Weakly Supervised Object Detection with Convex Clustering

Hakan Bilen[†§] Marco Pedersoli[†] Tinne Tuytelaars[†]
[†] ESAT-PSI / iMinds, KU Leuven, Belgium
[§] VGG, Dept. of Eng. Sci. University of Oxford
firstname.lastname@esat.kuleuven.be

Abstract

Weakly supervised object detection, is a challenging task, where the training procedure involves learning at the same time both, the model appearance and the object location in each image. The classical approach to solve this problem is to consider the location of the object of interest in each image as a latent variable and minimize the loss generated by such latent variable during learning. However, as learning appearance and localization are two interconnected tasks, the optimization is not convex and the procedure can easily get stuck in a poor local minimum, i.e. *the algorithm "misses" the object in some images. In this paper, we help the optimization to get close to the global minimum by enforcing a "soft" similarity between each possible location in the image and a reduced set of "exemplars", or clusters, learned with a convex formulation in the training images. The help is effective because it comes from a different and smooth source of information that is not directly connected with the main task. Results show that our method improves a strong baseline based on convolutional neural network features by more than 4 points without any additional features or extra computation at testing time but only adding a small increment of the training time due to the convex clustering.*

1. Introduction

The standard approach for supervised learning of object detection models requires the annotation of each target object instance with a bounding box in the training set. This fully supervised paradigm is tedious and costly for large-scale datasets. The alternative but more challenging paradigm is to learn from the growing amount of noisily and sparsely annotated visual data available. In this work, we focus on the specific "weakly supervised" case when the annotation at training time is restricted to presence or absence of object instances at image-level.

An ideal weakly supervised learning (WSL) for object detection is expected to guide the missing annotations to a solution that disentangles object instances from noisy and cluttered background. The standard WSL paradigm alternates between labeling the missing annotations and learning a classifier based on these labellings in a spirit similar to Expectation Maximization (EM). Due to the missing annotations, this optimization is non-convex and therefore prone to getting stuck in a local minimum (and sensitive to initialization). In practice, although the optimal solution would lead to a perfect localization of the objects, guiding the optimization to such point is quite challenging and the weakly supervised detection performance is far from the supervised one. We can comprehend the importance of properly guiding the optimization by considering the extreme case of perfect localization, *i.e.* the fully supervised case, which can be considered an upper bound in terms of performance. For instance, the best reported detection performance that relies on the convolutional neural network (CNN) features of [6] on the Pascal VOC 2007 dataset [7] is 58.5% ([10]) mean average precision (AP) in the fully supervised case while it is 30.9% ([30]) in the weakly supervised case. Thus improving the optimization of our non-convex problem is vital because it would directly lead to better detection performance.

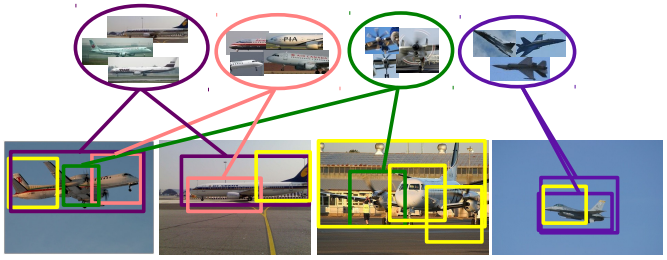

Figure 1. **An illustration of our learning model:** In the top row, we show clusters of objects and object parts that are simultaneously learned with the detectors during training. Our method encourages highly probable windows to be similar among them through the jointly learned clusters during training. The colored lines indicate similarity between windows and clusters. Best viewed in color.

Here, we investigate a possible way to improve the opti-

mization by imposing similarity among objects of the same class. A typical application where such similarity is exploited is co-segmentation [13, 24], which is the task of localizing similar objects in a set of images. The underlying principle behind co-segmentation is to search for a portion of the image that is the most similar among the set of given images. If the visual descriptors of the objects are similar among the images and the optimization function is smooth enough, then all the objects in the images can be properly localized. While the same assumption is also valid for the weakly supervised object detection problem, most state-of-the-art algorithms do not enforce directly similarity among the objects. Instead they follow an iterative and discriminative approach: first a classifier on the initially chosen image parts is learned and then the most discriminative portions of the image are found by applying the learned classifier. In fact, while learning a classifier on the previously chosen image portions and imposing distinctiveness from the background, the similarity among them usually weakens, as it is not explicitly enforced. Therefore, using an additional similarity channel can help to avoid overfitting on the current localization hypotheses and to better guide the weakly supervised localization.

To address the overfitting, previous work has developed a number of different strategies. Cinbis *et al*. [4] use a multi-fold splitting of the training set to prevent getting stuck to wrong labellings in the previous iteration. Deselaers *et al*. [5] use the similarity principle by enforcing pairwise connections among the chosen windows in the training data. In this paper we also use similarity, but in a novel and different way. First, we want to use a similarity measure that is "local". Enforcing a global similarity among all the data samples (*e.g*. distance to the average of the samples) is a too strict assumption that often does not hold, especially in modern and difficult datasets due to intra-class and viewpoint variations. Second, in contrast to [5] we want a similarity measure that is "smooth" over the samples, so that it is easier to optimize and it can support multiple and different hypotheses simultaneously. To this end, we do not limit our hypotheses to the object only but also include parts of it. It is well known that certain classes have parts that share appearance in almost all samples, but globally the objects can have a quite different appearance (*e.g*. bicycles can be very different but their wheels are generally very similar among different instances). Finally, we want our method to be scalable. Considering the exponential number of possible hypotheses when considering also object parts, we want to avoid the expensive CRF optimization needed in [5].

