Mach Learn (2014) 94:261–279
DOI 10.1007/s10994-013-5330-2

Learning what is where from unlabeled images: joint localization and clustering of foreground objects

Ashok Chandrashekar · Lorenzo Torresani · Richard Granger

Received: 1 May 2012 / Accepted: 21 January 2013 / Published online: 23 February 2013

Abstract "What does it mean, to see? The plain man's answer would be, to know what is where by looking." This famous quote by David Marr (Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, Freeman, New York, 1982) sums up the holy grail of vision: discovering what is present in the world, and where it is, from unlabeled images. In this paper we tackle this challenging problem by proposing a generative model of object formation and describe an efficient algorithm to automatically learn the parameters of the model from a collection of unlabeled images. Our algorithm discovers the objects and their spatial extents by clustering together images containing similar foregrounds. Our approach simultaneously solves for the image clusters, the foreground appearance models and the spatial regions containing the objects by optimizing a single likelihood function defined over the entire image collection. We describe two methods for efficient foreground localization: the first method does not require any bottom-up image segmentation and discovers the foreground region as a contiguous rectangular bounding box. The second method expresses the foreground as a collection of super-pixels generated through a bottom-up segmentation of the image. However, unlike previous methods, objects are not assumed to be encapsulated by a single segment. Evaluation on standard benchmarks and comparison with prior methods demonstrate that our approach achieves state-of-the-art results on the problem of unsupervised foreground localization and clustering.

Keywords Unsupervised · Clustering · Foreground · Localization

Editors: Antoine Bordes, Léon Bottou, Ronan Collobert, Dan Roth, Jason Weston, and Luke Zettlemoyer.

A. Chandrashekar (✉) · L. Torresani · R. Granger
Dartmouth College, Hanover, NH, USA
e-mail: ashok@cs.dartmouth.edu

L. Torresani
e-mail: lorenzo@cs.dartmouth.edu

R. Granger
e-mail: richard.granger@dartmouth.edu

1 Introduction

Object categorization requires recognizing the classes of objects appearing in an input photo. Rather than performing classification of the entire image as a whole, object-class recognition systems often operate by decomposing the photo into different regions corresponding to the objects present in the scene. Treating object localization and recognition jointly allows such methods to be more robust to clutter, variations in backgrounds, as well as presence of multiple objects.

We can distinguish several methodologies for object recognition and localization on the basis of the amount of human supervision needed during training. When the training images are manually segmented into semantic regions, object localization can be formulated as the task of densely matching regions of the input photo to the manually annotated segments of similar images in the database (Liu et al. 2010). In order to achieve good results, these methods require very large collections of annotated images so as to maximize the chance of a close image match in the database. However, due to the cost of collecting pixel-labels, such datasets are extremely time-consuming to generate and difficult to label accurately.

A second methodology involves the use of datasets where only the object of interest is manually segmented in the training images. Typically, recognition and localization are then achieved using a combination of bottom-up segmentation and top down classification (Borenstein et al. 2004; Leibe and Schiele 2003; Tu et al. 2005; Yu and Shi 2003). But these methods are computationally expensive to run and, again, the requirement for detailed segmentation in the training set is far too onerous.

An efficient alternative is object detection (Chum and Zisserman 2007; Dalal and Triggs 2005), which involves sliding a subwindow classifier exhaustively over all rectangular regions of the test image in order to robustly localize the box that is most likely to contain the object. A branch and bound strategy has been recently proposed (Lampert et al. 2008) to make this brute-force evaluation more efficient, by rapidly removing from consideration a large portion of regions. These algorithms normally require the object to be delineated using a bounding box in the training dataset, which is easier to generate compared to full segmentation. However, even this form of labeling is expensive to acquire and effectively restricts the size of the training set. Furthermore, the sizes and locations of the bounding boxes are typically chosen arbitrarily by the labeler and are consequently unlikely to be optimal for recognition.

When images have labels indicating the objects present in them but no locality information for the objects, semi-supervised methods can be applied to learn automatically the correspondences between image regions and the labels of the image. Most methods in this genre use bottom-up segmentation as a preprocessing to produce candidate segments, and then perform top down learning on the segments (Duygulu et al. 2002; Andrews et al. 2003; Chen and Wang 2004). However the main weakness in such methods is relying on the ill-defined task of bottom-up segmentation (based on low-level visual cues such as edges and texture) to segment images such that objects or semantically-coherent regions are represented by a single segment. Thus, such approaches typically yield poor classification accuracy. Recently, Nguyen et al. (2009) and Deselaers et al. (2010) have proposed weakly-supervised object localization methods avoiding the need of bottom-up segmentation: the idea of these methods is to simultaneously localize discriminative subwindows in the training images and to learn a classifier to recognize such regions. However, even such methods require supervision in terms of class labels.

In this paper we contrast the traditional methodologies for object localization and recognition outlined above, by presenting a fully-unsupervised method which completely eliminates the need for time-consuming and suboptimal human labeling. The intuition behind our approach is that objects can be viewed as recurring foreground patterns appearing as coherent image regions. Thus, we can formulate object discovery as the task of partitioning an unlabeled collection of images into K subsets (clusters), such that all images within each subset share a similar foreground. In order to obtain a method scalable to large collections and many classes, we adopt a foreground mask-based representation of objects, which enables fast localization given the object model. Specifically, we represent the object in an image as a histogram of quantized local features occurring in the enclosing foreground mask. We view each object instance as a random variable drawn from an unknown distribution common to all instances of that object class. This common distribution assumption constrains all foreground histograms of an object class to represent subtle variations around a prototypical average histogram. Based on this assumption, our approach poses object discovery as a maximum likelihood estimation problem, to be optimized over the entire collection of unlabeled images. We present a method that maximizes this objective by simultaneously solving for the histogram model parameters of the object classes, detecting the object instances of each class in the unlabeled images, and performing a soft semantic clustering of images in the dataset. In the next section we review prior methods for unsupervised object discovery and discuss their relation to our approach.

2 Related work

Class-agnostic methods for object discovery (Alexe et al. 2010b; Itti and Koch 2001), attempt to discover image regions that are likely to contain objects, irrespective of their categories. These methods operate on individual images by applying a single, general object model capturing visual properties common to most classes. However, the notion of "objectness" is poorly-defined; furthermore, these techniques do not attempt to learn distinct models of the different objects and thus cannot be used for recognition.

Our approach is more closely related to methods that discover objects from collections of unlabeled images by identifying statistically reoccurring image fragments. Lee and Grauman (2009) have proposed an approach to automatically localize foreground features from unlabeled images: by learning the 'significance' weights of semi-local features iteratively through image grouping, their method determines for each image which features are most relevant, given the image content in the remainder of the collection. While this work successfully demonstrates that a mutual reinforcement of object-level and feature-level similarity improves unsupervised image clustering, there is no clear way of translating feature weights into foreground localization and object extents. Furthermore, it performs clustering from pairwise image matches and therefore the computational cost *at each iteration* is quadratic in number of images. Finally, the algorithm alternates between image clustering and updating the foreground weights without a unifying formal objective and thus its convergence properties are unclear.

Various semantic topic models (Fergus et al. 2005; Sudderth et al. 2005; Fritz and Schiele 2008; Kim and Torralba 2009; Deselaers et al. 2010) have been proposed for similar tasks where the location of the object is treated as a latent variable to be estimated. However, most of these methods are not fully unsupervised and often resort to an expensive sliding window mechanism for object detection.

