ETH zürich

ETH Library

Weakly Supervised Localization and Learning with Generic Knowledge

Journal Article

Author(s):
Deselaers, Thomas; Bogdan, Alexe; Ferrari, Vittorio

Publication date:
2012-12

Permanent link:
https://doi.org/10.3929/ethz-b-000053937

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
International Journal of Computer Vision 100(3), https://doi.org/10.1007/s11263-012-0538-3

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

Int J Comput Vis (2012) 100:275–293
DOI 10.1007/s11263-012-0538-3

Weakly Supervised Localization and Learning with Generic Knowledge

Thomas Deselaers · Bogdan Alexe · Vittorio Ferrari

Received: 24 July 2011 / Accepted: 11 May 2012 / Published online: 30 May 2012
© Springer Science+Business Media, LLC 2012

Abstract Learning a new object class from cluttered training images is very challenging when the location of object instances is unknown, i.e. in a weakly supervised setting. Many previous works 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 exploit generic knowledge learned beforehand from images of other classes for which location annotation is available. Generic knowledge facilitates learning any new class from weakly supervised images, because it reduces the uncertainty in the location of its object instances. 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 several datasets, including the very challenging PASCAL VOC 2007. Furthermore, our method allows training any state-of-the-art object detector in a weakly supervised fashion, although it would normally require object location annotations.

Keywords Object detection · Weakly supervised learning · Transfer learning · Conditional random fields

T. Deselaers (✉) · B. Alexe · V. Ferrari
Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland
e-mail: deselaers@vision.ee.ethz.ch

B. Alexe
e-mail: bogdan@vision.ee.ethz.ch

V. Ferrari
e-mail: ferrari@vision.ee.ethz.ch

Present address:
T. Deselaers
Google, Zurich, Switzerland

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 to reduce the manual labeling effort to learn object classes (Bagon et al. 2010; Chum and Zisserman 2007; Crandall and Huttenlocher 2006; Galleguillos et al. 2008; Kim and Torralba 2009; Nguyen et al. 2009). 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 to select 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 the CRF employs generic knowledge to guide the selection process as it *reduces the location uncertainty*. Over the iterations the CRF progressively adapts to the new class, learning more and

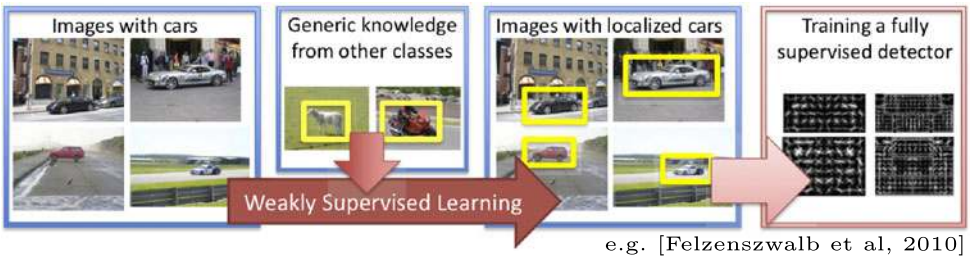


Fig. 1 Learning scenario. Starting from weakly supervised images of a new class, we localize its object instances while learning an appearance model of the class. In order to support learning this new class, we use Generic Knowledge learned beforehand from other classes. Our method can be used to produce bounding-boxes for training any fully supervised object detector

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 (Everingham et al. 2007) (Figs. 8, 9).

The main contribution of this paper is a novel method to jointly localize and learn a new class from WS data. Therefore, in Sect. 6 we directly evaluate the performance of our method by measuring how well it localizes instances of a new class in WS training images. We compare to various baselines and three existing methods (Chum and Zisserman 2007; Kim and Torralba 2009; Russell et al. 2006). Moreover, we also demonstrate an application of our method: we train the fully supervised model of Felzenszwalb et al. (2010) from objects localized by our method, evaluate it on a test set, and compare its performance to the original model trained from ground-truth bounding-boxes. These experiments show that our method enables training good object detectors from weakly supervised datasets, even when they consist of highly challenging images.

1.1 Related Work

Weakly Supervised Learning of Object Classes We focus here on WSL methods to learn object classes (i.e. requiring no object locations). Many approaches are based on a *bag-of-word* model for the entire image (Dorkó and Schmid 2005; Zhang et al. 2007). Although they have demonstrated impressive classification performance (Everingham et al. 2007), they are usually unable to localize objects.

There are several WSL methods that achieve localization. In Table 1 we summarize the main characteristics of many popular approaches. Two major families are part-based models (Crandall and Huttenlocher 2006; Fergus et al. 2003), and segmentation-based models (Alexe et al. 2010a; Arora et al. 2007; Cao and Li 2007; Galleguillos et al. 2008; Russell et al. 2006; Todorovic and Ahuja 2006; Winn and Jojic 2005a), and a wide variety of other techniques have been proposed (Bagon et al. 2010; Chum and Zisserman 2007; Lee and Grauman 2009a; Nguyen et al. 2009). However, most methods have been demonstrated on datasets such as CALTECH4 (Arora et al. 2007; Crandall and Huttenlocher 2006; Fergus et al. 2003; Galleguillos et al. 2008; Lee and Grauman 2009a; Nguyen et al. 2009; Winn and Jojic 2005a), Weizmann horses (Borenstein and Ullman 2004; Cao and Li 2007; Winn and Jojic 2005a), or CMU Faces (Nguyen et al. 2009). The objects in such datasets 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.

The field has made significant progress in recent years, as several methods have tried to go beyond and experiment on more challenging datasets, such as ETHZ Shape Classes (Bagon et al. 2010; Lee and Grauman 2009a), PASCAL VOC 06 (Chum and Zisserman 2007; Kim and Torralba 2009), and LabelMe (Russell et al. 2006). However, often the authors reduce the difficulty of the dataset by manually providing information about the scale of the target objects (Bagon et al. 2010; Lee and Grauman 2009a), their location (Lee and Grauman 2009a), or select easier subsets of images with dominant objects (Chum and Zisserman 2007). Blaschko et al. (2010) report experiments on the *cat* class from PASCAL VOC 07 in a semi-supervised setting, where their method is given the location of some of the target objects. Russell et al. (2006) automatically segment out regions similar across many images from the difficult LabelMe dataset (Russel and Torralba 2008), but reports that it is very hard to find small objects such as cars in it. Chum and Zisserman (2007) is especially 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 only a modest portion of the images and for classes such as horses, for which local texture features have little discriminative power. Kim and Torralba (2009) cluster windows of similar appearance using link analysis techniques. We quantitatively compare to Chum and Zisserman (2007), Russell et al. (2006) in Sect. 6, and to Kim and Torralba (2009) in Sect. 6.3.

As summarized in Table 1, methods are evaluated with a variety of different measures. In this work we are particularly interested in evaluating the ability of a method

Table 1 Overview of methods for weakly supervised learning of object classes. For each paper we give the type of approach, the datasets used for evaluation, the information given at training time, what is evaluated on training and test data, whether the approach handles objects at different scales at training time, and mention main limitations

Work	Approach	Datasets	Training information	Evaluate on		Scale changes
				train img.	test img.	
Fergus et al. (2003)	Parts	C4	CVP	no	Classif	
Fei-Fei et al. (2003)	Parts	C4	CVP+MT	no	Classif	
Borenstein and Ullman (2004)	Seg	WH, C4	CVP	Segm	no	no
Fei-Fei et al. (2004)	Parts	C101	CVP+MT	no	Classif	
Winn and Jojic (2005a)	Seg+Gen	C4, W	CVP	Segm	no	
Russell et al. (2006)	Seg+Topic	C4, MSRC, L	Unlabeled	Segm	no	yes
Todorovic and Ahuja (2006)	Seg	C4, UIUC	CVP	no	Det	
Fritz and Schiele (2006)	Parts	TUD, UIUC	CVP	no	Det	
Crandall and Huttenlocher (2006)	Parts	C4, GB	CVP+Scale	no	Classif	
Grauman and Darrell (2006)	Shape+Clust	C4	Unlabeled	Purity	Classif	
Arora et al. (2007)	Seg+CRF	C4	CVP	no	Classif+Segm	
Chum and Zisserman (2007)	Exemplar	P6-DO	CVP	no	Det	
Cao and Li (2007)	Seg+Topic	W, C4, C101*	Unlabeled for C4, CVP for W	Segm, Classif	no	
Galleguillos et al. (2008)	Seg+MIL	C4	CVP	no	Classif	
Lee and Grauman (2009b)	Clust	C101*, MSRC	CVP	FD	no	
Lee and Grauman (2009a)	Shape+Clust	C4, ETHZ, L	Unlabeled for C4, BB for ETHZ	Purity+BBHR	Det	
Nguyen et al. (2009)	BoVW+MIL	C4, CMU, X	CVP+Scale	no	Classif	
Kim and Torralba (2009)	Clust+LA	P6	C	Det	no	yes
Bagon et al. (2010)	SS+TM	ETHZ, X	CVP+Scale	no	Det	no
Alexe et al. (2010a)	Seg+CRF	W, C4, C101*	CVP+MT	Segm	no	
Payet and Todorovic (2010)	Shape+Clust	ETHZ, W, C101*	Unlabeled	Purity+BBHR	no	yes
Blaschko et al. (2010)	StructSVM	INRIA, P7-cat	CVP+Semi	no	Det	yes
this paper	CRF+GK	C4, P6, P7	CVP+MT	CorLoc	Det	yes

Approach: Parts: part-based, Topic: topic models, Gen: other generative model, Exemplar: exemplar model, Clust: clustering, LA: link analysis, StructSVM: structural SVM, CRF: conditional random field, MIL: multiple instance learning, TM: template matching, Seg: segmentation-based, Shape: contour descriptors, BoVW: bag of visual words, SS: self-similarity features

Datasets: C4: CALTECH4, C101: CALTECH101, C101*: a subset of C101 with 4-28 classes. GB: Graz bicycles, W: Weizmann Horses, WH: Weizmann Horses cropped to heads, MSRC: Microsoft Research Cambridge segmentation database, L: LabelMe subset, CMU: CMU faces, ETHZ: ETHZ Shape Classes, TUD: TU Darmstadt motorbikes and cows, UIUC: UIUC cars, INRIA: INRIA person detection dataset, X: private dataset, P6: PASCAL VOC 06, P7: PASCAL VOC 07, P6-DO: A subset of P6 with 6 classes (car, bicycle, bus, motorbike, cow, sheep). About 20 images per class manually selected. Most of them with large dominant objects. P7-cat: only the *cat* class from P7