In this paper we propose to couple a smooth discriminative learning procedure as proposed in our earlier work [2] with a convex clustering algorithm [17]. While the discriminative learning estimates a model to best separate positive and negative data, the clustering searches for a small set of exemplars. These exemplars that best describe our training data are not directly forced to be the localization hypotheses but they are selected based on the probability of being part of the object. This indirectly enforces the localized hypotheses to be similar to one another (similar to the cluster centers) and therefore it is a way to enforce local similarity without the need of the expensive pairwise CRF. Furthermore, the optimal number of clusters is automatically selected by the algorithm. This also allows the clustering procedure to optimally adapt to the new localization of object instances at any point of the learning and, due to the convexity of the optimization, it does not depend on the initialization. This idea is illustrated in Fig.1.

The remainder of the paper is structured as follows. Section 2 discusses the related work. Section 3 explains the inference and learning procedures. Section 4 details the experiments on the PASCAL VOC 2007 dataset [7] and Section 5 concludes the paper.

2. Related Work

Non-convex optimization. To alleviate the shortcomings of the non-convex optimization problem, previous work mainly focused on smoothing the latent SVM formulation [2, 14, 20, 27], developing initialization strategies [4, 16] or regulating the latent space [2]. Joulin *et al*. [14] propose a weakly supervised learning formulation that is based on a convex relaxation of a soft-max loss and show that such a learning is less prone to get stuck in a local minimum. Similarly, Song *et al*. [27] smooth the latent SVM formulation of [8] by applying Nesterov's smoothing technique [22]. As a matter of fact, we use the soft-max formulation that has been proposed in [2]. This formulation does not require initialization of missing annotations, enables us to use a quasi-Newton optimization and thus leads to a faster convergence. Kumar *et al*. [16] propose an iterative self-paced learning algorithm that iteratively selects a set of easy samples and learns a new classifier. Song *et al*. [27] initialize the object locations via a sub-modular clustering method. Additionally, Bilen *et al*. [2] propose a posterior regularization formulation that regularizes the latent (object location) space by penalizing unlikely configurations based on symmetry and mutual exclusion of objects. In fact, our approach can also be seen as a regularization technique that enforces similarity between object windows. Although we did not include them in our experiments, symmetry and mutual exclusion can also be used in our method.

Convex clustering. The clustering formulation of our method builds on the work of [17], that casts a clustering problem into a convex minimization by assigning to each sample a sparse distribution of weights that represent their importance. In contrast to non-convex clustering methods

[11] such as k-means and Gaussian mixture model, the algorithm is guaranteed to converge to the global minimum and does not need manual setting the number of clusters. These characteristics are important for our method because they avoid the optimization to get stuck in a poor local minimum and it is independent of the initialization. In Section 4.2 we compare the performance of our algorithm when using k-means clustering and convex clustering. While [17] fits well into our learning due to its probabilistic (soft assignment to clusters) aspect, it is worth mentioning other convex clustering algorithms [3, 15] in the literature. Bradley *et al*. [3] pose the clustering problem into a sparse coding formulation and propose a convex relation of the sparse coding formulation. Komodakis *et al*. [15] pose the same problem as an NP-hard linear integer program and use an efficient linear programming algorithm to solve it.

Clustering in WSL. Recent literature on weakly supervised object detection [27, 28, 30] uses clustering to initialize their latent variables (*i.e*. object windows, part configurations and sub-categories respectively) and learns object detectors based on this initialization. In contrast, our method iteratively refines discriminative clusters that help to localize object instances better in the following iterations. Song *et al*. [27, 28] formulate a discriminative submodular algorithm to discover an initial set of image windows and part configurations respectively that are likely to contain the target object. Wang *et al*. [30] apply a latent semantic discovery via probabilistic latent Semantic Analysis (pLSA) on the windows of positive samples and further employ these clusters as sub-categories. It assumes that the clustering algorithm can find a single compact cluster for the foreground (the object itself) class and multiple ones for the related background (*e.g.* aeroplane - sky and trees). The algorithm requires a careful tuning of cluster numbers to obtain good clusters for each category. In contrast, our method only focuses on the intra-class variance in the foreground via the obtained clusters and does not explicitly model the related background. In addition, our learning automatically determines the optimal number of clusters. The modelling of related background is complimentary to our method and can be expected to further improve our performance.

Learning sub-categories. Our formulation also bears some similarities to the work of [1, 8, 12] that simultaneously cluster the positive samples into sub-categories and learn to separate each cluster from the negative samples. Using multiple sub-categories can also be used in our method and that may further improve results. However, in our case we focus on a more challenging problem in which we are not given the ground truth bounding boxes nor the sub-category membership of positive samples. Learning jointly to cluster, localize and classify remains a challenge.

