets and they confirmed that their method performs poorly on them.

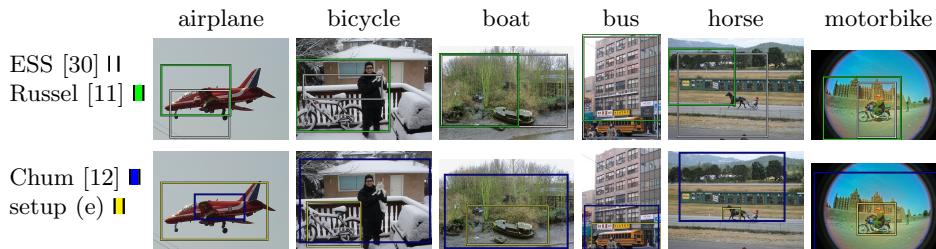


Fig. 4. Comparison to baselines and competitors. Example objects localized by different methods in their weakly supervised training images (i.e. only object presence is given for training, no object locations). Top row: the ESS baseline [30] \blacksquare and the method from [11] \blacksquare . Bottom row: the method from [12] \blacksquare and our method in setup (e) \blacksquare . Our method localizes object visibly better than both baselines and competitors, especially in difficult images.

investigate the impact of generic knowledge, we perform experiments with several stripped-down versions of our CRF model. Models (a) and (b) use only GIST descriptors in the pairwise similarity score Γ_f . (a) uses 100 random candidate windows with uniform scores in Ω ; (b) uses 100 candidate windows sampled from the objectness measure Ω (sec. 5.1). While method (a) is not able to localize any object, (b) already performs quite well.

By adding the remaining appearance cues Γ_f in setup (c), results improve further and all baselines and competitors are outperformed. Using only the localization stage, our method already localizes more than 70% of the objects in Caltech4 and PASCAL06-[12], and 39% of the objects in PASCAL07-6x2.

Localization and Learning (d)-(e). Here we run our full method, iteratively alternating localization and learning. In setup (d), we learn only appearance models Υ_f specific to the target class. In setup (e), all parameters of the CRF are adapted to the target class. The considerable increase in CorLoc shows that the learning stage helps localization. The full method (e) substantially outperforms all competitors/baselines on all datasets, and in particular reaches about twice their CorLoc on PASCAL07-6x2. Overall, it finds most objects in Caltech4 and PASCAL06-[12], and half of those in PASCAL07-6x2 (fig. 4, 5).

As tab. 1 shows, each variant improves over the previous one. Showing that (i) the generic knowledge elements we incorporate are important for a successful initial localization (setups (a)-(c)) and (ii) the learning stage successfully adapts the model to the target class (setups (d)-(e)).

Pascal07-all. For completeness, we report in tab. 1 also results over the PASCAL07-all set, which contains 42 class/aspect combinations, including many for which even fully supervised methods fail (e.g. ‘potted plant’). Compared to PASCAL07-6x2, CorLoc drops by about half for all methods, suggesting that WS learning on *all* PASCAL07 classes is beyond what currently possible. However, it is interesting to notice how the relative performance of our method (setup (e)) compared to the competitors [11, 12] is close to what observed in PASCAL07-6x2: our method performs about twice as well as them.



Fig. 5. Example results comparing our method in setup (c) ■ to setup (e) ■. If only ■ is visible, both setups chose the same window. The learning stage in setup (e) leads to more correctly localized objects.

Objectness Measure (f)-(g). We also evaluate the 100 windows sampled from Ω . The percentage (f) of objects of the target class covered by a sampled window gives an upper-bound on the CorLoc that can be achieved by our method. As the table shows, most target objects are covered. The percentage (g) of sampled windows covering an object of the target class gives the signal-to-noise ratio that enters the CRF model. This ratio is much higher than when considering all image windows.

8 Experiments: Localizing Objects in New Test Images

Our method enables training a fully-supervised object detector from weakly supervised data, although it would normally require object location annotations. To demonstrate this point, we train the fully supervised object detector of [8]⁵ from objects localized using setup (e) and compare its performance to the original model trained from ground-truth bounding-boxes.

We perform this experiment for all 12 class/aspect combinations in PASCAL07-6x2. The detection performance for each class/aspect is measured by the average precision (AP) on the entire PASCAL 2007 test set (4952 images). We report below the mean AP over the 12 class/aspect combinations (mAP). As usual in a test stage, no information is given about the test images, not even

⁵ The source code is available at <http://people.cs.uchicago.edu/~pff/latent/>

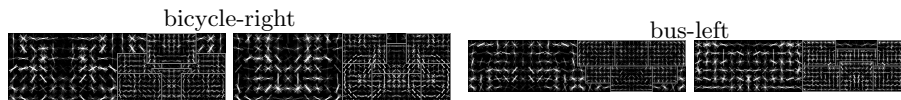


Fig. 6. Models [8] trained from the output of our method (left) and from ground-truth bounding-boxes (right).

whether they contain the object. Notice how this test set is entirely disjoint from the train+val set used for training and meta-training.