Training information: CVP: images contain objects of the target class in roughly the same viewpoint; C: images contain objects of the target class. Scale: the size of the target objects is given to the algorithm; BB: images cropped around the bounding-box of the target object, to a fixed region relative to the object size; Unlabeled: unlabeled images with multiple categories (object discovery setting); Semi: object locations given for some images. MT: external meta-training data from *other* classes given

Evaluate on (training/test images): Purity: how well the learner clusters training images into object classes (discovery setting only); Segm: pixelwise accuracy of foreground/background segmentation; CorLoc: percentage of correctly localized objects up to a BB (Sect. 6.2); BBHR: Bounding-box Hit Rate, measuring the percentage of local features labeled as the object that fall into the ground-truth BB. It does not measure localization of *whole objects*; Classif: object present/absent classification on test images; Det: detection accuracy on test images (captures both whole-image classification and localization up to a BB); FD: weighted ratio of features on objects and background; no: no evaluation reported. Overall, only methods tagged with Segm, CorLoc, BBHR, Det evaluate localization in some form. Only methods tagged with CorLoc, Det evaluate localization of whole objects

Scale changes: no: the method is described as not supporting multiple scales. yes: the method is described as supporting multiple scales and the evaluation gives evidence for it. If neither yes nor no: the evaluation does not show scale changes, but the method could potentially support them

to localize objects. Several previous works evaluate their method indirectly, as the performance of the learned model on a separate set of test images. In several cases, test time performance is evaluated only as whole image classification (Fergus et al. 2003), while other works evaluate localization (Bagon et al. 2010). Conversely, some works evaluate how well their method localizes objects in the training images, but do not try the learned model on novel test images, e.g. Arora et al. (2007), Winn and Jojic (2005a). In this paper, we evaluate localization both *directly* on the training images, as well as on novel test images (by training the model of Felzenszwalb et al. (2010) from the output of our method). Moreover, to the best of our knowledge, we are the first to demonstrate weakly supervised learning of object categories on the very challenging PASCAL07 dataset.

Transfer Learning in Computer Vision Our use of generic knowledge is related to previous work on transfer learning (Raina et al. 2007; Thrun 1996) in computer vision, where learning the new class (target) is helped by labeled examples of other related classes (sources) (Deselaers et al. 2010; Fei-Fei et al. 2004; Lampert et al. 2009a; Lando and Edelman 1995; Quattoni et al. 2008; Rohrbach et al. 2010; Stark et al. 2009; Tommasi and Caputo 2009; Tommasi et al. 2010; Torresani et al. 2010).

Transfer learning for visual recognition is a relatively new trend, but it is gaining increasing attention. One of the earliest works, Lando and Edelman (1995) learns a new face from just one view, supported by images of other faces. Fei-Fei et al. (2003) learn priors on parameters of a part-based classifier from a set of mixed classes, and then incorporate these priors when learning a new class using a Bayesian approach. These priors are a form of generic knowledge. They help biasing the *parameters* of the model of the target class. Instead our generic knowledge is designed to help *localizing* objects of the target class in their training images. Fei-Fei et al. (2004) extends Fei-Fei et al. (2003) to sequentially update a part-based classifier trained on source classes to fit the target class. Stark et al. (2009) transfer shape knowledge from one manually selected source class to the target class. Tommasi and Caputo (2009) use the parameters of the SVM for one source class as a prior for the target class. Their follow-up work (Tommasi et al. 2010) transfers from multiple source classes automatically selected by minimizing a leave-one-out error on the training set of the target class. Lampert et al. (2009a) transfer knowledge from 40 animal classes through an intermediate attribute layer. The lists of which attributes belong to which class are manually defined. Rohrbach et al. (2010) improve by automatically compiling these lists through text mining on the Internet (e.g. counting the number of occurrences of an attribute-noun pair such as 'striped tiger'). They also present a model where the amount of transfer is guided by the semantic similarity between the names of the source and target classes.

Most previous work on transfer learning in CV learn models for *classifying* an entire image as containing the target class or not. Our method instead learns models capable of *localizing* objects up to a bounding-box. This is a harder task (Everingham et al. 2010), especially when bounding-boxes are not available for training. To achieve this, we transfer a substantially different kind of knowledge, which reduces the location uncertainty of the target class in its training images. Automatically localizing instances of the new class in training images is the central objective of our work. Moreover, previous works aim at reducing the number of images necessary to learn the target class, improving generalization from a few examples. Here instead, we reduce the *degree of supervision* from object bounding-boxes to image labels. Finally, the above works transfer knowledge from source classes *related* to the target class, whereas our generic knowledge provides a broad basis on top of which it is easier to learn *any* new class.

Multiple-Instance Learning Our method is also related to multiple-instance learning (Andrews et al. 2002; Chen et al. 2006; Viola et al. 2005), if we represent an image as a bag and the windows therein as instances. We have shown in Deselaers and Ferrari (2010) how a generalization of the CRF proposed here can be used for multiple-instance learning in general problems. Note however, that in this paper we are not interested in bag classification but in automatically selecting a positive instance in each positive bag (which gives the localization of the object class).

1.2 Plan of the Paper

Our new CRF model is described in Sect. 2. In Sect. 3 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. Section 4 details the generic knowledge that is incorporated into the process and how it is obtained. Section 5 describes the image cues we use and in Sects. 6–7 we experimentally evaluate the method.

2 The CRF Model for Localizing 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 the energy of a conditional random field (CRF) defined globally over all training images (Eq. (2)). Ideally the energy is minimal when all selected windows contain an object of the same class.

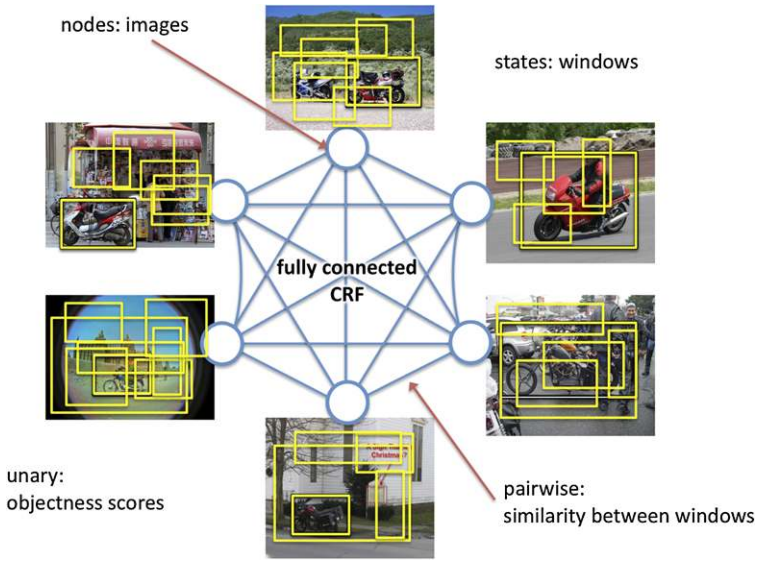


Fig. 2 The localization model is a fully connected CRF where each training image is a node. The state space of a node is the set of windows in the image. The unary potential measures how likely a window is to contain an object of *any* class. The pairwise potential measures how likely two windows are to contain objects of the same, but unknown, class

Initially the CRF is driven by *class-generic* knowledge (GK) that is learned beforehand from meta-training data (Sect. 4). GK guides the initial selection of windows on the training images of the target class (*localization* stage, Sect. 3.1). Next, we use the selected windows to learn appearance and shape models specific to the target class, and incorporate them as new terms in the CRF (*learning* stage, Sect. 3.2). In the next iteration we optimize the updated CRF to refine the selection of windows. Alternating the localization and learning stages progressively transforms the CRF from a class-generic object localizer into one specialized to the target class. The two stages help each other, as better localization leads to more accurate class-specific models, which in turn sharpens localization. This combination allows for WSL on highly cluttered images with strong scale and appearance variations (Sect. 6).

2.1 Configuration of Windows L

The set of training images $\mathcal{I} = (I_1, \ldots, I_N)$ is represented as a fully connected CRF (Fig. 2). 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 $L = (l_1, \ldots, l_N)$ can be written as

$$p(L|\mathcal{I}, \Theta) \propto \exp\big(-E(L|\mathcal{I}, \Theta)\big) \tag{1}$$

$$\text{with } E(L|\mathcal{I}, \Theta) = \sum_n \rho_n \Phi(l_n|I_n, \Theta) \tag{2}$$

$$+ \sum_{n,m} \rho_n \rho_m \Psi(l_n, l_m|I_n, I_m, \Theta) \tag{3}$$

where each l_n is a window in image I_n. More precisely, l_n is an index into a list of candidate windows for image I_n (Sect. 4.1); Θ are the parameters of the CRF; ρ_n is the confidence for image I_n, weighting its impact on the overall energy (Sect. 3.2.3). $\Phi(l_n|I_n, \Theta)$ is a unary potential which describes the cost to select a window l_n in an image I_n (Sect. 2.2). $\Psi(l_n, l_m|I_n, I_m, \Theta)$ is a pairwise potential which assigns a cost to selecting window l_n in image I_n and window l_m in image I_m (Sect. 2.3).

For reference, we give an overview over the notation used for the model components in Table 2.

2.2 The Unary Potential Φ

The unary $\Phi(l_n|I_n, \Theta)$ measures how likely an image window l_n is to contain an object of the target class

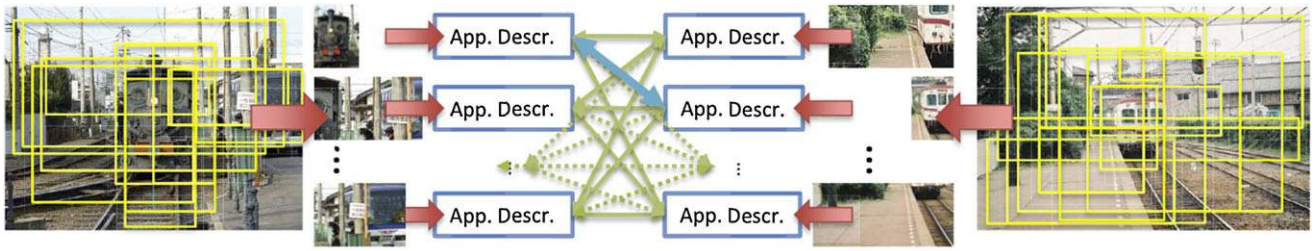
$$\Phi(l_n; I_n) = \alpha_\Omega \Omega(l_n|I_n, \theta_\Omega) + \alpha_\Pi \Pi(l_n|\theta_\Pi) + \sum_f \alpha_{\Upsilon_f} \Upsilon_f(l_n|I_n, \theta_{\Upsilon_f}) \tag{4}$$

It is a linear combination of:

- Ω: the likelihood that l_n contains an object of *any* class, rather than background (Alexe et al. 2010b) (Sect. 4.1);
- Π: a model of the overall shape of the windows, specific to the target class (Sect. 3.2.2);
- Υ_f: appearance models, one for each cue f, specific to the target class (Sect. 3.2.1). In our experiments we consider four appearance cues: GIST, color histograms, bag of words, and HOG (Sect. 5).