3. Inference and Learning

Problem Formulation. Our goal is to detect the locations of the objects of a target class (*e.g*. "bicycle", "person"), if there is any, in a previously unseen image. To do so, we learn an object detector for the target class by using a set of positive images (images where at least one object of the target class is present) and negative images (images where there is no object of the target class present). As the locations of the target objects in the positive images are not given, we formulate the task in a latent support vector machine (LSVM) formulation [8, 31] where we aim to find the latent parameter (object window) for each training sample that best discriminates positive images from negative ones. In general the object of the target class is the region of the image that is the most similar among positive images. Thus, with this procedure, we jointly learn the location of object instances for each positive training image and a detector that is able to localize that object. In the following part of this section we define the problem in a formal way.

Let $x \in \mathcal{X}$, $y \in \{-1, 1\}$ and $h \in \mathcal{H}$ denote an image, its binary label and the object location (bounding box) respectively. To generate the set of possible object locations ($\mathcal{H}$), we use the selective search method of Uijlings *et al*. [29] which produces around 1,500 windows per image. This helps us to speed-up our inference and to avoid many background regions. To represent the candidate windows, we rely on the powerful Convolutional Neural Network features of [6] and denote the feature vector for the window h of the image x with $\phi(x, h)$.

To detect the presence y and the location h of target objects in an unseen image x for a given detector defined by a vector of parameters w, we maximize a linear prediction function as:

$$\{y^*, h^*\} = \arg\max_{\substack{y \in \mathcal{Y} \\ h \in \mathcal{H}}} w \cdot \Phi(x, y, h), \tag{1}$$

where $\Phi(x, y, h)$ is a joint feature vector:

$$\Phi(x, y, h) = \begin{cases} \phi(x, h) & \text{if } y = 1 \\ \vec{0} & \text{if } y = -1. \end{cases}$$

In words, the prediction rule (1) labels the image x as negative, if the score of the best window (h^*) is not positive.

To learn w, we first define an objective function $\mathcal{L}$ on a set of training samples $\mathcal{S} = \{(x^i, y^i), i = 1, \ldots, N\}$ and minimize it with respect to w:

$$\mathcal{L}(w, \mathcal{S}) = \mathcal{L}_R(w) + \lambda \mathcal{L}_m(w, \mathcal{S}) \tag{2}$$

where $\mathcal{L}_R$ is the standard l_2 regularization defined as $\frac{1}{2}\|w\|_2^2$. $\mathcal{L}_m$ is a margin loss that is explained in the remainder of this section and includes the main contribution of our work. Finally, λ defines the trade-off between regularization and loss.

Review of latent SVM. We want our object models to score high for positive images (*i.e.* $y = 1$) and low for negative images (*i.e.* $y = -1$). To train such object models that can separate between positive and negative samples, a common formulation to measure the mismatch between the image, label and window is the max-margin latent SVM (LSVM) [31]:

$$\begin{aligned} l_{mm}(w, x^i, y^i) = &\max_{y,h} \big(w \cdot \Phi(x^i, y, h) + \Delta(y^i, y)\big) \\ &- \max_{h} w \cdot \Phi(x^i, y^i, h) \end{aligned} \tag{3}$$

where $\Delta(y^i, y)$ is zero-one error, *i.e.* $\Delta(y^i, y) = 0$ if $y = y^i$, 1 else. This formulation aims to separate the highest scoring window h from the other configurations. However, this formulation has certain shortcomings for the object detection task: (i) it can only choose one window for each positive image, and this limits the learning to leverage multiple object instances, (ii) the optimization is sensitive to initialization of latent parameters for positive images. Therefore, we use a smoother learning method, the soft-max latent SVM (SLSVM) formulation of [2] that can consider multiple object instances in a single image and does not require initialization of latent parameters. The soft-max term l_{sm} is given as:

$$\begin{aligned} l_{sm}(w, x^i, y^i) = &\frac{1}{\beta} \log \sum_{y,h} \exp\big(\beta w \cdot \Phi(x^i, y, h) + \beta \Delta(y^i, y)\big) \\ &- \frac{1}{\beta} \log \sum_{h} \exp\big(\beta w \cdot \Phi(x^i, y^i, h)\big) \end{aligned} \tag{4}$$

where β is a tunable temperature parameter. It can be shown that Eq.(4) reduces to the max-margin formulation of [31], as $\beta \to \infty$. We set this parameter to 1 in all our experiments. The margin loss for the training set is then $\mathcal{L}_m(w, \mathcal{S}) = \sum_{i=1}^{N} l_{sm}(w, x^i, y^i)$.

Convex Clustering. Now, we want to introduce in the objective function an additional term $\mathcal{L}_c$ that enforces similarity among the selected windows so that the new objective is:

$$\mathcal{L}(w, \mathcal{S}) = \mathcal{L}_R(w) + \lambda \mathcal{L}_{sm}(w, \mathcal{S}) + \gamma \mathcal{L}_c(w, \mathcal{S}). \tag{5}$$

However, enforcing similarity is a challenging task because: (i) in the absence of annotated objects, it is not clear between which window pairs to enforce similarity, and (ii) object categories may contain significant variance in appearance and forcing a global similarity among all windows can hurt performance.