Our work is inspired by the approach of Russell et al. (2006), who propose a fully-unsupervised algorithm for object discovery. Multiple segmentations are performed for each image by varying the parameters of a segmentation method. The key assumption is that each object instance is correctly segmented (as a single contiguous segment) at least once through multiple segmentation and therefore the correct segments corresponding to object classes occur more often than random background. This suggests that the features of correct segments form object-specific coherent clusters discoverable using latent topic models from text analysis. Although the algorithm is shown to be able to discover many different objects, it still suffers from its reliance on bottom-up segmentation to find a single segment encapsulating the object. In practice this assumption is often violated as bottom-up segmentation is an inherently ill-posed task: it is necessary to know the category of the object in order to reliably segment it from the scene. Also, we would like to point out that their method addresses a different objective from ours, as their technique does not provide a way to cluster the images or determine which regions in the images correspond to image foregrounds. Nevertheless, in the experiments we propose adaptations of their method to our task in order to perform a quantitative comparison between the approaches.

Differently from prior work, we propose a generative model of foreground formation that enables simultaneous image clustering and foreground localization via maximum likelihood estimation. Unlike Russell et al. (2006), our approach treats each image as a composition of foreground and background where the foreground is explained by a single model shared with images of the same object class and the background is image-specific and hence not modeled. We treat the foreground mask as a parameter to be estimated as part of the likelihood optimization. We demonstrate that this leads to better localization and image clustering. Apart from the proposed unified framework of maximum likelihood estimation for foreground clustering and localization, the main contribution of this paper is to show that our choice of foreground model enables the use of two efficient methods for detection of object foregrounds in images. In the first method, the foreground is described by a rectangular bounding box enclosing the object, and it is localized without the need for bottom-up segmentation. The second method does rely on bottom-up segmentation. However, the segments generated are assumed to be nothing more than "super-pixels". In particular, we do not assume that the foreground is captured by a single segment. Hence, we overcome most of the drawbacks of previous methods which rely on the unrealistic assumption that bottom-up segmentation will produce a segment for each object in the image.

3 Generative model for unsupervised object discovery

We now describe our proposed generative model for unsupervised object discovery. We assume we are given as input a collection of N unlabeled images $z_1, \ldots, z_N$, with each image containing one of K objects. Our objective is twofold: to separate the images into K disjoint subsets (clusters) corresponding to the K object classes and to localize the object within each image. We denote with x_n the unknown foreground mask enclosing the foreground object of image z_n. We represent the foreground region x_n of image z_n by computing the un-normalized histogram $h(z_n, x_n) \in \mathbb{N}^W$ of the visual words (i.e., quantized local visual features) occurring inside x_n: here W represents the number of unique words in the visual codebook, which, as usual, is learned during an offline stage from training images. We assume that the foreground histograms of images belonging to the k-th object class are generated from a common model defined by parameters θ_k^F. Specifically, let $l_n \in \{1, \ldots, K\}$ denote the unknown cluster label of image z_n, which we assume to be drawn from a Multinomial distribution with parameters $\pi = \{\pi_1, \ldots, \pi_K\}$. Then, we model the foreground histogram $h(z_n, x_n)$ as a random variable drawn from a Gaussian distribution with parameters $\theta_{l_n}^F = \{\mu_{l_n}, \Sigma_{l_n}\}$, i.e., $h(z_n, x_n) \sim \mathcal{N}(\mu_{l_n}, \Sigma_{l_n})$. In Sects. 4.1, 4.2 we demonstrate that this simple Gaussian assumption is the key to enable efficient foreground localization, which

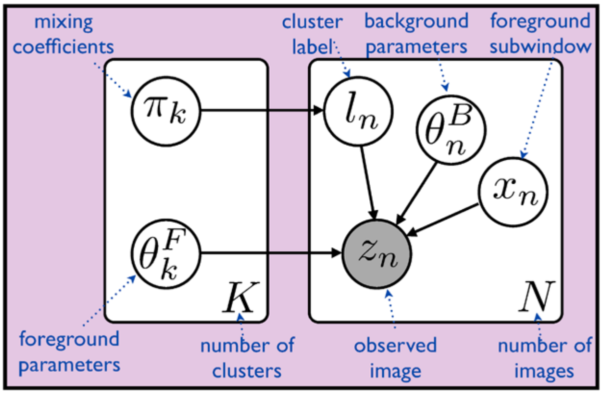

Fig. 1 Our generative model of image formation: image z_n is obtained by first drawing its object class (l_n); then the appearance of the object inside the foreground location (x_n) is generated from a distribution ($\theta_{l_n}^F$) common to all objects instances of that class. The background model (θ_n^B) is assumed to change with every image

is a fundamental requirement for scalability to large image collections. In order to reduce the number of parameters to be estimated, we assume the covariance Σ_k of each cluster k to be diagonal: $\Sigma_k = \text{diag}(\lambda_{k1}, \dots, \lambda_{kW})^{-1}$. Finally, each image is assumed to have its own independent background model defined by parameters θ_n^B. For our objective of object discovery, the background parameters can be left unresolved. The complete generative model is summarize graphically in Fig. 1. We propose to maximize the likelihood of this model by marginalizing over the cluster labels, which we treat as hidden variables. In other words, our objective is to find parameters $\theta = \{\theta^F, \pi\}$ and foreground regions $x = \{x_1, \dots, x_n\}$ maximizing

$$p(z|x,\theta)p(x) = \prod_{n=1}^{N} p(z_n|x_n,\theta)p(x_n) = \prod_{n=1}^{N}\sum_{k=1}^{K} p(z_n, l_n = k|x_n,\theta)p(x_n) \tag{1}$$

where $p(x_n)$ is a prior penalizing unlikely configurations of the foreground mask.

4 Optimization

We can maximize the proposed penalized likelihood via an Expectation Maximization (EM) algorithm alternating between estimating the distribution over the cluster labels l_n and solving for the foreground models and locations. Next, we show how to perform each of these steps and demonstrate that our modeling choices lead to efficient localization of the object regions given the foreground parameters θ. The penalized complete log-likelihood of our model is given by:

$$\begin{aligned}
\mathcal{L} &= \log \prod_{n=1}^{N} p(z_n, l_n|x_n,\theta)p(x_n) \\
&= \log \prod_{n=1}^{N} p(z_n|l_n, x_n,\theta)p(l_n|\theta)p(x_n) \\
&= \sum_{n=1}^{N} \log p(z_n|x_n, l_n,\theta) + \log p(l_n|\theta) + \log p(x_n)
\end{aligned} \tag{2}$$

The E-step of the algorithm involves calculating the latent posterior distribution $\gamma_{nk} \equiv p(l_n = k|z_n, x_n, \theta)$ given the current estimates for θ and x. It can be seen that this reduces to an evaluation of the following equation:

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(h(z_n, x_n); \mu_k, \Sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(h(z_n, x_n); \mu_{k'}, \Sigma_{k'})} \tag{3}$$

The M-step requires maximizing the expected log-likelihood $\langle \mathcal{L}(\theta) \rangle_\gamma$ with respect to θ and x. We begin by writing the expected log likelihood:

$$\begin{aligned} \langle \mathcal{L} \rangle_\gamma = {} & \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \log \mathcal{N}\bigl(h(z_n, x_n); \mu_k, \Sigma_k\bigr) \\ & + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \pi_k + \sum_{n=1}^{N} \log p(x_n) + \text{const.} \end{aligned} \tag{4}$$

The update steps for parameters θ can be obtained by setting the respective derivatives to zero. This leads to the following rules:

$$\pi_k \leftarrow \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk} \tag{5}$$

$$\mu_k \leftarrow \frac{1}{\sum_{n'=1}^{N} \gamma_{nk}} \sum_{n'=1}^{N} \gamma_{nk} h(z_n, x_n) \tag{6}$$

$$\lambda_{kw}^{-1} \leftarrow \frac{1}{\sum_{n'=1}^{N} \gamma_{nk}} \sum_{n'=1}^{N} \gamma_{nk} \bigl(\bigl[h(z_n, x_n)\bigr]_w - [\mu_k]_w\bigr)^2 \tag{7}$$

where $[a]_w$ denotes the w-th entry of a vector a.