The scalars α weight the terms.

Note how Π, Υ carry knowledge specific to the target class. They are initially unknown and set to uniform values. They are learned after the first localization stage and then used in all subsequent iterations (Sects. 3.2.1, 3.2.2).



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

Fig. 3 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 dissimilarity cues Γ_f and the aspect-ratio dissimilarity Λ

Table 2 Notation used throughout the paper

Symbol	Meaning	Description
L	configuration of windows $(l_1, \ldots, l_N)$	2.1
$\mathcal{I}$	set of training images $(I_1, \ldots, I_N)$	2.1
I_n	one image	2.1
l_n	on window/state in image I_n	2.1
Θ	parameters of CRF model	2.1
ρ_n	confidence for image I_n	3.2.3
$\Phi(l_n \| I_n, \Theta)$	unary potential of CRF	2.2
$\Psi(l_n, l_m \| I_n, I_m, \Theta)$	pairwise potential of CRF	2.3
α	weights for terms in the model	2.2, 2.3
$\Omega(l_n \| I_n, \theta_\Omega)$	objectness term	4.1
$\Pi(l_n \| \theta_\Pi)$	class-specific shape model	3.2.2
$\Upsilon_f(l_n \| I_n, \theta_{\Upsilon_f})$	class-specific appearance model	3.2.1
$\Lambda(l_n, l_m \| \theta_\Lambda)$	shape dissimilarity between two windows	4.2
$\Gamma_f(l_n, l_m \| I_n, I_m)$	appearance dissimilarity	4.3

2.3 The Pairwise Potential Ψ

The pairwise potential $\Psi(l_n, l_m | I_n, I_m, \Theta)$ measures the dissimilarity between two windows, assessing how likely they are to contain objects of the same class (Fig. 3)

$$\Psi(l_n, l_m | I_n, I_m, \Theta) = \alpha_\Lambda \Lambda(l_n, l_m | \theta_\Lambda) + \sum_f \alpha_{\Gamma_f} \Gamma_f(l_n, l_m | I_n, I_m) \quad (5)$$

It is a linear combination of

- Λ: a prior on the shape dissimilarity between two windows l_n, l_m. It depends only on the states l_n, l_m, not on the image content (Sect. 4.2);
- Γ_f: a potential measuring the appearance dissimilarity between l_n and l_m according to multiple cues f. It depends on the image content (Sect. 4.3).

The scalars α weight the terms. Figure 3 illustrates the computation of the pairwise potential for every pair of windows between two images.

2.4 The Parameters $\theta_\Omega, \theta_\Lambda$

The parameters θ_Ω, θ_Λ of the individual terms and the weights α carry generic knowledge and are learned from the meta-training data (Sect. 4). The class-specific models Π, Υ and the image confidences ρ_n carry class-specific knowledge and are initially unknown. During the first localization stage we set them to uniform. They are progressively adapted to the target class over the following iterations during the learning stage (Sect. 3.2).

Note that our model connects nodes (windows) *between* images, rather than elements *within* an image as is typically done for CRFs in other computer vision domains (e.g. pixels in segmentation (Rother et al. 2004), body parts in human pose estimation (Ramanan 2006)).

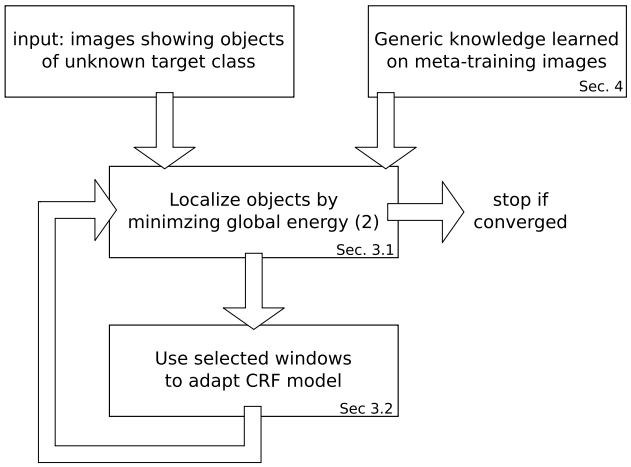


Fig. 4 Localization and learning. The localization and learning stages are alternated. *Localization*: one window is selected among the candidates for each image (Sect. 3.1); *Learning*: the CRF model is adapted to the target class. (Sect. 3.2). These two steps are alternated until convergence, i.e. the selected windows remain the same between two iterations

3 Localization and Learning

When given a set of images $\mathcal{I}$ of a target class the goal is to localize its object instances and learn a model of the class. Initially our CRF is driven by generic knowledge, which was learned beforehand in the meta-training stage (Sect. 4). This drives the first *localization* stage (Sect. 3.1) that attempts to select windows covering instances of the target class. Next, these windows are used to *learn* knowledge specific to the target class, which is then incorporated into the CRF (Sect. 3.2). The localization and learning stages are alternated, optimizing one while keeping the other fixed, thus progressively adapting the CRF to the target class (Fig. 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 (Felzenszwalb et al. 2010) are commonly used to learn in the presence of latent variables (in our case L^*).

3.1 Localization

Localizing objects corresponds to finding the configuration L^* that minimizes the global energy (2):

$$L^* = \arg\min_L \{E(L|\mathcal{I}, \Theta)\} \tag{6}$$

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

Optimizing this energy exactly is impractically expensive (complexity $O(W^{|\mathcal{I}|})$, with W the average number of windows in an image). Exact inference is inefficient because the CRF is fully connected, has arbitrary non-submodular pairwise potentials, and the nodes have huge state spaces (potentially all windows in the images).

Therefore we use the objectness measure of Alexe et al. (2010b) as a location prior. We randomly sample 100 windows per image proportionally to their probability of containing an object and use only these as states (Sect. 4.1). We now approximate the global optimum of the model in this reduced state space using the tree-reweighted message passing algorithm TRW-S (Kolmogorov 2006a). This has complexity $O(kW|\mathcal{I}|)$, with k a small number of iterations (typically $k < 10$). TRW-S also returns a lower bound on the energy. When this coincides with the returned solution, we know it found the global optimum of the model in the reduced state space. 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 computed configurations L^* are very close to the global optimum.

3.2 Learning

Based on the selected windows L^*, we adapt several components of the CRF to the target class:

- the class-specific appearance models Υ_f (Sect. 3.2.1),
- the class-specific shape model Π (Sect. 3.2.2),
- the image confidences ρ_n (Sect. 3.2.3), and
- the weights α of the cues (Sects. 3.2.4, 3.2.5).

During this stage the CRF is progressively adapted from generic to class-specific. This adaptation involves an additional negative image set $\mathcal{N}$, which does not contain any object of the target class.

3.2.1 Class-Specific Appearance Models Υ_f

Any model trainable from annotated object windows could be used here (e.g. Dalal and Triggs 2005; Felzenszwalb et al. 2010; Lampert et al. 2009b). 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 (Gaidon et al. 2009). First, the SVM θ_{Υ_f} is trained to separate all windows L^* from windows randomly sampled from $\mathcal{N}$. Then, this SVM is used to score every selected window $l_n^* \in L^*$. The top scored κ % windows are then used to retrain θ_{Υ_f}. In our experiments we use $\kappa = 50$ and repeat this procedure 10 times. As explained in Gaidon et al. (2009) this iterative procedure brings the benefit of cleaning up the training set, by ranking low windows not belonging to the target class.

After training the SVMs, we set the energy $\Upsilon_f(l_n|I_n, \theta_{\Upsilon_f})$ of a candidate window l_n in Eq. (4) to the signed distance

between the SVM hyperplane and the appearance descriptor $l_n^f(I_n)$ of l_n:

$$\Upsilon_f(l_n|I_n,\theta_{\Upsilon_f}) = \beta_{\Upsilon_f} + \theta_{\Upsilon_f} l_n^f(I_n) \tag{7}$$

where β_{Υ_f} is the bias term of the SVM. The SVM is trained such that the selected windows are class "−1", and the negative windows are class "+1" aiming for the SVM to give a low energy to windows that are classified as "selected".

3.2.2 Class-Specific Shape Model Π

The class-specific shape model $\Pi(l_n|\theta_\Pi)$ models the aspect-ratio of the target class as an univariate Gaussian with parameters $\theta_\Pi = \{\mu_\Pi, \sigma_\Pi\}$

$$p(l_n|\Theta_\Pi) = \frac{1}{\sqrt{2\Pi\sigma_\Pi^2}} \exp\left(\frac{|\mu_\Pi - l_n^\Pi|^2}{\sigma_\Pi^2}\right) \tag{8}$$

where l_n^Π is the aspect-ratio of window l_n (i.e. width divided by height). We learn μ_Π, σ_Π to fit the distribution of the aspect-ratios of the selected windows L^*, according to the maximum-likelihood criterion.

After learning this Gaussian, we set the energy $\Pi(l_n|\Theta_\Pi)$ of a candidate window l_n in Eq. (4) to

$$\Pi(l_n|\Theta_\Pi) = -\log(p(l_n|\Theta_\Pi) \tag{9}$$

3.2.3 Image Confidences ρ_n

The image confidences ρ_n emphasize images where the model is confident of having localized an object of the target class (Eq. (2)). We set ρ_n proportional to the negative energy of a selected window l_n^* according to the class-specific appearance model

$$\rho_n \propto -\sum_f \left(\alpha_{\Upsilon_f} \Upsilon_f\left(l_n^*|I_n,\theta_\Upsilon\right)\right) \tag{10}$$

The class-specific appearance model has a high confidence on images where the object is localized accurately and can be easily recognized (i.e. it has a large negative distance from the SVM hyperplane, Eq. (7)). Such images receive a high confidence. Conversely, it gives a low confidence (i.e. high energy) to images where the object is either not well localized or is difficult to recognize (e.g. poor illumination conditions), such images receive a low confidence. This reduces the impact of particularly difficult images and makes the model more robust to incorrect selections in L^*. The image confidences ρ_n are linearly scaled so that the image with the highest confidence has $\rho = 2.0$, and the image with the lowest confidence has $\rho = 0.5$. Note how the confidences implicitly adapt every term in the CRF toward the target class.