To address the first challenge, we avoid a hard decision for choosing an object window and use a soft measure that gives a probability of a window h of image x for the target object class:

$$p(h|x, w) = \frac{\exp\{\beta w \cdot \phi(x, h)\}}{\sum_{h \in \mathcal{H}} \exp\{\beta w \cdot \phi(x, h)\}}. \tag{6}$$

To mitigate the second problem, we enforce similarity between object windows and "representative" clusters in positive training images instead of between each object window pair. For the sake of brevity, we introduce a new variable u to denote window h of image x. $\mathcal{U} = \mathcal{S}^+ \times \mathcal{H}$ denotes the set of possible windows from the set of positive training images S^+. We learn scalar weights q_u to measure how representative a window u:

$$\sum_{u \in \mathcal{U}} q_u = 1 \text{ s.t. } q_u \geq 0,$$

.

Finally, inspired by [17], we propose a clustering term that enforces such similarity:

$$\mathcal{L}_c = -\sum_{u \in \mathcal{H}} p(u, w) \log\Big(\sum_{u' \in \mathcal{U}} q_{u'} e^{-\alpha \mathrm{d}_\phi(u, u')}\Big) \tag{7}$$

where $\mathrm{d}_\phi(u, u') = \|\phi(u) - \phi(u')\|_2$ is the Euclidean distance between two window representations $(\phi(u), \phi(u'))$. α is a positive temperature parameter and controls the sparseness of the q terms. The convex clustering term penalizes configurations with discriminative windows of high probability $(p(u, w))$ far from the important clusters (windows h with high q_u).

Moreover, the term $\mathcal{L}_c$ has two desirable properties: (i) it is convex given w so it is guaranteed to find the optimal solution, and (ii) it results in a sparse selection of clusters (window u with q_u greater than zero). Thus it automatically finds the number of clusters which is optimal for the given α.

Optimization. We minimize the objective function $\mathcal{L}$ in (5) iteratively in two steps with coordinate descent. We first initialize the cluster weights q uniformly, fix them, and minimize $\mathcal{L}$ for w. As our objective function is smooth, our optimization can benefit from the quasi-Newton method L-BFGS [18] which we found faster and more accurate than stochastic gradient descent. In the next step, we fix the found w and optimize $\mathcal{L}_c$ for q. To update the vector q, we use the iterative method as in [17] which is guaranteed to find the global optimum. We define a similarity measure $s_{u,u'} = e^{-\alpha \mathrm{d}_\phi(u, u')}$ and introduce two auxiliary vectors z and η:

$$z_u^{(t)} = \sum_{u' \in \mathcal{U}} s_{u,u'} q_{u'}^{(t)}, \quad \eta_{u'}^{(t)} = \sum_{u \in \mathcal{U}} p(u, w) \frac{s_{u,u'}}{z_u^{(t)}}. \tag{8}$$

The update rule for the cluster weights can now be written as:

$$q_{u'}^{(t+1)} = \eta_{u'}^{(t)} q_{u'}^{(t)}. \tag{9}$$

We can see from Eq. (9) that the update rule for clusters depends on the probability $p(u, w)$ and the probability depends on the learned w (see Eq. (6)). As in the first iterations the learned w is not accurate yet, we observe that in these conditions the clustering term can be detrimental to our learning. Thus we assign a small weight to γ in the first iteration and gradually allow it to grow to its defined value, similarly to deterministic annealing approaches [9].

The clustering term $\mathcal{L}_c$ requires the computation of pairwise distances between all the windows in the positive images. For efficiency we pre-compute the distances once at the beginning of the training. In order to speed up the algorithm, we also rank the values of $s_{u,u'}$ and keep only the largest 1000 values. We use the approximate nearest neighbor algorithm in [21]. We observe that this approximation has negligible effect in our final results.

K-means clustering To evaluate the importance of the use of convex clustering in our method we also introduce a clustering term based on the standard k-means method [19] which is non-convex. In that case the clustering loss defined in Eq. 7 becomes:

$$\mathcal{L}_c = \sum_{u \in \mathcal{U}} p(u, w) \Big(\min_{c \in \mathcal{C}} \|\phi(u) - c\|_2^2 \Big), \qquad (10)$$

where $c \in \mathcal{C}$ are the cluster centers. As in standard k-means, in this case the optimization of the loss is performed in two step: (i) compute the sample-cluster assignments, and (ii) re-compute the cluster centers c as the weighted mean of the samples $p(u, w)\phi(u)$ belonging to each cluster.

In the same way as we have modified the original convex clustering to account for the discriminativity of the windows, we multiply each square distance to a cluster with the probability term $p(u, w)$. In this case, as the clustering is not convex, to avoid to get stuck in poor local minima, at each iteration of the full loss defined in Eq. 5, we re-start the k-means algorithm with a random initialization of the cluster centers. Notice that in contrast to Eq. 7, in this case we do not need the term q_u because now each cluster c is a latent variable by itself. We call this clustering weighted k-means algorithm. In the experimental results we compare this approach to the convex clustering quantitatively.

4. Experiments

4.1. Dataset and implementation details

We evaluate our method on the Pascal VOC 2007 dataset [7] which allows us to compare to previous work. For a fair comparison with the state-of-the-art on weakly supervised object detection methods, we only discard the *images* with the "difficult" flag and do not use any *instance level* annotation by following the standard practice in the classification task of the challenge [7].