In the M-step we also need to update the estimate of the foreground mask x_n by solving the following optimization:

$$\begin{aligned} \arg\max_{x_n} \langle \mathcal{L} \rangle_\gamma &= \arg\max_{x_n} \left\{ \log p(x_n) + \sum_{k=1}^{K} \gamma_{nk} \log \mathcal{N}\bigl(h(z_n, x_n); \mu_k, \Sigma_k\bigr) \right\} \\ &= \arg\max_{x_n} \left\{ \log p(x_n) - \sum_{k=1}^{K} \gamma_{nk} \sum_{w=1}^{W} \lambda_{kw} \bigl(\bigl[h(z_n, x_n)\bigr]_w - [\mu_k]_w\bigr)^2 \right\} \end{aligned} \tag{8}$$

We now show that this objective can be rewritten in a form that leads to efficient optimization. Let $\lambda_k = [\lambda_{k1}, \dots, \lambda_{kW}]^T \in \mathbb{R}^W$, $\boldsymbol{c} = [\gamma_{n1} \lambda_1^T, \dots, \gamma_{nK} \lambda_K^T]^T \in \mathbb{R}^{WK}$, $\hat{\mu} = [\mu_1^T, \dots, \mu_K^T]^T \in \mathbb{R}^{WK}$. Finally let us denote with $\hat{h}(z_n, x_n)$ the vector containing K copies of $h(z_n, x_n)$, i.e., $\hat{h}(z_n, x_n) = [h(z_n, x_n)^T, \dots, h(z_n, x_n)^T]^T \in \mathbb{R}^{WK}$. Then, we can rewrite the objective of (8) equivalently as follows:

$$\arg\max_{x_n} \langle \mathcal{L} \rangle_\gamma = \arg\max_{x_n \in \mathcal{X}} \left\{ \log p(x_n) - \sum_{j=1}^{WK} c_j \bigl(\bigl[\hat{h}(z_n, x_n)\bigr]_j - [\hat{\mu}]_j\bigr)^2 \right\} \tag{9}$$

We next introduce methods to optimize this objective efficiently.

4.1 Image foregrounds as rectangular bounding boxes

A popular way for circumscribing an object in an image is by using rectangular bounding boxes. Traditionally for the object detection task, the bounding boxes are determined using an expensive sliding window method (Chum and Zisserman 2007; Dalal and Triggs 2005). However, recently Lampert et al. (2008) have introduced a branch and bound optimization procedure to localize bounding boxes efficiently. In our first proposed approach to localize foregrounds, we treat the foreground of each image as a contiguous rectangular region which is represented by the variable $x_n \in \mathcal{X}$. Here $\mathcal{X}$ indicates the space of all rectangular subwindows inside the image. The foreground content $h(z_n, x_n)$ is then the histogram of all features that occur within the rectangle.

Consider (9): note that the second term in this objective is a weighted Euclidean distance between $\hat{\mu}$ and the histogram $\hat{h}(z_n, x_n)$ computed from the visual words in subwindow x_n. For such term, we can define a quality lower bound function over sets of subwindows as described by Lampert et al. (2008). Let x^{min} and x^{max} be the smallest and largest rectangles in a candidate set of rectangles $X \subset \mathcal{X}$. We observe that the value of each histogram bin $[\hat{h}(z_n, x_n)]_j$ over the set of rectangles X can be bounded from below and from above by the number of features with corresponding cluster index that fall into x^{min} and x^{max}, respectively. We denote these bounds by $[\underline{h}_n]_j$ and $[\overline{h}_n]_j$ respectively. Each summand in the second term of (9) can now be bounded from below as follows:

$$c_j\left(\left[\hat{h}(z_n, x_n)\right]_j - [\hat{\mu}]_j\right)^2 \geq \begin{cases} c_j\left([\underline{h}_n]_j - [\hat{\mu}]_j\right)^2 & \text{if } [\hat{\mu}]_j < [\underline{h}_n]_j \\ 0 & \text{if } [\underline{h}_n]_j \leq [\hat{\mu}]_j \leq [\overline{h}_n]_j \\ c_j\left([\overline{h}_n]_j - [\hat{\mu}]_j\right)^2 & \text{if } [\hat{\mu}]_j > [\overline{h}_n]_j \end{cases} \tag{10}$$

This provides us with an efficient way to compute a lower bound over the second term of our objective. As for the first term of (9), in our implementation we model $p(x_n)$ as a simple Gaussian over the relative area of the foreground subwindow, measured as fraction of the image area. The mean of this Gaussian is set to 0.25 and the variance is set to 10^{-5} for all datasets. Therefore, the bound over sets of subwindows can be trivially defined for $\log p(x_n)$. This implies that our complete objective can now be *globally* optimized over $x_n \in \mathcal{X}$ using the branch and bound method for efficient subwindow search described in Lampert et al. (2008).

4.2 Image foregrounds as a set of super-pixels

Modeling foregrounds as rectangular regions is appropriate for contiguous box-shaped objects. For other types of objects, this model may result in the inclusion of random background clutter as part of the window. This is undesirable and is particularly troublesome for highly contoured objects and object classes with large pose variance. To address this concern, we propose a second method of representing foregrounds. Here, each image z_n undergoes bottom-up segmentation once at the start of the clustering procedure and is split into a number of appearance-based segments $\{s_n^1, s_n^2, \ldots, s_n^M\}$. In our work we choose the number of segments (M) to be large enough for the image to be deemed as over-segmented, i.e., each object in the scene is typically split into multiple segments. We refer to these segments as super-pixels. Thus, the goal of finding the foreground becomes equivalent to finding which super-pixels may be part of the foreground. An important property of considering an image as a collection of super-pixels is that, unlike Russell et al. (2006), Duygulu et al. (2002) and several other approaches, this does not require that the entire foreground object region be captured by a single bottom-up segment. Instead, we treat the foreground as a set of super-pixels. Formally, the foreground mask x_n from Fig. 1 is defined by a set of scalar variables $\{x_n^1, x_n^2, \dots, x_n^M\}$, where x_n^i corresponds to segment s_n^i. We treat each x_n^i as a continuous variable such that $x_n^i \in [0,1]$, with the interpretation that higher values imply that the super-pixel s_n^i is to be part of the foreground region and a value close to 0 implies that s_n^i is assigned as part of the background. Formalizing this intuition, we define the foreground image content as $h(z_n, x_n) \equiv \sum_i x_n^i h(z_n, s_n^i)$ where $h(z_n, s_n^i)$ is the histogram of features occurring in a super-pixel s_n^i. As before, we denote with $\hat{h}(z_n, s_n^i)$ the concatenation of K copies of $h(z_n, s_n^i)$. Under this model, we rewrite (9) as

$$\begin{aligned}\arg\max_{x_n} \langle \mathcal{L} \rangle_\gamma = \arg\max_{x_n} \Bigg\{ & \log p\bigg(\sum_i x_n^i P_n^i\bigg) - \sum_{f,g \in S_N} \frac{(x_n^f - x_n^g)^2}{\sigma} \\ & - \sum_{j=1}^{KW} c_j \Bigg(\Bigg[\sum_{i=1}^{M} x_n^i \hat{h}\big(z_n, s_n^i\big) \Bigg]_j - [\hat{\mu}]_j \Bigg)^2 \Bigg\} \end{aligned} \qquad (11)$$

$$\text{subject to:} \quad x_n^i \in [0,1]. \qquad (12)$$

The first two terms on the right side in (11) capture our choice of prior $p(x_n)$ from (9) with σ being a scalar constant. In image z_n, P_n^i is the fraction of pixels belonging to segment s_n^i. For the first term, similar to the treatment of foreground as bounding boxes, we model the normalized size of the foreground as a Gaussian random variable with mean and standard deviations being the same as that for the bounding boxes. The second term penalizes configurations where neighboring segments have widely differing values and thus forces foreground segments to be localized together. In this term, S_N is the set of all pairs of neighboring segments, where we define two segments s_1 and s_2 to be neighbors if there is at least one pair of adjacent pixels (p_1, p_2) in the image such that $p_1 \in s_1$ and $p_2 \in s_2$. Note that, since the segments do not change after the initial image segmentation, neither do neighborhood relationship between segments. It can be seen that (11) is a simple convex optimization objective when x_n^i is allowed to be a real value and hence can be maximized efficiently using quadratic programming.