3.2.4 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, sheep using color, mugs using shape-gradient features). Here we adapt to the target class the weights α_{Υ_f} of the class-specific appearance models Υ_f.

To determine the discriminative power of the individual appearance models Υ_f, we train a linear SVM w on the space of vectors of appearance scores $[\Upsilon_f(l_n|I_n,\theta_{\Upsilon_f})]$. As in our experiments we use 4 appearance cues, these vectors are of length 4 (Sect. 5). As positive training data we use the κ % of the selected windows L^* which have the highest score according to the unary models Υ_f (i.e. the highest confidence of covering an object the target class). As negative training data we randomly sample windows from $\mathcal{N}$. The trained SVM hyperplane w gives higher weights to cues that are particularly suited to discriminate windows of the target class from other windows.

After learning the hyperplane w, we update the weights α_{Υ_f} to $\alpha_{\Upsilon_f} \leftarrow \frac{1}{2}(\alpha_{\Upsilon_f} + w(f))$, where $w(f)$ is the weight of cue f.

3.2.5 Pairwise Appearance Cue Weights α_{Γ_f}

We proceed analogously to Sect. 3.2.4. To determine the importance of the pairwise appearance cues, we train a linear SVM on vectors of pairwise appearance similarities $[\Gamma_f(l_n,l_m|I_n,I_m)]$. As positive training data we use the appearance similarities between all pairs of the top κ % selected windows. As negative training data we use (a) appearance similarities between pairs of one positive window and one negative window (sampled from $\mathcal{N}$), (b) appearance similarities between all pairs of negative windows.

After training the SVM, the weights α_{Γ_f} are updated in the same manner as in Sect. 3.2.4.

3.2.6 Other Terms

The objectness Ω, the shape dissimilarity Λ, and the appearance dissimilarity Γ_f terms are not explicitly adapted to the target class. However, their impact on the overall energy (Eq. (2)) is adapted through the weights $\alpha_{\Upsilon_f}, \alpha_{\Gamma_f}$, and the image confidences ρ_n.

3.3 Discussion

3.3.1 Convergence

Our overall algorithm is defined by two decoupled optimization problems: localization and learning. The algorithm terminates when two consecutive localization steps return the same selection of windows. In our experiments this always happened within 10 iterations.

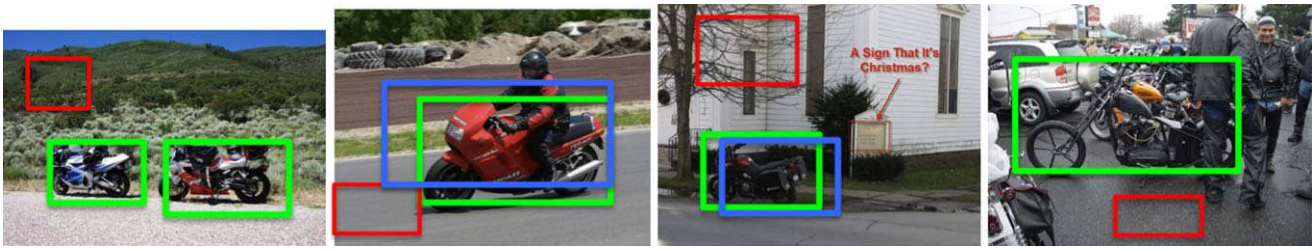


Fig. 5 Ideal behavior of the objectness measure. The objectness score for a window should be highest when fitting an object tightly (*green*), lower when covering objects partially (*blue*), and lowest when containing only background (*red*)

3.3.2 Optimality of the Localization Phase

The localization optimization problem is not solved globally optimally because minimizing the energy of our fully connected CRF is impractical (Kolmogorov 2006b). However, as described in Sect. 3.1, the approximation we obtain with TRW-S is very close to the global optimum.

3.3.3 Optimality of the Learning Phase

The class-specific parameters are trained optimally according to their respective training criteria:

parameters of the class-specific appearance models:
the class-specific appearance models are SVMs and thus their training problem is convex.

class-specific shape model:
the class-specific shape model is a single Gaussian, which is easily trained according the maximum likelihood criterion.

3.3.4 Runtime

Running the entire method on a set of 100 images takes about 10 hours using our unoptimized, single-threaded Matlab implementation. Most of the time is spent in feature extraction (total: 7.5 h; per image: 4 sec for objectness; 10 sec for GIST, 80 sec for CHIST, 180 sec for SURF, 5 sec for HOG). After feature extraction, computing the pairwise potentials takes a total of 2 h. Finally, one iteration of localization and learning takes about 1 min (<2 seconds for localization; about one minute for the learning step).

4 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 $\mathcal{M}$. These contain objects of known classes annotated with bounding-boxes.

4.1 Objectness Ω

We use the objectness measure $\Omega(l|I,\theta_\Omega)$ of Alexe et al. (2010b), 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. The ideal behavior of the objectness measure is shown in Fig. 5.

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 objectness probability. These form the set of states for node I (i.e. the candidate windows the CRF can choose from). The objectness probability forms the unary term $\Omega(l|I,\theta_\Omega)$.

This procedure brings two advantages. First, it greatly reduces the computational complexity of minimizing (2), which is quadratic in the number of states (there are $\simeq 10^8$ windows in an image (Lampert et al. 2009b)). Second, the sampled windows and their scores Ω attract the CRF toward selecting objects rather than background windows. This is crucial in a WSL setup, as typically the background contains frequently recurring appearance patterns with low variability between images. Importantly, this variability is ofter smaller than that among the actual object instances, antagonizing the learner. Therefore, our use of objectness steers the CRF away from trivial solutions, e.g. where all selected windows cover a piece of sky in airplane training images (Nguyen et al. 2009), or a piece of road in motorbikes images. In Sect. 6 we evaluate objectness quantitatively.

We note that as an alternative to the objectness measure of Alexe et al. (2010b), we could also have used the related methods of Endres and Hoiem (2010) or Carreira et al. (2010).

4.2 Pairwise Shape Dissimilarity Λ

θ_Λ is learned as the Bayesian posterior $\Lambda(l_n,l_m|\theta_\Lambda) = -\log p(l_n \overset{c}{=} l_m|\mathrm{AD}(l_n,l_m))$ from many window pairs con-

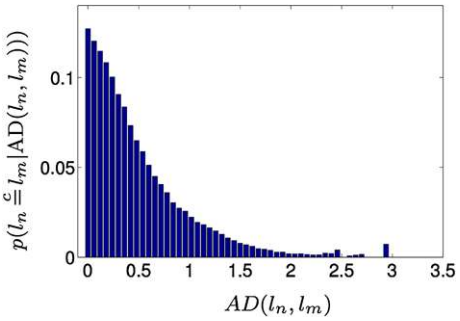


Fig. 6 Pairwise shape dissimilarity model Λ: $p(l_n \stackrel{c}{=} l_m|\mathrm{AD}(l_n,l_m))$ (*vertical axis*) as a function of $\mathrm{AD}(l_n,l_m)$ (*horizontal axis*). At the left-most point $\mathrm{AD}(l_n,l_m)=0$, i.e. l_n and l_m have the same aspect-ratio

taining the same ($l_n \stackrel{c}{=} l_m$) and different classes. The function $\mathrm{AD}(l_n,l_m)$ measures the aspect-ratio dissimilarity between windows l_n and l_m as

$$\mathrm{AD}(l_n,l_m)=\left|\log\left(\frac{w_n/h_n}{w_m/h_m}\right)\right| \tag{11}$$

where w_n, w_m are the widths and h_n, h_m the heights of windows l_n, l_m. We use a 60-bin histogram θ_Λ to represent this distribution. In practice this learns that instances of the same class have similar aspect-ratios (Fig. 6).

4.3 Pairwise Appearance Dissimilarity Γ_f

The pairwise appearance dissimilarity $\Gamma_f(l_n,l_m|I_n,I_m)$ assesses whether two windows l_n and l_m contain an object of the same class, *regardless of the class*. This is different from a distance measure assessing whether two images contain an object of the same *known* class, which is often addressed using distance learning methods (Babenko et al. 2009; Frome et al. 2007; Malisiewicz and Efros 2008; Weinberger et al. 2005). Another related, but also different task, is to decide whether two images show the *same object instance* (Nowak and Jurie 2007).

We evaluated several distance learning methods (Nowak and Jurie 2007; Weinberger et al. 2005) on the meta-training data and found that none of them outperformed a simple sum of squared distances (SSD) between appearance descriptors.

Our pairwise appearance dissimilarity Γ_f between two windows l_n, l_m in images I_n, I_m is computed as the SSD between their appearance descriptors $l_n^f(I_n), l_m^f(I_m)$:

$$\Gamma_f(l_n,l_m|I_n,I_m)=\left\|l_n^f(I_n)-l_m^f(I_m)\right\|^2 \tag{12}$$

4.4 Weights α

The overall goal of the methods in this section is to find weights α between the various terms of our CRF so as to maximize the number of meta-training images in which an object of the target class is localized correctly by our technique (Sect. 3). Following the spirit of the other GK components (Sects. 4.1–4.3), these weights are chosen jointly over *all* meta-training classes. Hence, these weights are in a good ballpark that tends to perform well in general, i.e. also on novel target classes. We determine the weights in a two-step scheme.

Step 1: weights for localization terms (Sect. 4.4.1). We determine the weights α_Ω, α_Λ, α_{Γ_f} so that the windows L^* returned by the localization stage (Sect. 3.1) best cover the meta-training bounding-boxes $\mathcal{M}$ (according to the criterion of Sect. 6.2). We achieve this using a constraint-generation algorithm inspired by structured output SVMs (Tsochantaridis et al. 2005). These weights are determined using only the localization stage, as they contain no class-specific knowledge.

Step 2: weights for class-specific terms (Sect. 4.4.2). The remaining weights α_Π, α_{Υ_f} cannot be directly learned in the constraint generation framework because the class-specific terms Π, Υ_f are adapted in every iteration (potentially depending on the weights). Instead, we first fix α_Ω, α_Λ, α_{Γ_f} in step 1, and then determine $\alpha_\Pi, \alpha_{\Upsilon_f}$ using grid-search to maximize localization performance on $\mathcal{M}$ after the localization and learning iterations (Sect. 3).