We evaluate the localization performance of our detectors using two measures. First we assess CorLoc [5], *i.e*. the percentage of positive "training" (*trainval*) images in which a method correctly localizes an object of the target class with more than 50% intersection-over-union ratio. Second, we follow the standard VOC procedure [7] and report average precision (AP) on the Pascal VOC 2007 *test* split. We use both *train* and *val* splits to train our final detectors. Note that for simplicity we do not double the amount of training data by adding horizontally flipped training images that can lead to a possible additional improvement in our results.

Our training involves tuning four parameters, the regularization parameter λ, the weight of the clustering term γ and two temperature parameters β and α of the soft-max in Eq.(4) and of the convex clustering in Eq.(7) respectively. We tune these parameters based on the classification accuracy in the validation set. We do not tune these parameters for each class separately but use a single value ($\alpha = 100$ and $\beta = 1$) for all classes. These values result in a sparse selection of clusters, roughly 20% of windows from positive images. For the k-means clustering baseline, we use 1000 cluster centers based on a cross-validation of the classification scores. We initialize these centers with the most discriminative object windows based on the learned classifier w after the first learning iteration and jointly learn them with the classifier parameters in the following iterations.

We stress that the proposed method does not lead to any additional inference time. Our method has the same computational complexity as the LSVM and SLSVM method which involves the computation of a dot product between the learned linear model w and a feature vector for each selective search window [29]. We represent each selective search window region with a 4096 dimensional fc7 ReLU layer output of the CNN model that is provided by Donahue *et al*. [6]. We also encode aspect ratio (8 bins), relative size (8 bins) and relative center position (2×8 bins) for each window. The use of the additional features leads to a similar improvement (0.4% mAP) in LSVM, SLSVM and our method. Finally, the average training times for LSVM, SLSVM and our method are approximately 1, 1 and 2 hours respectively on a 16 core i-7 CPU, after the CNN features and the pairwise distances are pre-computed.

4.2. Convex Clustering

In this part, we use our implementation of latent SVM (LSVM) (see Eq. (3)), soft-max latent SVM (SLSVM) (see (4)) as our baselines and compare them to our method. We also evaluate a variation of our algorithm (Ours (k-means)) where the clustering is performed with the weighted k-means algorithm (see 10) instead of our convex formulation. We present the performance of the methods in Figure 2 for the 20 VOC 2007 classes in terms of average precision (AP). First we compare the LSVM to its soft

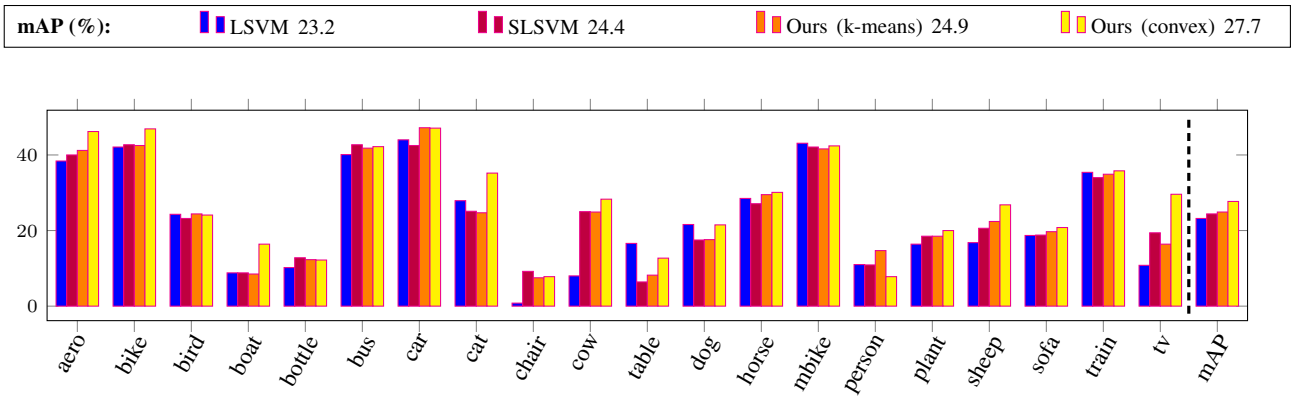

Figure 2. A comparison of our method with k-means (Ours (k-means)) and convex (Ours (convex)) clustering to the baselines LSVM, soft-max LSVM (SLSVM) and SLSVM in terms of average precision (AP). Our method with convex clustering significantly outperforms the baselines for most of the categories. Best viewed in color.

version and see that smoothing the hard-max formulation leads to an improvement of 1.2 points. As expected, this improvement is more prominent in categories which often have multiple instances in a single image, such as "chair", "cow", "sheep" and "tv-monitor" because there, in contrast to LSVM, SLSVM can exploit the presence of multiple objects. It should be noted that we use a single temperature parameter β for all categories and tuning this parameter in a category specific way can further improve both SLSVM and our method. While adding the weighted k-means clustering (Ours (k-means)) improves 0.5 points over the baseline SLSVM, the convex clustering formulation (Ours (convex)) achieves a significant improvement in most of the classes and 3.3% in mAP over SLSVM. The performance gap between the two clustering formulations shows the importance of a smooth and convex formulation for clustering. We also observe that our method fails to learn discriminative clusters and improve the baseline in classes with relatively low detection rates ($\sim$ 10% AP) such as "bottle", "chair" and "person" that often appear in cluttered indoor scenes.