The mAP resulting from models trained in a weakly supervised setting from the output of our method is 0.16, compared to 0.33 of the original fully supervised models. Therefore, our method enables to train [8] *without* ground-truth bounding-box annotations, while keeping the detection performance on the test set at about 48% of the model trained from ground-truth bounding-boxes. We consider this a very encouraging result, given that we are not aware of previous methods demonstrated capable of localizing objects on the PASCAL07 test set when trained in a weakly supervised setting. Fig. 6 visually compares two models trained from the output of our method to the corresponding models trained from ground-truth bounding-boxes.

9 Conclusion

We presented a technique for localizing objects of an unknown class and learning an appearance model of the class from weakly supervised training images. The proposed model starts from generic knowledge and progressively adapts more and more to the new class. This allows to learn from highly cluttered images with strong scale and appearance variations between object instances. We also demonstrated how to use our method to train a fully supervised object detector from weakly supervised data.

References

1. Arora, H., Loeff, N., Forsyth, D., Ahuja, N.: Unsupervised segmentation of objects using efficient learning. In: CVPR. (2007)
2. Crandall, D.J., Huttenlocher, D.: Weakly supervised learning of part-based spatial models for visual object recognition. In: ECCV. (2006)
3. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR. (2003)
4. Galleguillos, C., Babenko, B., Rabinovich, A., Belongie, S.: Weakly supervised object localization with stable segmentations. In: ECCV. (2008)
5. Todorovic, S., Ahuja, N.: Extracting subimages of an unknown category from a set of images. In: CVPR. (2006)
6. Winn, J., Jojic, N.: LOCUS: learning object classes with unsupervised segmentation. In: ICCV. (2005)
7. Nguyen, M., Torresani, L., de la Torre, F., Rother, C.: Weakly supervised discriminative localization and classification: a joint learning process. In: ICCV. (2009)

8. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. *PAMI* (2009) in press.
9. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 Results (2007)
10. Borenstein, E., Ullman, S.: Learning to segment. In: *ECCV*. (2004)
11. Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: *CVPR*. (2006)
12. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: *CVPR*. (2007)
13. Dorkó, G., Schmid, C.: Object class recognition using discriminative local features. Technical Report RR-5497, INRIA - Rhone-Alpes (2005)
14. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV* (2007)
15. Cao, L., Li, F.F.: Spatially coherent latent topic model for concurrent segmentation and classification of objects and scene. In: *ICCV*. (2007)
16. Lee, Y.J., Grauman, K.: Shape discovery from unlabeled image collections. In: *CVPR*. (2009)
17. Kim, G., Torralba, A.: Unsupervised detection of regions of interest using iterative link analysis. In: *NIPS*. (2009)
18. Russel, B.C., Torralba, A.: LabelMe: a database and web-based tool for image annotation. *IJCV* **77** (2008) 157–173
19. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.: Self-taught learning: transfer learning from unlabeled data. In: *ICML*. (2007)
20. Thrun, S.: Is learning the n-th thing any easier than learning the first? In: *NIPS*. (1996)
21. Lando, M., Edelman, S.: Generalization from a single view in face recognition. In: Technical Report CS-TR 95-02, The Weizmann Institute of Science. (1995)
22. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In: *CVPR Workshop of Generative Model Based Vision*. (2004)
23. Stark, M., Goesele, M., Schiele, B.: A shape-based object class model for knowledge transfer. In: *ICCV*. (2009)
24. Tommasi, T., Caputo, B.: The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories. In: *BMVC*. (2009)
25. Alexe, B., Deselaers, T., Ferrari, V.: What is an object ? In: *CVPR*. (2010)
26. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. *SIGGRAPH* **23** (2004) 309–314
27. Ramanan, D.: Learning to parse images of articulated bodies. In: *NIPS*. (2006)
28. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. *PAMI* **28** (2006) 1568 – 1583
29. Dalal, N., Triggs, B.: Histogram of Oriented Gradients for Human Detection. In: *CVPR*. (2005)
30. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. *PAMI* (2009) in press.
31. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV* **42** (2001) 145–175
32. Bay, H., Ess, A., Tuytelaars, T., van Gool, L.: SURF: Speeded up robust features. *CVIU* **110** (2008) 346–359
33. Everingham, M., Van Gool, L., Williams, C.K.I., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2006 (VOC2006) (2006)