Note how it would be possible to determine all weights using a grid-search procedure (Deselaers et al. 2010), but the constraint generation algorithm in step 1 is more elegant and computationally much more efficient. However, it typically does not lead to better results than a grid-search with a sufficiently fine grid.

4.4.1 Constraint Generation

The goal is to find weights $\alpha=(\alpha_\Omega,\alpha_\Lambda,\alpha_{\Gamma_f})$ so that the configuration of windows with the lowest energy (2) correctly localizes one object in each meta-training image. Note how the total number of possible configurations L grows exponentially with the number of images, and how there may be many configurations localizing one object correctly in every image (though most will not).

Formally, we search for α so that

- there exists one configuration $\hat{L}$ that correctly localizes an object in every image
- the energy of $\hat{L}$ is *lower* than the energy of any configuration that does not

When these two criteria are met, the lowest energy configuration of the global energy function (2) maximizes localization performance. This will result in the optimal behavior of the localization stage (Sect. 3.1).

We learn α according to a max-margin criterion following the constraint-generation approach used to train structured output SVMs (Tsochantaridis et al. 2005), analogously

to other work on learning the parameters of a CRF (Deselaers and Ferrari 2010; Finley and Joachims 2008; Szummer et al. 2008). More precisely, we learn α by solving a generalized support vector training problem:

$$\begin{aligned} &\min_{\alpha,\xi} \frac{1}{2}\|\alpha\|^2 + C\sum_{k=1}^{K}\xi_k \\ &\text{s.t. } E\big(L|\mathcal{I}_k,\Theta^\alpha\big) - E\big(\hat{L}_k|\mathcal{I}_k,\Theta^\alpha\big) \geq \Delta(\hat{L}_k,L) - \xi_k, \quad (13) \\ &\quad \forall k, \forall L \neq \hat{L}_k \\ &\xi_k \geq 0, \alpha \geq 0 \end{aligned}$$

$\mathcal{I}_k$ is the set of meta-training images for class k and $\hat{L}_k$ is the configuration composed of ground-truth windows, each guaranteed to cover an instance of the class. This configuration achieves optimal localization performance and therefore it should have the lowest energy. $C > 0$ is a constant controlling the trade-off between training error minimization and margin maximization. In our experiments we set $C = 0.1$. Each ξ_k is a slack variable for class k. Θ^α are the parameters of the CRF according to the weight vector α.

The loss function $\Delta(\hat{L},L) = \sum_n (1 - \frac{\cap(\hat{l}_n,l_n)}{\cup(\hat{l}_n,l_n)})$ penalizes deviations from $\hat{L}$ ($\frac{\cap(\hat{l}_n,l_n)}{\cup(\hat{l}_n,l_n)} \in [0,1]$ denotes the intersection-over-union overlap between two windows). This loss function continuously gives smaller penalties to windows l_n which overlap more with the ground-truth $\hat{l}_n$. It better reflects the quality of localization and yields a smoother learning problem than a hard 0/1 loss giving 1 to all $l_n \neq \hat{l}_n$.

Note how solving (13) leads to a single weight vector α optimized over all meta-training classes combined. Therefore, α is a form of generic knowledge.

In this formulation, every possible configuration L yields a constraint, so the number of constraints is exponential in the number of images. Therefore, it is infeasible to consider all constraints explicitly while solving (13). The constraint generation technique (Tsochantaridis et al. 2005) only considers a small subset of constraints explicitly. Starting with an empty set of constraints, it iteratively adds the constraint which is most violated by the current setting of α.

First, note how each constraint correspond to exactly one configuration of windows. The configuration L^*, which violates the constraints the most is that one which has a lower energy than the desired configuration $\hat{L}$ and a high loss $\Delta(\hat{L},L)$.

It can be found by solving a subproblem of the same form as (6), but incorporating the loss $\Delta(\hat{L},L)$ as an additional term into E (Eq. (2)). Note how $\Delta(\hat{L},L)$ is a sum over the images in each meta-training class, and how each term in Δ depends only on the state of a single node in the CRF. Therefore, Δ can be incorporated into E as an additional *unary term*, leading to the following subproblem

$$L^* = \arg\min_L \big\{ E(L|\mathcal{I},\Theta) - \Delta(\hat{L},L) \big\} \quad (14)$$

Note that finding the most violating constraint L^* potentially has to be performed very often and therefore it must be found *efficiently*. As (14) has the same form as (6), it can be efficiently solved to a very good approximation using TRW-S (Sect. 3.1). Then, the most-violating configuration L^* is added to the set of active constraints, and then an updated weight vector α is found by minimizing (13) over the active constraints.

This procedure is iterated until $\hat{L}$ (the best possible configuration) is the minimum energy configuration. When this is achieved, all constraints are fulfilled and the procedure terminates.

In general, constraint generation is guaranteed to converge when the subproblem of finding L^* can be solved optimally (Tsochantaridis et al. 2005). Although we solve it approximately here, in all our experiments the constraint generation algorithm terminated in 20 to 50 iterations.

4.4.2 Grid Search

While keeping the weights $\alpha_\Omega, \alpha_\Lambda, \alpha_{\Gamma_f}$ fixed, we now determine the best possible $\alpha_\Pi, \alpha_{\Upsilon_f}$. As in Sect. 4.4.1, we aim at finding a generic set of weights maximizing the average localization performance jointly over all meta-training classes. To this end, we evaluate all combinations of weights $\alpha_\Pi, \alpha_{\Upsilon_f}$ on a 5D grid (1 dimension for α_Π, 4 dimensions for α_{Υ_f}). We retain the combination of weights $(\alpha_\Pi, \alpha_{\Upsilon_1}, \alpha_{\Upsilon_2}, \alpha_{\Upsilon_3}, \alpha_{\Upsilon_4})$ that, on average over all meta-training classes, leads to the best localization result after running our full method (Sect. 3).

4.5 Other Parameters

We briefly mention here how we set the remaining components of Θ from the meta-training data $\mathcal{M}$.

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 better. We set the regularization parameter $C = 1.0$ in our experiments.

Percentage κ of Images With the weights α and the SVM kernels fixed, we determine the percentage κ of selected windows to use for the iterative training in Sect. 3.2.1. We set κ to maximize localization performance on $\mathcal{M}$ after our full method.

Class-Specific Parameters The remaining parameters of the CRF are specific to the target class and are not learned from meta-training data, i.e. the class-specific appearance models Υ_f, the class-specific shape model Π, and the image confidences ρ_n. They are initially unknown and set uniformly.

5 Appearance Cues

We extract four appearance descriptors f from each candidate window and use them to calculate the appearance similarities Γ_f and the class-specific appearance scores Υ_f.

GIST (Oliva and Torralba 2001) is based on local histograms of gradient orientations. It captures the rough spatial arrangement of image gradients, and has been shown to work well for describing the overall appearance of a scene. Here instead, we extract GIST from each candidate *window*. In our experiments we use GIST with the default parameters.

Color Histograms (CH) provide complementary information to gradients. We describe a window with a single $10x20x20$ histogram in the LAB color space.

Bag of Visual Words (BOW) are de-facto standard for many object recognition tasks (Chum and Zisserman 2007; Dorkó and Schmid 2005; Lampert et al. 2009b; Zhang et al. 2007). We use SURF descriptors (Bay et al. 2008; Lampert et al. 2009b) and quantize them into 2000 words using k-means. A window is described by a BOW of SURF descriptors extracted at three different scales on a 32×32 grid.

Histograms of Oriented Gradients (HOG) also are an established descriptor for object class recognition (Dalal and Triggs 2005; Felzenszwalb et al. 2010). We extract HOGs on a 32×32 grid.

6 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. Table 3 gives an overview of the datasets used for the experiments.

6.1 Datasets

CALTECH4 (Fergus et al. 2003) 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 negative images, for each class we use the images from the three other classes.

Table 3 Overview of the datasets. The left half of the table gives the total number of images in the training sets of the target classes used to evaluate localization in weakly supervised images, the number of target classes, and of class/viewpoint combinations (remember that each class/viewpoint combination is input to our method separately). The right half of the table gives the same information about the meta-training sets used to learn the generic knowledge (i.e. the initial parameters of the CRF, Sect. 4)

Dataset	training sets			meta-training sets		
	images	cls	sets	images	cls	sets
CALTECH4	400	4	4	1040	6	34
PASCAL06-6x2	779	6	12	1249	5	17
PASCAL06-all	2184	10	33	1249	5	17
PASCAL07-6x2	463	6	12	1255	6	24
PASCAL07-all	2047	14	45	1255	6	24

As meta-training data $\mathcal{M}$ we use 1040 train+val images from 6 PASCAL07 classes (bicycle, boat, bus, cow, sheep, train) with bounding-box annotations. $\mathcal{M}$ is used to learn the parameters for initializing our CRF (Sect. 4). This is done only once. The same parameters are then reused in all experiments on CALTECH4.

PASCAL06-*6x2* (Everingham et al. 2006) We evaluate our method on a subset of the PASCAL06 dataset containing all images[1] from 6 classes (bicycle, car, cow, horse, motorbike, sheep) of the PASCAL06 train+val dataset from the left and right viewpoint. For each class we use all images containing at least one object not marked as difficult or truncated in the ground-truth. This holds also for all other PASCAL datasets below.

Each of the 12 class/viewpoint combinations contains between 31 and 132 images. As negative set $\mathcal{N}$ we use 2000 random images taken from train+val not containing any instance of the target class.

As meta-training data $\mathcal{M}$ we use 1249 train+val images from 5 PASCAL07 classes (bird, boat, bottle, chair, train) with between 1 and 4 viewpoints each.

PASCAL06-*all* (Everingham et al. 2006) For completeness, we evaluate our method on the entire PASCAL06 train+val dataset consisting of 10 classes (bicycle, bus, car, cat, cow, dog, horse, motorbike, person, sheep) with all viewpoints that have more than 20 images (leading to a total of 2184 images). The negative set $\mathcal{N}$ is chosen analogously to the negative set for the PASCAL06-6x2 datasets. Further, we re-use the meta-trained parameters from the experiments on PASCAL06-6x2. Note that there is

[1] This differs from the setting in the previous version of this work (Deselaers et al. 2010), where we used a smaller subset of images selected by Chum and Zisserman (2007), which are considerably easier as most of them contain a large dominant object.