5 Implementation details

5.1 Image representation

Our representation is based on histograms of quantized SIFT features (Lowe 2004). We experimented with both SIFT descriptors calculated densely over the entire image and also those produced using an interest point detector. Similarly to what reported by Lee and Grauman (2009), we obtained better results using dense descriptors calculated at every pixel in the image. Thus, here we present experiments based only on dense features. As per common practice, we quantize the SIFT descriptors using a vocabulary of visual words generated by running k-means on a set of SIFT descriptors obtained from the collection of input images. Although we have obtained good results even by using directly this bag-of-word representation as input to our method, slightly better performance was achieved by learning a codebook of LDA topics (Blei et al. 2003) over the quantized SIFT features via Gibbs Sampling (Griffiths and Steyvers 2004). Therefore, each image is viewed as a document of visual words generated from a mixture of topics and the final histogram is produced by assigning each quantized SIFT descriptor to its most likely topic.

5.2 Initialization

Our method requires initial estimates of the following parameters: mixture coefficients (π_k), histogram means (μ_k) and variances (Σ_k), as well as foreground masks for all images (x_n).

First, we initialize the foreground masks. In order to do this, we employ a form of co-segmentation (Rother et al. 2006) by matching pairs of images. Specifically, for a pair of images (z_i, z_j), we find the pair of subwindows $(x_{ij}, x_{ji}) \in \mathcal{X} \times \mathcal{X}$ that minimizes a regularized distance between histograms computed from these subwindows as defined by the following objective:

$$\left\| h(z_i, x_{ij}) - h(z_j, x_{ji}) \right\|_1 - C \left\| h(z_i, x_{ij}) + h(z_j, x_{ji}) \right\|_1 \tag{13}$$

where $\|.\|_1$ denotes the L1-norm and C is a hyperparameter trading off the objectives of finding similar histograms and choosing large subwindows (in our implementation C is set to 0.05). It is easy to see that this objective can be minimized using a simple variant of the branch and bound method described in Lampert et al. (2008).

In order to initialize the foreground subwindows for all images, we first sample a subset of T images from the entire dataset. We refer to these T images as seeds. These seeds are chosen through an iterative sampling procedure (described below) that aims at selecting at least one seed image for each object class in the dataset so as to represent all categories. The seed selection works as follows: starting from an empty seed set, at each iteration we add to it a new seed image randomly selected from a candidate set $\mathcal{C}$ of images, which is initially the entire dataset, i.e., $\mathcal{C} \equiv \{z_1, \dots, z_N\}$. Then, we define the foreground x_s of the newly selected seed z_s by performing pairwise matching (as described by (13)) between z_s and all the images in the dataset. This gives us $N - 1$ candidate foreground masks $\{x_{si}\}_{i \neq s}$ for z_s. From this set, we pick the 3rd largest window by area to be the initial mask x_s of the seed image. The intuition behind the choice of picking the 3rd largest window by area is that close matches will result in larger windows and that the largest windows probably contain background regions due to matching to near-duplicates. Before we pick the next seed image randomly from the candidate set $\mathcal{C}$, we eliminate from $\mathcal{C}$ the nearest $\lfloor \frac{N}{T} \rfloor$ images to z_s, using the L2-norm $\|h(z_s, x_s) - h(z_i, x_{is})\|_2$. This helps in obtaining a good coverage of object classes.

Finally, for any image z_i that is not in the seed set, we select its initial foreground x_i from the candidate T subwindows x_{is} by picking the one that yields the smallest matching score among the T matches to the seed images. If the set of seeds includes at least one example of each class, then the best match is likely to occur with a seed of the same class as z_i and the foreground of this match will enclose the correct object.

The above procedure has runtime complexity of $O(TN)$. T is a design parameter which determines how densely we sample the image collection for obtaining good initial windows. The quality of the initial windows improves with increasing value of T. In our experiments, we set T to be $3K$. The same initial windows were used for both the foreground localization methods described in this paper. We found experimentally that initializing our methods with these foregrounds produces better results than when using the boxes obtained with Alexe et al. (2010b).

For initializing the mixture parameters, we tried a variation of careful seeding (Arthur and Vassilvitskii 2007), which we found to be robust against outliers.

6 Experimental results

6.1 Algorithms

There are very few published quantitative evaluations on the task of unsupervised clustering and foreground localization. In this paper, we benchmark the performance of our proposed approach principally against the foreground focus (FF) method published in Lee and Grauman (2009), where results on our intended task are reported.

We do not compare directly to the methods described in Tuytelaars et al. (2010) as these algorithms do not consider the problem of object localization and instead perform image clustering merely based on global features calculated from the entire image. Instead we include as baselines k-means and a Gaussian mixture model applied to whole images (k-means-Whole, GMM-Whole) and to ground truth bounding boxes (k-means-GT, GMM-GT). We also report results for k-means and GMM applied to bounding boxes computed with the bottom-up method described in Alexe et al. (2010b) (k-means-Obj, GMM-Obj) (for each image we use the bounding box having highest probability according to the "objectness" measure).

Finally, we include in our comparison our adaptation of the method described in Russell et al. (2006) (Multi-Seg) for our task. As already discussed, this method was designed for a different objective: it does not explicitly cluster the images or specify which segments are foregrounds. We tried adapting this method to work on our task in two different ways:

1. We ran the code of Russell et al. (2006): for each image I, multiple segmentations were computed and a topic model was fit to the segments. Cluster membership was determined as the topic (T_I) of the segment (S_{best}) with the smallest KL divergence to its topic. Then, to localize the foreground, we selected all segments having T_I as the most probable topic from the segmentation containing S_{best}.
2. We trained the topic model of Russell et al. (2006) on the same super-pixels used as input by our method; then we applied the procedure described in point 1 above for clustering and localization.

We have included the results for option 1 in the plots of Figs. 2 and 3 as the results obtained with option 2 are slightly worse. The only exception are the results on the Pascal dataset in the right-bottom plot of Fig. 2 where we report the performance obtained by using option 2 since option 1 is so expensive that it cannot be run on this large database.

We refer to our two methods for joint foreground clustering and localization as Joint-Subwindow for the version using bounding-boxes (Sect. 4.1) and Joint-Superpixels for the version based on super-pixels (Sect. 4.2).

6.2 Datasets and features

In Lee and Grauman (2009), the authors have evaluated their method on the MSRC-v1 dataset and two subsets (a 4-class and a 10-class collection) of the Caltech101 dataset. MSRC-v1 contains 7 classes with 30 images per class. The 4-class collection—Caltech4—is a subset of the 10-class collection—Caltech10. Both these datasets include the first 50 images per class and the classes were manually selected by the authors of Lee and Grauman (2009) to be categories with high amount of clutter in the images. The 4-class dataset consists of the Faces, Dalmatians, Hedgehogs and Okapi classes, and the 10-class collection contains Leopards, Car_Side, Cougar_face, Guitar, Sunflower and Wheelchair classes in addition to the classes of Caltech-4. Here we report our findings using exactly the same experimental setup and sets of images.

We also report the results on a subset of the Pascal VOC 2007 dataset. For each of the 20 categories in this collection we selected the first 50 images that do not contain objects of the other 19 categories, as determined by the ground truth annotations. We apply this selection strategy in order to reduce the original multi-label classification problem to a standard single-label recognition task. Note that this does not imply that each image contains only one object: as it can be seen from the examples in (8), each image typically contains multiple distinct objects with the only constraint being that it can include only one of the 20 predefined categories of Pascal.