In Figure 3 we illustrate the refinement in the localization of object instances during the iterative learning. The first column depicts the final detections of SLSVM and our method in purple and yellow respectively. The second column shows a response map of SLSVM, while third, fourth and fifth columns depict response maps of our method during different iterations. In the first two samples of "aeroplane" and "car", SLSVM detections contain some related background "sky" and "road" respectively. Our method progressively eliminates the background with the help of similarity with the found clusters. The third and fourth examples depict cases when parts of an object are more discriminative than the whole object and therefore the clustering term iteratively recovers the whole object. In the last example, we show a case where our method fails to localize the object accurately. The detection contains the bicycle and also the person riding the bicycle. Since "bicycle" and "person" classes co-occur in many training images, we obtain clusters that contain both classes.

We also illustrate some of the found clusters during training of different detectors in Figure 4. We see that the clusters contain objects and object parts with only a small portion of background and that they capture variations in appearance, pose and background.

4.3. Comparison to the state-of-the-art

In this part, we compare the results of our method to the state-of-the-art in WSL object detection. The results in Table 1 show that our method is comparable to the state-of-the-art in CorLoc. While our method outperforms the previous work of [26, 25, 4], it is worse than [30] on average. This method [30] focuses on obtaining a compact cluster for a foreground (object) class and multiple clusters for the related background (*e.g.* "sky" around "aeroplane"). It learns different appearance models for each cluster, whereas we focus on modeling the intra-class variance in the foreground via the found clusters. Wang *et al.* [30] apply an initial clustering on the windows of positive images and the method depends on the fact that the clustering can find a single compact cluster for the foreground. Therefore the performance of the method is sensitive to the number of clusters and requires tuning of this parameter for each class. In contrast, our method automatically learns the number of clusters and therefore uses a single parameter set for all classes. Moreover, [30] relies on multiple appearance models and on a expensive super-vector encoding of the CNN features that significantly increases the dimensionality of the feature vector, whereas our method uses a single appearance model and does not bring any additional computational load during inference compared to the standard LSVM. We could include more feature tuning as well as more complex features *etc.* to our model as well but that would clutter the experiments