Table 4 Results. The first block reports results for the baselines and the second for the competitors (Chum and Zisserman 2007; Russell et al. 2006). Rows (a)–(d): results for our method using only the localization stage. Rows (e)–(g): results for our full method using the localization and learning stages. All results are given in CorLoc. Column (Color) shows the colors used for visualization in Figs. 8, 9. Class-wise results for setup (g) are given in Table 5

Method	CALTECH4	PASCAL06		PASCAL07		Color
		6 × 2	all	6 × 2	all	
image center	66	44	36	25	16	
ESS	43	24	21	27	14	
Russell et al. (2006) (30 topics)	41	28	27	22	14	
Chum and Zisserman (2007)	55	45	34	33	19	
this paper—localization only						
(a) random windows	0	0	0	0	0	
(b) objectness windows with uniform score	73	50	35	30	17	
(c) objectness windows and score	75	55	41	37	23	
(d) all pairwise cues	63	58	45	37	23	
this paper—localization and learning						
(e) single cue (GIST), full adaptation	83	64	46	40	24	
(f) all cues, learning only Υ_f, Π	78	62	48	45	26	
(g) all cues, full adaption	81	64	49	50	28	

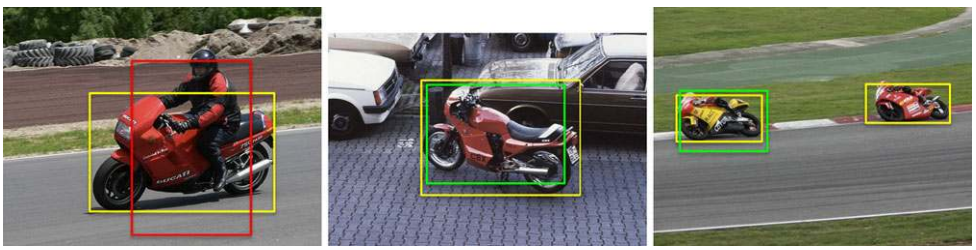


Fig. 7 Evaluation measure CorLoc. The *red* window overlaps $<$ 0.5 with the *yellow* ground-truth window. The green windows overlap $\geq$ 0.5 with the corresponding ground-truth windows. Therefore, over these three images, CorLoc is 66 %. Note how selecting any of the two motorbikes in the *right* image leads to the same CorLoc

no overlap between the meta-training classes and the training classes.

PASCAL07-*6x2* (Everingham et al. 2007) 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 viewpoint each. Each of the 12 class/viewpoint combinations contains between 21 and 50 images for a total of 463 images. As negative set $\mathcal{N}$ we use 2000 random images taken from PASCAL07 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 variation within a viewpoint (Figs. 8, 9). We report in detail on these classes because they represent compact objects on which fully supervised methods perform reasonably well (Everingham et al. 2007) (as opposed to classes such as 'potted plant' where even fully supervised methods fail). As meta-training data $\mathcal{M}$ we use 1255 train+val images from 6 other PASCAL07 classes (bird, car, cat, cow, dog, sheep).

PASCAL07-*all* (Everingham et al. 2007) Further, we also report results for *all* class/viewpoint combinations in PASCAL07 with more than 20 images (our method, as well as the competitors and baselines to which we compare, fails when given fewer images) leading to a total of 2047 images. We use the same meta-training data as for PASCAL07-6x2. In total, the PASCAL07-all set contains 45 class/viewpoint combinations, covering all 14 classes not used for meta-training.

Further, we re-use the meta-trained parameters from the experiments on PASCAL07-6x2. Note that there is no overlap between the meta-training classes and the training classes.

6.2 Evaluation

We *directly* evaluate the ability of our method to localize objects in a set of training images $\mathcal{I}$ only known to contain a

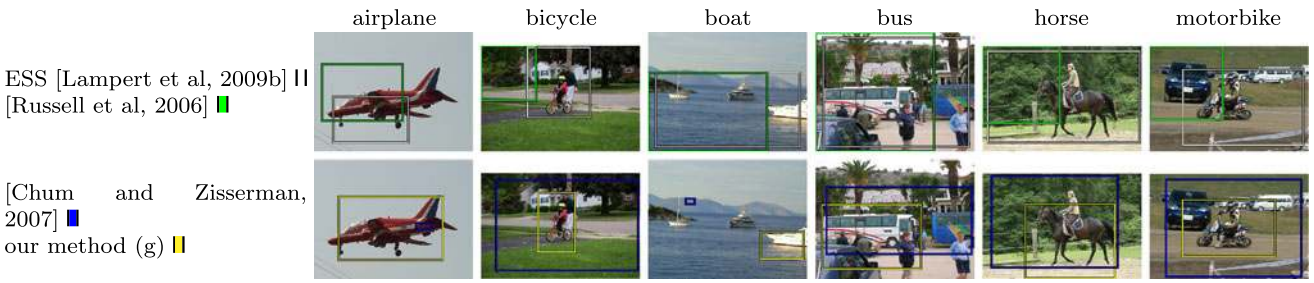


Fig. 8 Qualitative comparison to baselines and competitors. Example objects localized by different methods in their weakly supervised training images (i.e. only object presence is given for training, no locations). *Top row*: the ESS baseline (Lampert et al. 2009b) and the method of Russell et al. (2006). *Bottom row*: the method of Chum and Zisserman (2007) and our method in setup (g). Our method localizes object visibly better than both baselines and competitors, especially in cluttered images with small objects

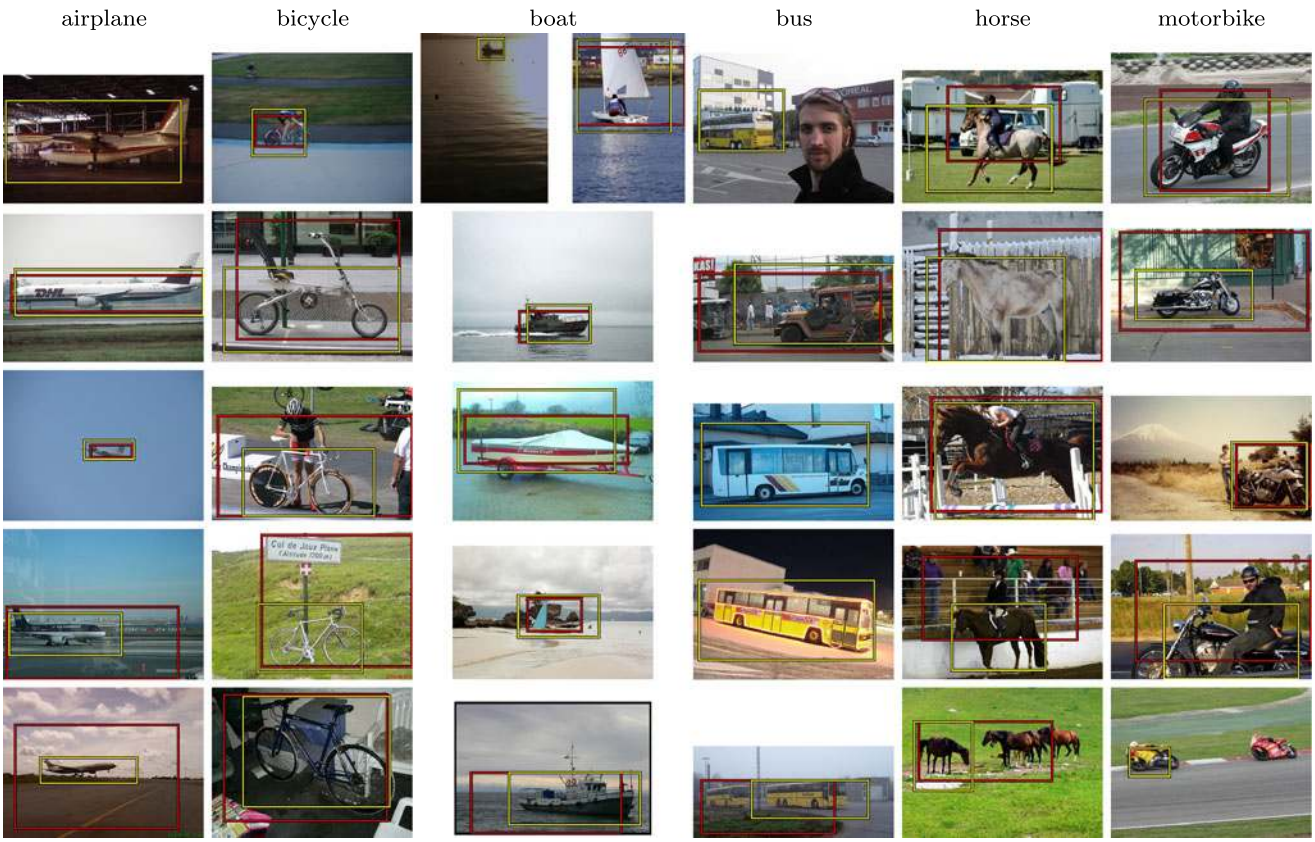


Fig. 9 Example results comparing our method in setup (d) to setup (g). If only is visible, both setups return the same window. The learning stage in setup (g) leads to more correctly localized objects

target class (Sect. 6). This direct evaluation reveals how well a method solves the auto-localization problem intrinsic to WSL, and it measures the quality of the input to training off-the-shelf fully supervised object detectors from the output of WSL (Sect. 7). Moreover, there are applications where the localization performance on an input set of weakly supervised images directly matters (e.g. co-segmentation or when annotating images downloaded from image search engines on the web). Finally, we note how our direct evaluation is analog to the standard evaluation protocol in the related fields of co-segmentation, unsupervised segmentation and object discovery, where no later test stage on new images is performed (see Table 1, rows with a "no" in column "evaluate on test data").

Table 4 shows results for two baselines, two competing methods (Chum and Zisserman 2007; Russell et al. 2006) 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, Fig. 7). No location of any object in $\mathcal{I}$ is given to any method beforehand. The detailed analysis in the next four paragraphs focuses on the CALTECH4, PASCAL06-6x2, and PASCAL07-6x2 datasets. Then we discuss results on the PASCAL06-all and PASCAL07-all dataset, and finally the last paragraph evaluates the quality of the candidate windows proposed by the objectness measure.