For all datasets, we set the number of foreground clusters, K, to be equal to the number of classes. As traditionally done in unsupervised clustering, we view K as a hyperparameter chosen by the user. The human cost of specifying this value is small and consequently acceptable for most application scenarios. In spite of this, we also present an experiment where we study the sensitivity of our method with respect to this parameter.

All of the results described here are based on a codebook of 50 LDA topics computed from 500 SIFT words and in each experiments we use the same dictionary for all competing methods. The only exception are the results in Table 1 where we study how the performance of our method varies for different representations.

For the Joint-Superpixels method, we generate 20 bottom-up segments for every image using an implementation of normalized cuts (Shi online). To optimize (9), we use the implementation of efficient subwindow search made publicly available by the authors of Nguyen et al. (2009); to optimize (11), we use the ILOG cplex solver from IBM (online).

6.3 Quality of image clustering

We begin by evaluating the quality of clustering in terms of the F-measure metric with respect to the ground truth class labels: $F = \sum_i \frac{N_i}{N} \max_j F'(i, j)$, where N_i is the number of images belonging to class i, $F'(i, j) = \frac{2R(i,j)P(i,j)}{R(i,j)+P(i,j)}$, and $P(i, j)$ and $R(i, j)$ denote precision and recall, respectively, measured for class i and cluster j. The F-measure is a good index of cluster purity with high values indicating that each cluster contains objects predominantly from one class.

Figure 2 summarizes the results obtained on all four datasets. We see that our two methods (Joint-Subwindow and Joint-Superpixels) greatly outperform the k-means and GMM models applied to full images (k-means-Whole, GMM-Whole) on all the datasets. Furthermore, somewhat surprisingly, our approaches also do much better than k-means and GMM applied to the foreground ground truth subwindows available for the Caltech subsets (k-means-GT, GMM-GT). We speculate that this happens because the manual annotations are subjective and unreliable. Particularly in classes with high degree of variance, the human-selected boxes might work against the clustering attempt as the content expressed within the foreground regions of images within the same class might not be similar. Also note that, unsurprisingly, the results of k-means-Obj and GMM-Obj are poor since determining objects from a single image is an ill-defined task.

Furthermore, we see that our two methods outperform Multi-Seg (Russell et al. 2006) and also the results of FF reported in Lee and Grauman (2009), even though this algorithm uses a more sophisticated image representation encoding relative location of features in spatial neighborhoods. The difference in performance is especially noticeable on the most challenging MSRC-v1 dataset, which contains objects at different scales and in different positions within the image.

On the Pascal dataset all methods yield overall much lower clustering accuracy due to the challenges posed by this image collection which includes classes exhibiting large variations

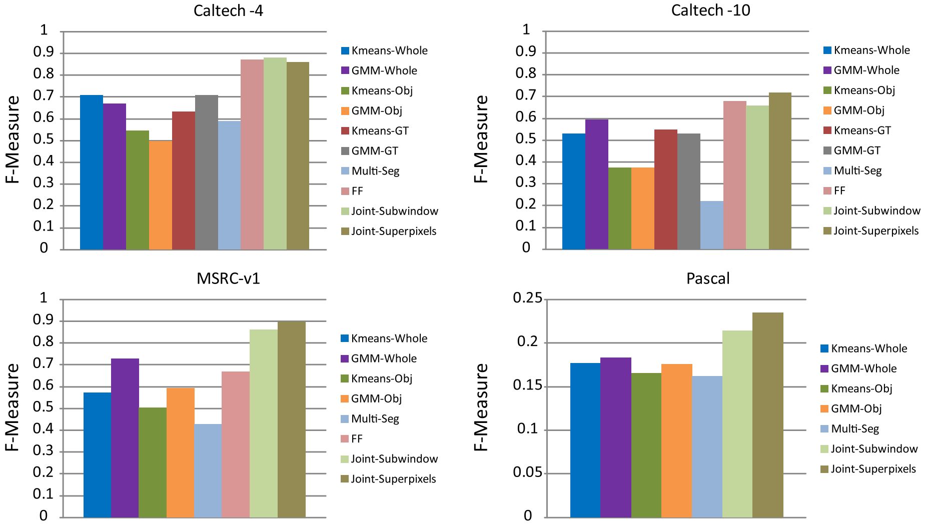

Fig. 2 The quality of image clustering in terms of the F-measure metric for the four datasets. The compared methods are k-means applied to whole images (k-means-Whole), ground truth subwindows (k-means-GT) and object boxes computed using (Alexe et al. 2010b) (k-means-Obj). GMM is also applied with the same settings: (GMM-Whole), (GMM-GT) and (GMM-Obj). The figure also includes results for Multi-Seg (Russell et al. 2006), and FF (Lee and Grauman 2009). Our proposed algorithms of joint clustering and localization are Joint-Subwindow (Sect. 4.1) and Joint-Superpixels (Sect. 4.2)

Table 1 F-measure obtained by our Joint-Superpixels method using different foreground representations

Dataset	SIFT-50	SIFT-500	LDA-50 over SIFT-500
Caltech-4	0.825	0.846	0.86
Caltech-10	0.65	0.5	0.72
MSRC-v1	0.719	0.68	0.90

in scale and appearance. However, on this difficult benchmark our approach provides clearly superior performance over all the other methods considered in this comparison.

We have also evaluated the performance of our Joint-Superpixels method when representing the foreground region as a histogram of quantized SIFT features rather than LDA topics. Table 1 shows the results for histograms defined over dictionaries of both 50 as well as 500 centroids learned from SIFT vectors. We also include the accuracy of our default system based on a dictionary of 50 LDA topics learned over 500 SIFT words. From these results we see that our system achieves good clustering accuracy even when directly applied to histograms of quantized SIFT. However, it is clear that the topic representation yields improvements in performance particularly for the challenging MSRC-v1 dataset.

6.4 Foreground localization

We now proceed to evaluate our approach in terms of object localization accuracy. In Lee and Grauman (2009), the authors determine the quality of the foreground localization by examining the normalized sum of the weights inside the ground truth foreground. While their performance on this metric does indicate that the foreground features get higher weight than background features, there is no clear way of determining the actual locality and

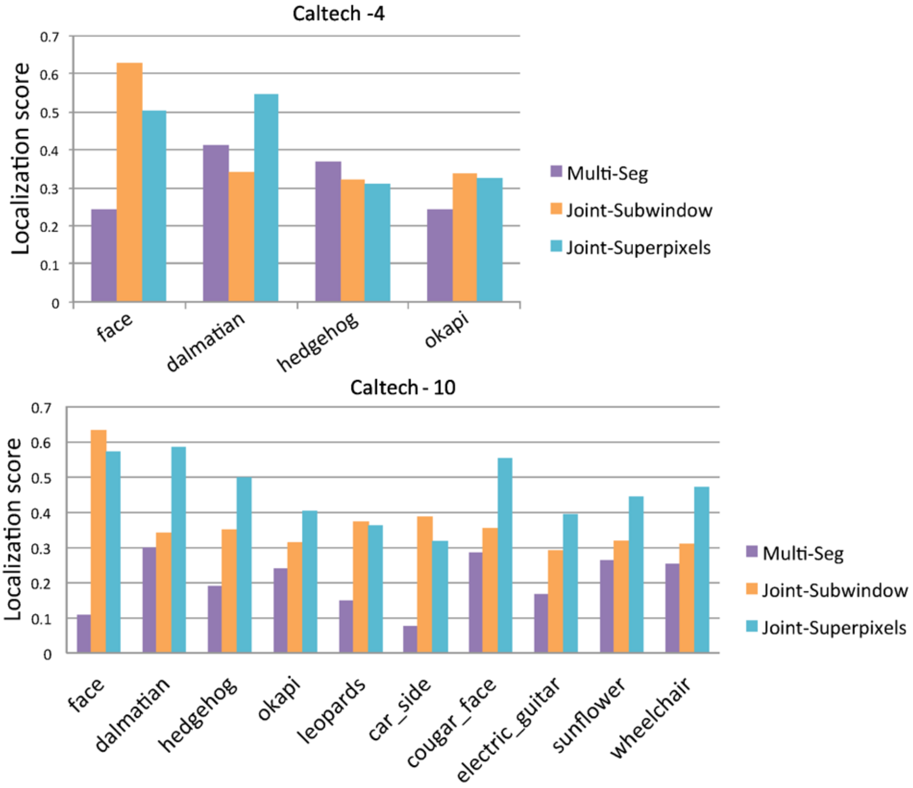

Fig. 3 Average localization scores achieved by our methods over all images from each ground truth class in the 4-class and the 10-class subsets of Caltech. We also show the localization scores achieved by Multi-Seg. Please see text for more details