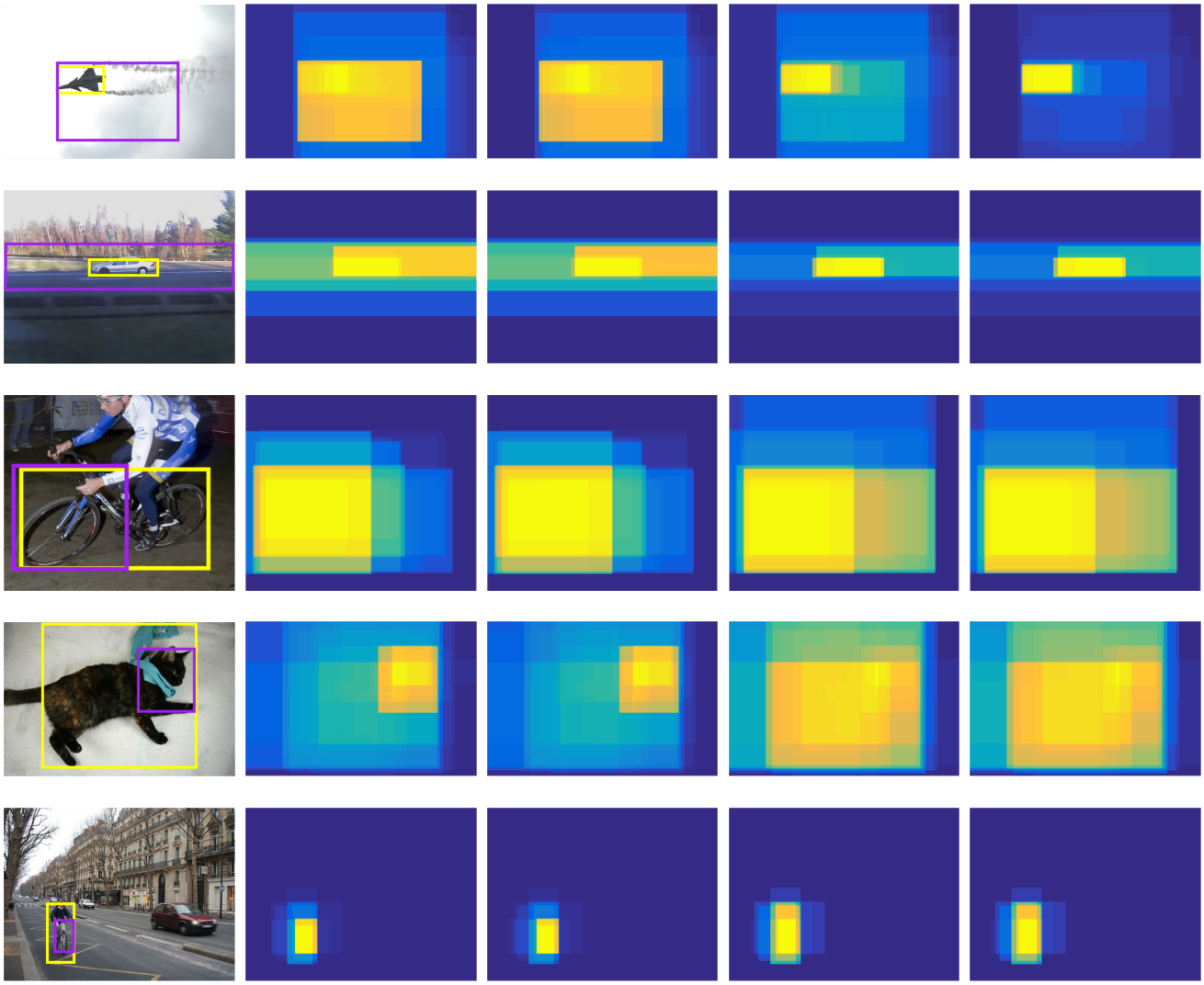

Figure 3. Examples of detection for SLSVM and our method. We show success cases of our method in the first four rows and a failure case in the last row. The first column shows the detection results of SLSVM (in purple) and our method (in yellow). The second column shows the response maps, *i.e*. weighted sum of window scores, of SLSVM and the third, fourth and fifth columns show the response maps of our method at various iterations. Best viewed in color.

and make the comparison to our baselines harder and less effective in demonstrating the effect of our main contribution.

We also compare our detection results to the state-of-the-art in terms of AP in Table 2. While Cinbis *et al*. [4] use Fisher Vectors [23] to represent the candidate object windows, the other methods in the table rely, as us, on powerful CNN features. Cinbis *et al*. [4] propose a method that uses a multi-fold splitting of positive images to alleviate the overfitting. Since we build our approach on a smoother learning framework, SLSVM and we also enforce similarity between objects and clusters, our method is less prone to overfitting and outperforms this method. Similarly to our work, Song *et al*. [27, 28] build their method on a different smoothed Latent SVM algorithm and use efficient clustering algorithms via sub-modular optimization. While Song *et al*. [27, 28] use clustering to initialize the latent parameters (*i.e*. object windows and part configurations), our method jointly learns to cluster and to detect object instances in a discriminative way and thus outperforms these methods significantly. Bilen *et al*. [2] employ a posterior regularization technique that enforces symmetry and mutual exclusion on window selection. While our method outperforms this work [2], the same regularization technique can be added to our learning and improve the detection performance further.

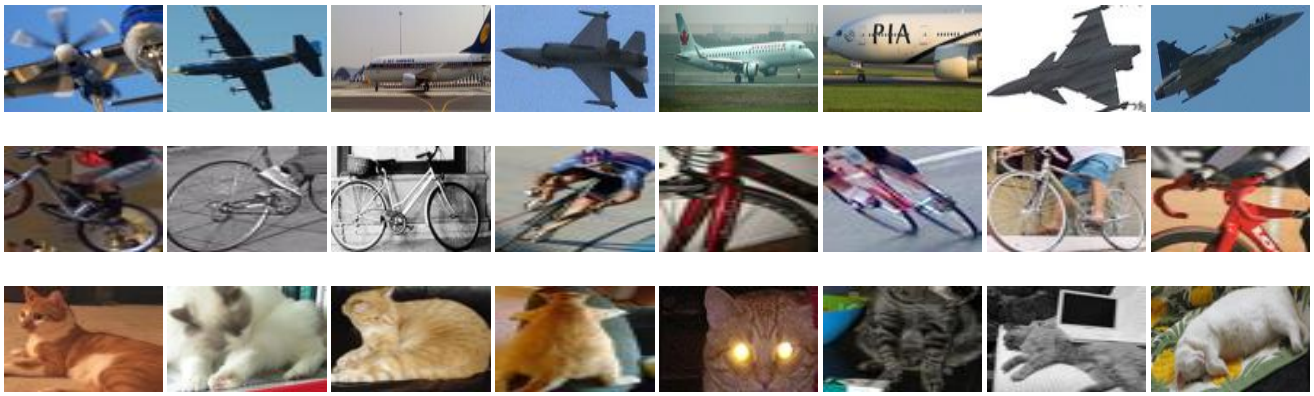

Figure 4. Cluster examples that are found during the training of different detectors. They contain objects, object parts with small portion of background and show significant variations in appearance, pose and background. Best viewed in color.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hors	mbik	pers	plnt	shp	sofa	train	tv	mean
Our method	66.4	59.3	42.7	20.4	**21.3**	**63.4**	**74.3**	59.6	21.1	**58.2**	14.0	38.5	49.5	60.0	19.8	**39.2**	41.7	30.1	50.2	44.1	43.7
Shi *et al*. [26]	54.7	22.7	33.7	**24.5**	4.6	33.9	42.5	57.0	7.3	39.1	**24.1**	43.3	41.3	51.5	25.3	13.3	28.0	29.5	54.6	11.8	32.1
Shi *et al*. [25]	67.3	54.4	34.3	17.8	1.3	46.6	60.7	**68.9**	2.5	32.4	16.2	**58.9**	51.5	64.6	18.2	3.1	20.9	34.7	**63.4**	5.9	36.2
Cinbis *et al*. [4]	56.6	58.3	28.4	20.7	6.8	54.9	69.1	20.8	9.2	50.5	10.2	29.0	58.0	64.9	**36.7**	18.7	56.5	13.2	54.9	59.4	38.8
Wang *et al*. [30]	**80.1**	**63.9**	**51.5**	14.9	21.0	55.7	74.2	43.5	**26.2**	53.4	16.3	56.7	**58.3**	**69.5**	14.1	38.3	**58.8**	**47.2**	49.1	**60.9**	**48.5**