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 (Bay et al. 2008) from all images of a dataset, cluster them into 2000 words using k-means, and weight each word by the log of the relative frequency of occurrence in positive vs. negative images of a class (as done by Chum and Zisserman 2007; Dorkó and Schmid 2005; Lampert et al. 2009b). Hence, these feature weights are class-specific. For localization, we use Efficient Subwindow Search (ESS) (Lampert et al. 2009b) to find the window with the highest sum of weights in an image.[2]

The image center baseline confirms our impressions about the difficulty of the datasets. It reaches about 66 % CorLoc on CALTECH4, 44 % on PASCAL06-6x2, but fails on PASCAL07-6x2. The trend is confirmed by ESS.

Competitors We compare to the method of Russell et al. (2006) using their implementation.[3] This method does not directly return one window per image. It determines a number of 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. We evaluated different numbers of topics and found 30 to perform best on the average. This method achieves a rather low CorLoc on the challenging PASCAL07-6x2, but does better on the easier PASCAL06-6x2 and CALTECH4 datasets (41 % CorLoc).

As a second competitor we reimplemented the method of Chum and Zisserman (2007), which directly returns one window per image. It works well on CALTECH4 and on PASCAL06-6x2, where it finds about half the objects. On the much harder PASCAL07-6x2 it performs considerably worse since its initialization stage often does not lock onto objects.[4] Overall, this method performs better than Russell et al. (2006) on all datasets.

[2]Baseline suggested by C. Lampert in personal communication.

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

[4]Unfortunately, we could not obtain the source code from Chum and Zisserman (2007). We asked them to process our PASCAL07-6x2 training sets and they confirmed that their method performs poorly on them.

Localization Only (a)–(d) Here we evaluate our method after the localization stage (Sect. 3.1), without running the learning stage (Sect. 3.2). In order to investigate the impact of generic knowledge, we perform experiments with several stripped-down versions of our CRF model. Setup (a)–(c) use only GIST descriptors in the pairwise dissimilarity score Γ_f. Setup (a) uses 100 random candidate windows with uniform scores in Ω. Setup (b) uses 100 candidate windows sampled from the objectness measure, but with uniform scores in Ω. Setup (c) uses 100 candidate windows sampled from the objectness measure, with their objectness score in Ω (Sect. 4.1). While setup (a) is not able to localize any object, (b) already performs quite well, and adding the objectness score (c) gives an additional improvement. This shows that objectness is a powerful source of generic knowledge, which greatly helps localizing objects in weakly supervised images.

By adding the remaining appearance cues Γ_f in setup (d), the results improve further (Sect. 5). At this point, using only the localization stage, our method already outperforms all baselines and competitors. It localizes about two thirds of the objects in CALTECH4, more than half in PASCAL06-6x2, and 37 % in PASCAL07-6x2.

Localization and Learning (e)–(g) Here we run our full method, iteratively alternating localization and learning. In setup (e), we build on setup (c) using only GIST descriptors and adapt all parameters of our model to the target class (Sect. 3.2). This setup obtains a significant improvement over (c) on all datasets and even outperforms the localization-only multiple-cue setup (d) on the easier datasets.

In setups (f) and (g), we build on setup (d) using all appearances cues both for the pairwise dissimilarity Γ_f and for the class-specific appearance models Υ_f. In setup (f) we learn only appearance models Υ_f and shape models Π_f specific to the target class (Sects. 3.2.1, 3.2.2). This already leads to a clear improvement on all datasets demonstrating that our procedure properly acquires new knowledge specific to the target class. In setup (g) all parameters of the CRF are adapted to the target class (Sect. 3.2) which brings an additional improvement. Interestingly, on the PASCAL07 datasets the learning stage helps localization by a larger amount when using all appearance cues. This is because multiple descriptors are particularly beneficial in harder imaging conditions, and because our learning stage automatically re-weights the appearance cues, specializing their combination to each target class (Sect. 4.4).

The full method (g) substantially outperforms all competitors/baselines on all datasets. It reaches about 150 % the CorLoc of the second best method of Chum and Zisserman (2007) on PASCAL07-6x2. Overall, it finds most objects in CALTECH4, about two thirds in PASCAL06-6x2, and half in PASCAL07-6x2 (Figs. 8, 9).

Table 5 Class-wise CorLoc for setup (g) in Table 4

PASCAL06-6x2			PASCAL07-6x2		
class	left	right	class	left	right
bicycle	85	68	aeroplane	58	59
car	77	67	bicycle	46	40
cow	73	70	boat	9	16
horse	44	46	bus	38	74
motorbike	42	67	horse	58	52
sheep	67	57	motorbike	67	76

As Table 4 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 (e), (g)).

Table 5 shows the CorLoc for setup (g) per class/viewpoint combination for both PASCAL06-6x2 and PASCAL07-6x2. The occasional performance differences between the left and right viewpoints of the same class are due to the different number of available images and the average size of the objects. For example, in PASCAL06-6x2 motorbike-left has 31 images with motorbikes of 179×205 pixels on average, whereas motorbike-right has 52 images with 435×370 pixels on average.

To further demonstrate the genericness of our GK, we perform an additional experiment on PASCAL06-6x2 analog to setup (g), but this time using the GK learned from the meta-training set originally used for PASCAL07-6x2 (see Sect. 6.1). Remarkably, the CorLoc on PASCAL06-6x2 varies by less than 1 % when changing between the two meta-training sets, which demonstrates the GK we propose is truly generic across classes. As additional evidence in this direction, we refer to the experiment in page 10 of Alexe et al. (2012), which shows that the performance of objectness does not change even when trained from very different image sets.

PASCAL-All Datasets For completeness, Table 4 also reports results on the PASCAL06-all and PASCAL07-all datasets, which contain 33 and 45 class/viewpoint combinations respectively, including many for which even fully supervised methods fail (e.g. 'potted plant'). Comparing PASCAL06-6x2 and PASCAL06-all, CorLoc drops by about a third. On PASCAL07-6x2 and PASCAL07-all, CorLoc drops by about half for all methods, suggesting that WS learning on *all* PASCAL07 classes is beyond what is currently possible. However, it is interesting to notice how the relative performance of our method (setup (g)) compared to the competitors (Russell et al. 2006; Chum and Zisserman 2007) remains close to what is observed on PASCAL07-6x2.

Table 6 Evaluation of the objectness measure. The precision and hit-rate of the windows sampled from the objectness measure for the target classes

Dataset	precision [%]	hit-rate [%]
Caltech 4	32	100
Pascal 06 6x2	26	89
Pascal 06 all	21	80
Pascal 07 6x2	19	85
Pascal 07 all	13	71

Objectness We also evaluate the 100 windows per image sampled from Ω (Table 6). The *hit-rate* is the percentage of objects of the target class covered by one of sampled window (up to intersection-over-union ≥ 0.5). It gives an upper-bound on the CorLoc that can be achieved by our method. As the table shows, most target objects are covered. The *precision* is the percentage of sampled windows covering an object of the target class. It gives the ratio between correct and incorrect windows that enter the CRF model. This ratio is much higher than when considering all image windows.

The hit-rates and precisions over the different datasets also confirm their perceived difficulty. On CALTECH4 all objects are covered and about 1 in 3 windows is on an object. On PASCAL07-6x2 only about 1 in 5 windows covers an object. However, the hit-rate is still high showing that objectness is a suitable focus of attention measure for weakly supervised learning, even in highly challenging imaging conditions.

6.3 Comparison to Kim and Torralba (2009)

We evaluate our method on PASCAL06 also in the experimental setup of Kim and Torralba (2009, Fig. 5): for every class we run our method (g) on all images showing an object of this class and then evaluate object detection accuracy on the test images. Performance is measured by mean Average Precision over all 10 classes (for details of this setup we refer to Kim and Torralba (2009)). Note how no PASCAL06 class appears in our meta-training set (which are 5 other classes from PASCAL07). In this setup our method brings a mAP of 0.24, which compares favorably to the 0.21 of Kim and Torralba (2009).[5]

7 Experiments: Object Detection in New Test Images

Our method enables training a fully-supervised object detector from weakly supervised data, although this would normally require object location annotations. To demonstrate

[5] Derived from the PR plots in their paper (Fig. 5).

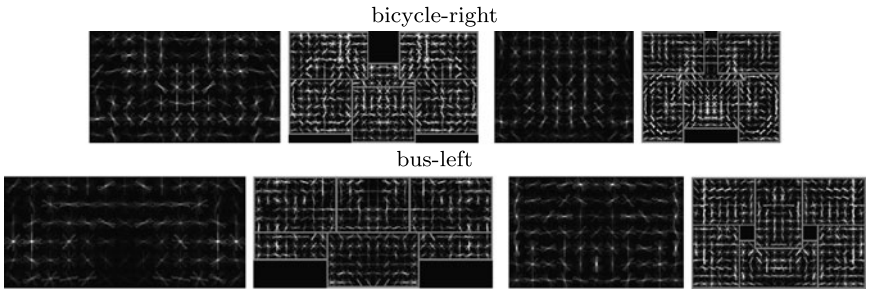


Fig. 10 Models of Felzenszwalb et al. (2010) trained on PASCAL07-6x2 from the output of our method (*left*) and from ground-truth bounding-boxes (*right*). Note how similar the models are

Table 7 Detection results on test images. mAP values for training the object detector by Felzenszwalb et al. (2010) on the output of setup (g) (WSL) and on the ground-truth bounding boxes (GT). The third column reports the ratio of the two values, which shows how well the weakly supervised setup performs relative to the fully supervised one

Dataset	WSL	GT	$\frac{\text{WSL}}{\text{GT}}$ [%]
Caltech 4	0.32	0.36	87 %
PASCAL06-6x2	0.28	0.36	78 %
PASCAL07-6x2	0.21	0.33	65 %

Table 8 Class-wise AP for the experiments in Table 7 in percent AP. The difference between the fully and the weakly supervised system is given in parentheses

PASCAL06-6x2				
class	left		right	
bicycle	51	(−6)	63	(0)
car	29	(−1)	29	(0)
cow	18	(2)	13	(−3)
horse	10	(−32)	0	(−42)
motorbike	31	(−24)	39	(−1)
sheep	22	(7)	29	(8)

PASCAL07-6x2				
class	left		right	
aeroplane	5	(−18)	18	(−14)
bicycle	49	(−10)	62	(−2)
boat	0	(−0)	0	(−1)
bus	0	(−21)	16	(4)
horse	29	(−16)	14	(−25)
motorbike	48	(−7)	16	(−26)

this point, we train the fully supervised object detector of Felzenszwalb et al. (2010)[6] from objects localized using our setup (g), and compare its performance to the original model trained from ground-truth bounding-boxes. In all experiments we use one component and six parts per class. As negative training images for a class we use the training images of the other classes.