extent of the foregrounds in the images. Furthermore, with their metric, it is possible to get a high score by having just a few highly weighted foreground features. Instead, it is useful for many applications to determine the actual location and size of the foreground. Our algorithms generate a natural solution to this requirement in the form of foreground bounding boxes for Joint-Subwindow and foreground segments corresponding to super-pixels with high foreground scores[1] in the case of Joint-Superpixels. We measure the quality of the foreground localization by using a metric commonly used in object detection: $J_n = area(x_n \bigcap x_n^{GT})/area(x_n \bigcup x_n^{GT})$ where x_n^{GT} is the ground truth for the object in image n. To evaluate the Joint-Subwindow method we use the bounding box ground truth provided for the images in Caltech 101. For the methods based on super-pixels (Joint-Superpixels and Multi-Seg) we use the full object contour ground truth provided.

Figure 3 shows the average localization scores per class achieved with our methods on the 4-class and the 10-class subsets of Caltech101. It is clear that Joint-Superpixels localizes the foreground more accurately than Joint-Subwindow, even though there is not much difference in terms of F-measure scores between our two methods. Conversely, the average localization scores achieved by Russell et al. (2006) are clearly inferior to those computed by

[1] We deem a super-pixel s_n^i to be part of the foreground if $x_n^i > 0.3$ at the end of the EM run.

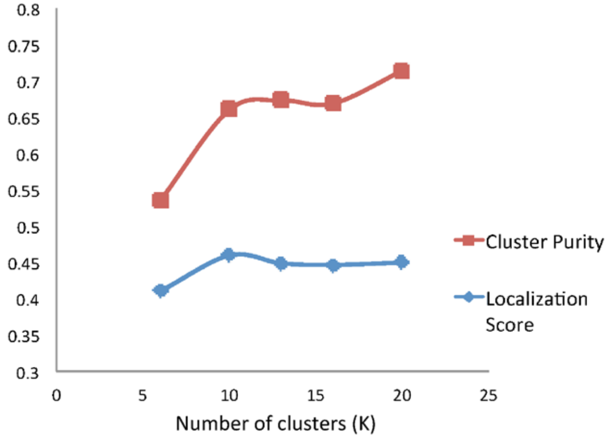

Fig. 4 Cluster purity and localization scores for different values of K (the number of clusters) using the Joint-Superpixels method on the Caltech-10 dataset. As K increases, precision increases but the localization scores remain similar. This suggests that the method is fairly robust to the choice of K

both our methods. While studying the scores, we want to emphasize that these are calculated with respect to the manually annotated ground truth. As we have already seen in the case of bounding boxes, they are somewhat arbitrary. In our methods, foreground detection is optimized for image clustering. So it is reasonable to get foregrounds which are inconsistent with the ground truth, but nevertheless play a role in improving image clustering.

The F-measure results in Fig. 2 show that our methods for joint clustering and localization outperfom a two-step process of localization (using objectness boxes) followed by clustering. We also examined whether the joint approach yields better *localization* than a disjoint approach of clustering followed by common object detection within each cluster. In order to test this hypothesis, we ran the implementation of the classcut technique (which performs class specific segmentation using a set of images known to contain an object from the class) made publicly available by the authors of Alexe et al. (2010a), on the k-means clusters produced for the Caltech-10 dataset. We tried different parameterizations (with and without objectness) for the classcut implementation and the best results were obtained by running classcut on full images, as reported by the authors in their paper. The mean localization score obtained by running classcut was 0.44. In comparison, our method (Joint-Superpixels) produced a mean localization score of 0.46 over the entire dataset. Thus our method marginally outperforms the classcut method applied to k-means clusters in terms of localization. However, it is important to note that applying localization after clustering using a two-stage process does not improve the quality of clustering (we remind the reader that on Caltech-10 the clustering F-measure of our approach is 0.72 versus a value of 0.52 for k-means). Furthermore, classcut is a very expensive method (it has a runtime of over 16 hours for the Caltech-10 dataset) and as a result, it may not be suitable as a component in an iterative procedure over large datasets.

As mentioned earlier, we assume that K—the number of output clusters for the method—is a hyperparameter set by the user. In Fig. 4, we study the effect of the choice of K on cluster purity and foreground localization for the Joint-Superpixels method. The fmeasure metric is unsuitable for evaluating performance of the model when the number of clusters is not equal to the number of ground truth classes. If $class_j$ is the set of samples in the dataset belonging to the jth ground truth class and $cluster_k$ is the set of elements in the kth output cluster of the clustering procedure, then we use the metric of cluster purity which is computed as:

$$Purity = \frac{1}{N}\sum_{k}^{K}\max_{j}|cluster_k \cap class_j| \tag{14}$$

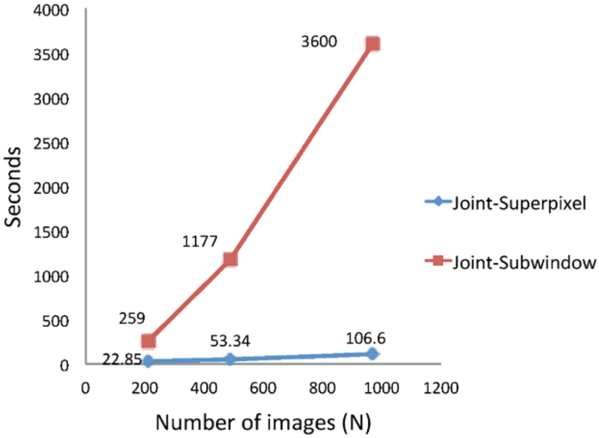

Fig. 5 The runtimes of our EM algorithms on MSRC-v1 ($N = 210$), Caltech-10 ($N = 489$) and Pascal-20 ($N = 968$) datasets. The graph shows that both our proposed approaches for foreground localization scale approximately linearly with the number of images and the runtimes are largely independent of the hyperparameter K (the number of clusters)