Table 1. Comparison of WSL object detectors on PASCAL VOC 2007 in terms of correct localization (CorLoc [5]) on positive training images.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hors	mbik	pers	plnt	shp	sofa	train	tv	mean
Our method	46.2	46.9	**24.1**	**16.4**	12.2	42.2	**47.1**	35.2	7.8	28.3	12.7	21.5	30.1	42.4	7.8	20.0	26.8	20.8	35.8	29.6	27.7
Cinbis *et al*. [4]	35.8	40.6	8.1	7.6	3.1	35.9	41.8	16.8	1.4	23.0	4.9	14.1	31.9	41.9	**19.3**	11.1	**27.6**	12.1	31.0	40.6	22.4
Song *et al*. [27]	27.6	41.9	19.7	9.1	10.4	35.8	39.1	33.6	0.6	20.9	10.0	27.7	29.4	39.2	9.1	19.3	20.5	17.1	35.6	7.1	22.7
Song *et al*. [28]	36.3	**47.6**	23.3	12.3	11.1	36.0	46.6	25.4	0.7	23.5	12.5	23.5	27.9	40.9	14.8	19.2	24.2	17.1	**37.7**	11.6	24.6
Bilen *et al*. [2]	42.2	43.9	23.1	9.2	**12.5**	**44.9**	45.1	24.9	8.3	24.0	13.9	18.6	31.6	43.6	7.6	**20.9**	26.6	20.6	35.9	29.6	26.4
Wang *et al*. [30]	**48.8**	41.0	23.6	12.1	11.1	42.7	40.9	**35.5**	**11.1**	**36.6**	**18.4**	**35.3**	**34.8**	**51.3**	17.2	17.4	26.8	**32.8**	35.1	**45.6**	**30.9**

Table 2. Comparison of WSL object detectors on PASCAL VOC 2007 in terms of AP in the test set [7].

5. Conclusion

We have presented a weakly supervised detection algorithm that encourages similarity between objects to avoid overfitting and local minima solutions in the learning. Our formulation allows a joint learning of detection and clustering in an efficient and principled way. We show that using similarity is beneficial, improves the detection performances over the baseline and gives comparable results with the state-of-the-art.

Acknowledgments: We thank Dr. Vinay Namboodiri for helpful discussions. The work is supported by EU FP7 project AXES, ERC starting grant COGNIMUND, IWT SBO project PARIS.

References

[1] H. Bilen, M. Pedersoli, V. Namboodiri, T. Tuytelaars, and L. Van Gool. Object classification with adaptable regions. *CVPR*, 2014.

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

[3] D. M. Bradley and J. A. Bagnell. Convex coding. In *Conference on Uncertainty in Artificial Intelligence*, pages 83–90. AUAI Press, 2009.

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

[5] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *IJCV*, 100(3):275–293, 2012.

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. 2014.

[7] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[9] P. V. Gehler and O. Chapelle. Deterministic annealing for multiple-instance learning. In *International conference on artificial intelligence and statistics*, pages 123–130, 2007.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587. IEEE, 2014.

[11] T. Hastie, R. Tibshirani, and J. J. H. Friedman. *The elements of statistical learning*, volume 1. 2001.

[12] M. Hoai and A. Zisserman. Discriminative sub-categorization. In *CVPR*, 2013.

[13] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, pages 1943–1950. IEEE, 2010.

[14] A. Joulin and F. R. Bach. A convex relaxation for weakly supervised classifiers. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1279–1286, 2012.

[15] N. Komodakis, N. Paragios, and G. Tziritas. Clustering via lp-based stabilities. In *Advances in Neural Information Processing Systems*, pages 865–872, 2009.

[16] M. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, pages 1189–1197, 2010.

[17] D. Lashkari and P. Golland. Convex clustering with exemplar-based models. In *NIPS*, pages 825–832, 2007.

[18] D. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[19] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[20] K. Miller, M. P. Kumar, B. Packer, D. Goodman, and D. Koller. Max-margin min-entropy models. In *AISTATS*, 2012.

[21] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014.

[22] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[23] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010.

[24] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *CVPR*, volume 1, pages 993–1000. IEEE, 2006.

[25] Z. Shi, T. M. Hospedales, and T. Xiang. Bayesian joint topic modelling for weakly supervised object localisation. In *ICCV*, pages 2984–2991. IEEE, 2013.

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

[27] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell. On learning to localize objects with minimal supervision. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1611–1619, 2014.

[28] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell. Weakly-supervised discovery of visual pattern configurations. In *Advances in Neural Information Processing Systems*, pages 1637–1645, 2014.

[29] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[30] C. Wang, W. Ren, K. Huang, and T. Tan. Weakly supervised object localization with latent category learning. In *ECCV 2014*, volume 8694, pages 431–445, 2014.

[31] C. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, pages 1169–1176, 2009.