We perform this experiment for CALTECH4, PASCAL06-6x2, and PASCAL07-6x2. The detection performance for each class/viewpoint is measured by the average precision (AP). For the PASCAL06-6x2 and PASCAL07-6x2 tasks the performance is measured on their full tests sets (2686 and 4952 images respectively). These test sets are entirely disjoint from their respective train+val sets used for training and meta-training. For CALTECH4, we form a test set by choosing 100 random images from each class (excluding images used for training). We then evaluate each model on the whole 400-image test set. As usual in a test stage, no information is given about the test images, also not whether they contain an object of the class being evaluated.

Table 7 reports the mean AP values (mAP) over all class/viewpoint combinations in each dataset. On the easy CALTECH4, the performance of the weakly-supervised method is close to that of the fully supervised model. Even on the more challenging PASCAL06-6x2 the WSL model still obtains almost 80 % of the mAP of the fully supervised model, while on the very hard PASCAL07-6x2 it yields about two thirds of its performance.

[6]The source code is available at http://people.cs.uchicago.edu/~pff/latent/.

These results demonstrate that it is possible to train a functional fully supervised object detector from weakly supervised images from the output of our method. 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. 10 visually compares two models trained from the output of our method to the corresponding models trained from ground-truth bounding-boxes.

Table 8 reports AP for each class/viewpoint combination separately. Interestingly, larger differences between the fully and weakly supervised setups occur when the weakly supervised method performs worse in localizing objects in their training images. For example, on PASCAL06 horses-left, horses-right, and motorbike-left, which are the three class-viewpoint combinations with the lowest CorLoc on the training data (Table 5). This correlation emphasizes the value of directly evaluating localization accuracy on the weakly supervised training images (Sect. 6.2).

To further demonstrate that performance at test time strongly depends on the quality of object localization at training time (CorLoc), we repeated this experiment when using the approach of Chum and Zisserman (2007) instead of ours to select windows in the WS training images. On PASCAL06-6x2 this achieves an AP of 0.12 and on PASCAL07-6x2 0.11, compared to our 0.28 and 0.21 respectively.

8 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 it 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.

Throughout the paper we used the wording 'generic knowledge' to convey the meaning of applying to most object classes, as opposed to being specific to one class (Everingham et al. 2010; Felzenszwalb et al. 2010; Fergus et al. 2003). However, GK is not an accurate nor complete representation of any particular class. For example, it could not be used on its own to reliably detect objects of a particular class. Instead, GK provides a broad basis about objects in general, which we have demonstrated in this paper to help learning new object classes.

In future work we plan to extend our method in various directions. First, we plan to learn separate models for different viewpoints of an object class from a single mixed training set. This could be achieved by extending the state-space of each node of the CRF to the cartesian product of the set of candidate windows and the set of viewpoints. Second, computational efficiency could be improved by decimating the fully connected CRF to a N-order Markov chain, or by removing edges between images of very different appearance. Third, we plan to exploit hierarchical dependencies between classes from large-scale datasets such as ImageNet. In this fashion a new class will not only benefit from generic knowledge, but also from more specific knowledge from semantically related classes. Ultimately, we hope to formulate a unified transfer learning framework where multiple sources of knowledge at many levels of generality are automatically selected and combined to help learning a new class in the most effective manner.

Acknowledgements The authors gratefully acknowledge support from the Swiss National Science Foundation.

References

Alexe, B., Deselaers, T., & Ferrari, V. (2010a). ClassCut for unsupervised class segmentation. In *ECCV*.

Alexe, B., Deselaers, T., & Ferrari, V. (2010b). What is an object? In *CVPR*.

Alexe, B., Deselaers, T., & Ferrari, V. (2012). Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2002). Support vector machines for multiple-instance learning. In *NIPS*.

Arora, H., Loeff, N., Forsyth, D., & Ahuja, N. (2007). Unsupervised segmentation of objects using efficient learning. In *CVPR*.

Babenko, B., Branson, S., & Belongie, S. (2009). Similarity metrics for categorization: From monolithic to category specific. In *ICCV*.

Bagon, S., Brostovski, O., Galun, M., & Irani, M. (2010). Detecting and sketching the common. In *CVPR*.

Bay, H., Ess, A., Tuytelaars, T., & van Gool, L. (2008). SURF: speeded up robust features. In *CVIU*.

Blaschko, B., Vedaldi, A., & Zisserman, A. (2010). Simultaneous object detection and ranking with weak supervision. In *NIPS*.

Borenstein, E., & Ullman, S. (2004). Learning to segment. In *ECCV*.

Cao, L., & Li, F. F. (2007). Spatially coherent latent topic model for concurrent segmentation and classification of objects and scene. In *ICCV*.

Carreira, J., Li, F., & Sminchisescu, C. (2010). Constrained parametric min cuts for automatic object segmentation. In *CVPR*.

Chen, Y., Bi, J., & Wang, J. Z. (2006). MILES: multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 1931–1947.

Chum, O., & Zisserman, A. (2007). An exemplar model for learning object classes. In *CVPR*.

Crandall, D. J., & Huttenlocher, D. (2006). Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV*.

Dalal, N., & Triggs, B. (2005). Histogram of Oriented Gradients for human detection. In *CVPR*.

Deselaers, T., & Ferrari, V. (2010). A conditional random field for multiple-instance learning. In *ICML*.

Deselaers, T., Alexe, B., & Ferrari, V. (2010). Localizing objects while learning their appearance. In *ECCV*.

Dorkó, G., & Schmid, C. (2005). Object class recognition using discriminative local features. Tech. Rep. RR-5497, INRIA, Rhone-Alpes.

Endres, I., & Hoiem, D. (2010). Category independent object proposals. In *ECCV*.

Everingham, M., Van Gool, L., Williams, C. K. I., & Zisserman, A. (2006). The PASCAL Visual Object Classes Challenge 2006 (VOC2006). http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2006/.

Everingham, M., Van Gool, L., Williams, C., Winn, J., & Zisserman, A. (2007). The PASCAL Visual Object Classes Challenge 2007 Results.

Everingham, M., et al. (2010). The PASCAL Visual Object Classes Challenge 2010 Results.

Fei-Fei, L., Fergus, R., & Perona, P. (2003). A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV* (pp. 1134–1141).

Fei-Fei, L., Fergus, R., & Perona, P. (2004). 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*.

Felzenszwalb, P., Girshick, R., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1627–1645.

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *CVPR*.

Finley, T., & Joachims, T. (2008). Training structural svms when exact inference is intractable. In *ICML*.

Fritz, M., & Schiele, B. (2006). Towards unsupervised discovery of visual categories. In *DAGM*.

Frome, A., Singer, Y., Sha, F., & Malik, J. (2007). Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*.

Gaidon, A., Marszalek, M., & Schmid, C. (2009). Mining visual actions from movies. In *BMVC*.

Galleguillos, C., Babenko, B., Rabinovich, A., & Belongie, S. (2008). Weakly supervised object localization with stable segmentations. In *ECCV*.

Grauman, K., & Darrell, T. (2006). Unsupervised learning of categories from sets of partially matching image features. In *CVPR*.

Kim, G., & Torralba, A. (2009). Unsupervised detection of regions of interest using iterative link analysis. In *NIPS*.

Kolmogorov, V. (2006a). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(10), 1568–1583.

Kolmogorov, V. (2006b). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(10), 1568–1583.

Lampert, C., Nickisch, H., & Harmeling, S. (2009a). Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*.

Lampert, C. H., Blaschko, M. B., & Hofmann, T. (2009b). Efficient subwindow search: A branch and bound framework for object localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(12), 2129–2142.

Lando, M., & Edelman, S. (1995). *Generalization from a single view in face recognition*. (technical report cs-tr 95-02). The Weizmann Institute of Science.

Lee, Y., & Grauman, K. (2009a). Shape discovery from unlabeled image collections. In *CVPR*.

Lee, Y. J., & Grauman, K. (2009b). Foreground focus: unsupervised learning from partially matching images. *International Journal of Computer Vision*, *85*, 143–166.

Malisiewicz, T., & Efros, A. A. (2008). Recognition by association via learning per-exemplar distances. In *CVPR*.

Nguyen, M., Torresani, L., de la Torre, F., & Rother, C. (2009). Weakly supervised discriminative localization and classification: a joint learning process. In *ICCV*.

Nowak, E., & Jurie, F. (2007). Learning visual similarity measures for comparing never seen objects. In *CVPR*.

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, *42*(3), 145–175.

Payet, N., & Todorovic, S. (2010). From a set of shapes to object discovery. In *ECCV*.

Quattoni, A., Collins, M., & Darrell, T. (2008). Transfer learning for image classification with sparse prototype representations. In *CVPR*.

Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. (2007). Self-taught learning: transfer learning from unlabeled data. In *ICML*.

Ramanan, D. (2006). Learning to parse images of articulated bodies. In *NIPS*.

Rohrbach, M., Stark, M., Szarvas, G., Gurevych, I., & Schiele, B. (2010). What helps where—and why? semantic relatedness for knowledge transfer. In *CVPR*.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: interactive foreground extraction using iterated graph cuts. *Computer Graphics*, *23*(3), 309–314.

Russel, B. C., & Torralba, A. (2008). LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1–3), 157–173.

Russell, B. C., Efros, A. A., Sivic, J., Freeman, W. T., & Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*.

Stark, M., Goesele, M., & Schiele, B. (2009). A shape-based object class model for knowledge transfer. In *ICCV*.

Szummer, M., Kohli, P., & Hoiem, D. (2008). Learning CRFs using graph cuts. In *ECCV*.

Thrun, S. (1996). Is learning the n-th thing any easier than learning the first? In *NIPS*.

Todorovic, S., & Ahuja, N. (2006). Extracting subimages of an unknown category from a set of images. In *CVPR*.

Tommasi, T., & Caputo, B. (2009). The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories. In *BMVC*.

Tommasi, T., Orabona, F., & Caputo, B. (2010). Safety in numbers: learning categories from few examples with multi model knowledge transfer. In *CVPR*.

Torresani, L., Szummer, M., & Fitzgibbon, A. (2010). Efficient object category recognition using classemes. In *ECCV*.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.

Viola, P. A., Platt, J., & Zhang, C. (2005). Multiple instance boosting for object detection. In *NIPS*.

Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *NIPS*.

Winn, J., & Jojic, N. (2005a). LOCUS: learning object classes with unsupervised segmentation. In *ICCV*.

Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, *73*(2), 213–238