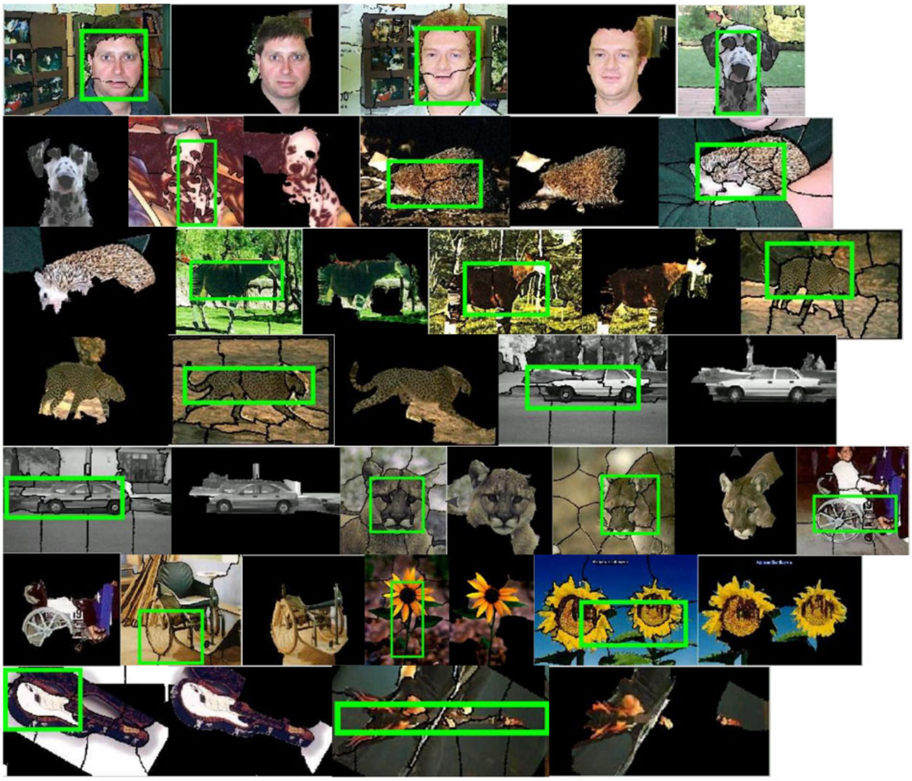

Fig. 6 Examples of foreground prediction in images from the 10-class subset of Caltech101. The *image on the left of each pair* shows the super-pixels obtained through bottom-up segmentation. The *box in green* is the foreground extent predicted by our Joint-Subwindow method. The *image on the right of each pair* shows the foreground discovered as a collection of super-pixels (selected if $x_n^i > 0.3$) by our Joint-Superpixels method

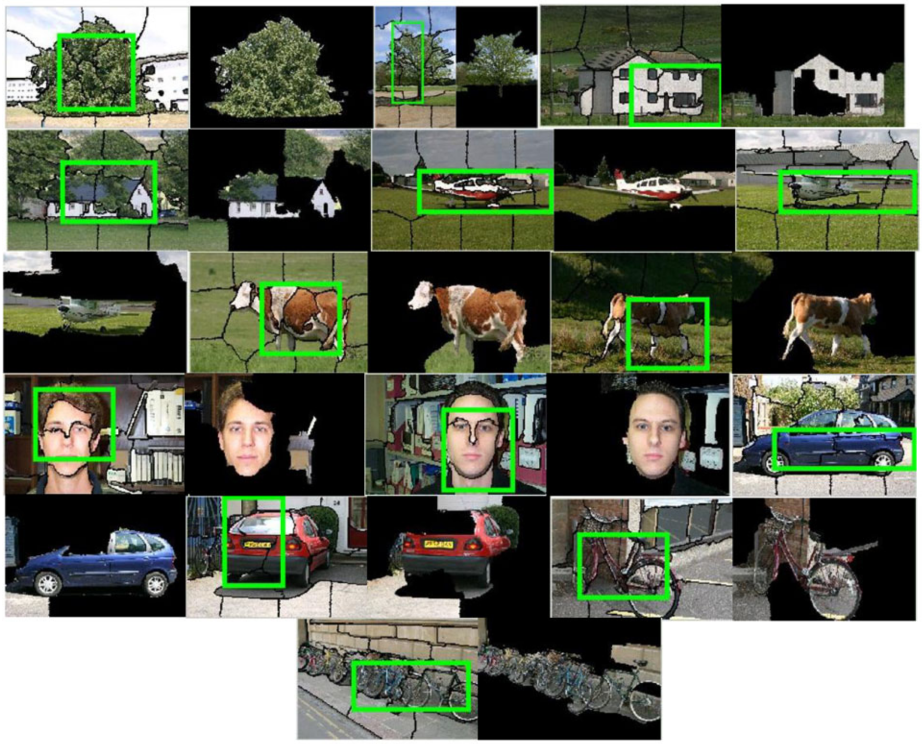

Fig. 7 Sample results for the MSRC-v1 dataset. The *box in green* is the foreground extent predicted by Joint-Subwindow. The *image on the right of each pair* shows the foreground discovered as a collection of super-pixels (selected if $x_n^i > 0.3$) by Joint-Superpixels

As we increase K we find purity slowly increasing (as expected) but localization quality remaining stationary. This shows that the model is not particularly sensitive to the exact value of K used.

Figures 6, 7, 8 show some examples of foreground prediction for our method both in terms of discovered subwindows and selected bottom-up segments. Please refer to supplemental data at VLG (2012) for additional visualizations.

6.5 Runtime analysis

Finally, we would like to comment on the computational costs of our approach. We found that typically our EM algorithms converge in less than 10 iterations. At the heart of the EM procedure are the two efficient foreground localization methods used to optimize (9) and (11). The branch and bound method for subwindow discovery optimizing (9) is typically sublinear in the number of pixels (Lampert et al. 2008) while the quadratic programming objective of (11) can be solved in polynomial time (in the number of segments). The plot in Fig. 5 shows the runtime of the EM procedure for the datasets of MSRCv1 (210 images), Caltech-10 (489 images) and Pascal-20 (968 images). The runtimes are computed using a single-core (3.2 GHz processor). From this plot, we can see that EM scales linearly with the number of images and is fairly independent of model complexity (K). This is to

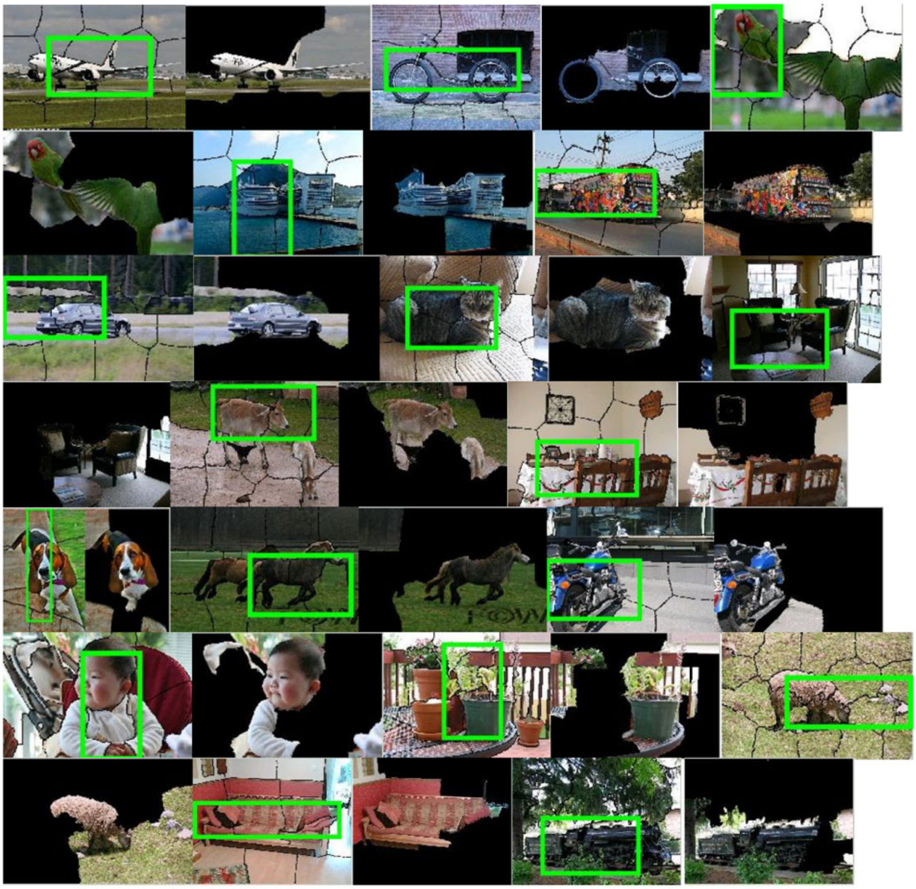

Fig. 8 Sample results for the Pascal dataset. The *box in green* is the foreground extent predicted by Joint-Subwindow discovery. The *image on the right of each pair* shows the foreground discovered as a collection of super-pixels (selected if $x_n^i > 0.3$) by Joint-Superpixels

be expected since the most time consuming operation within EM is the localization procedure which operates over individual images and is dependent on K only in the creation of concatenated histogram in (9). The plot suggests that the Joint-Subwindow is much more expensive than Joint-Superpixels. However, there are two important details to keep in mind: (1) the implementation of the branch and bound method may not be optimized; (2) there is a fixed initial cost to segment the images which is not included in the graph. However, despite this, we do believe that Joint-Superpixels method is more scalable as it quantizes each image (irrespective of the size) to a fixed number of segments, thereby making the optimization less expensive. Both these approaches are significantly faster than sliding window methods which typically have cost $O(n^4)$ for an $n \times n$ image. Multi-Seg (Russell et al. 2006) is significantly more expensive than our approach while performing substantially worse at the task. As a frame of reference, the implementation for Multi-Seg, made available by the authors (Russell et al. 2006) takes 5500 seconds on the MSRC-v1 dataset and 17000 seconds on Caltech-10. This is in addition to the time needed to segment the images in the dataset, which is an operation much more expensive than our super-pixel segmentation as their method needs to run multiple segmentations.

The main computational cost of our algorithms is the initialization procedure for the foregrounds, which is a one-time cost not included in the graph. In this step we perform $O(TN)$ number of pairwise image co-localizations through branch and bound. By using small values of T ($K < T \ll N$), the cost remains reasonable. In our experiments, we set $T = 3K$. As a reference, on MSRC-v1 using a cluster of 40 cores the initialization takes 300 seconds. It may be possible to reduce this cost further by downsampling the images for initialization. It is also important to note that the other approaches considered in our comparison are even more expensive. Lee and Grauman (2009) does not provide details of runtime nor a software implementation that we can evaluate. However, *each iteration* in their method technique is $O(N^3)$, while also operating on complex features that are very expensive to compute.

7 Conclusions

Unsupervised foreground discovery is an important but difficult means of extracting structure from unlabeled image datasets. In this work, we have developed a probabilistic method to perform simultaneous image clustering and foreground localization in unlabeled collections. We have shown that harnessing the natural synergy between the two tasks leads to improved performance at both the tasks. Our approach can efficiently localize object foregrounds without resorting to expensive sliding window mechanisms or relying on the unrealistic expectation that brittle bottom-up segmentation will yield segments corresponding to objects in the scene. We note that while our foreground appearance model is admittedly simple, it is precisely this model simplicity that allows us to cast foreground clustering and localization elegantly as a single *joint* optimization. We believe we are the first to propose such a joint optimization for the two tasks. Furthermore, we empirically show that the approach outperforms methods that use more complex foreground models but that have to resort to alternation between distinct objectives (e.g., Lee and Grauman 2009) or to a two-step solution (e.g., Russell et al. 2006) to solve the problem. We believe there is high value in simple models shown to perform well in practice. In the future we are interested in extending this work to videos. Our probabilistic formulation also enables straightforward integration of non-visual cues such as text or tags associated to the images, which may yield more semantically meaningful clusters. The software implementing our algorithm is made available at (VLG 2012).

Acknowledgements The authors world like to thank Minh Hoai Nguyen for sharing his code for co-segmentation of image pairs. This work was funded in part by NSF CAREER award IIS-0952943, by grants from the Office of Naval Research and by the Defense Advanced Research Projects Agency.

References

Alexe, B., Deselaers, T., & Ferrari, V. (2010a). Classcut for unsupervised class segmentation. In *European conference on computer vision*. http://dl.acm.org/citation.cfm?id=1888150.1888181.

Alexe, B., Deselaers, T., & Ferrari, V. (2010b). What is an object? In *IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2010.5540226.

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2003). Support vector machines for multiple-instance learning. In *Advances in neural information processing systems* (Vol. 15, pp. 561–568). Cambridge: MIT Press.

Arthur, D., & Vassilvitskii, S. (2007). k-Means++: the advantages of careful seeding. In *ACM-SIAM symposium on discrete algorithms*. http://dl.acm.org/citation.cfm?id=1283383.1283494.

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*. http://dl.acm.org/citation.cfm?id=944919.944937.

Borenstein, E., Sharon, E., & Ullman, S. (2004). Combining topdown and bottom-up segmentation. In *CVPR workshop on perceptual organization in computer vision*. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1384838.

Chen, Y., & Wang, J. (2004). Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*. http://dl.acm.org/citation.cfm?id=1005332.1016789.

Chum, O., & Zisserman, A. (2007). An exemplar model for learning object classes. In *IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2007.383050.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2005.177.

Deselaers, T., Alexe, B., & Ferrari, V. (2010). Localizing objects while learning their appearance. In *European conference on computer vision*. http://dl.acm.org/citation.cfm?id=1888089.1888124.

Duygulu, P., Barnard, K., deFreitas, N., & Forsyth, D. (2002). Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In *European conference on computer vision*. http://dl.acm.org/citation.cfm?id=645318.649254.

Fergus, R., Fei-Fei, L., Perona, P., & Zisserman, A. (2005). Learning object categories from Google's image search. In *International conference on computer vision*. doi:10.1109/ICCV.2005.142.

Fritz, M., & Schiele, B. (2008). Decomposition, discovery and detection of visual categories using topic models. In *IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2008.4587803.

Griffiths, T., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Science*. doi:10.1073/pnas.0307752101.

IBM (online). IBM ilog cplex optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

Itti, L., & Koch, C. (2001). Computational modeling of visual attention. *Nature Reviews Neuroscience*. doi:10.1038/35058500.

Kim, G., & Torralba, A. (2009). Unsupervised detection of regions of interest using iterative link analysis. In *Advances in neural information processing systems* (Vol. 22, pp. 961–969).

Lampert, C., Blaschko, M., & Hofmann, T. (2008). Beyond sliding windows: object localization by efficient subwindow search. In *IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2008.4587586.

Lee, Y., & Grauman, K. (2009). Foreground focus: unsupervised learning from partially matching images. *International Journal of Computer Vision*. doi:10.1007/s11263-009-0252-y.

Leibe, B., & Schiele, B. (2003). Interleaved object categorization and segmentation. In *BMVC*, pp. 759–768.

Liu, C., Yuen, J., & Torralba, A. (2010). Sift flow: dense correspondence across different scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. doi:10.1007/978-3-540-88690-7_3.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. doi:10.1023/B:VISI.0000029664.99615.94.

Marr, D. (1982). *Vision: a computational investigation into the human representation and processing of visual information*. New York: Freeman.

Nguyen, M., Torresani, L., de la Torre, F., & Rother, C. (2009). Weakly supervised discriminative localization and classification: a joint learning process. In *International conference on computer vision*. doi:10.1109/ICCV.2009.5459426.

Rother, C., Minka, T., Blake, A., & Kolmogorov, V. (2006). Cosegmentation of image pairs by histogram matching—incorporating a global constraint into MRFS. In *IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2006.91.

Russell, B., Efros, A., Sivic, J., Freeman, W. T., & Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2006.326.

Shi, J. (online). Normalized cuts segmentation code. http://www.cis.upenn.edu/~jshi/software/.

Sudderth, E., Torralba, A., Freeman, W., & Willsky, A. (2005). Learning hierarchical models of scenes, objects, and parts. In *International conference on computer vision*. doi:10.1109/ICCV.2005.137.

Tu, Z., Chen, X., Yuille, A., & Zhu, S. (2005). Image parsing: unifying segmentation, detection and recognition. *International Journal of Computer Vision*. doi:10.1109/ICCV.2003.1238309.

Tuytelaars, T., Lampert, C., Blaschko, M., & Buntine, W. (2010). Unsupervised object discovery: a comparison. *International Journal of Computer Vision*. doi:10.1007/s11263-009-0271-8.

VLG (2012). Finding what is where—supplementary material. http://vlg.cs.dartmouth.edu/learningwhatiswhere/.

Yu, S., & Shi, Y. (2003). Object-specific figure-ground segregation. In *IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2003.1211450.