Unsupervised Image Matching and Object Discovery as Optimization

Huy V. Vo[1,2,3], Francis Bach[1,2], Minsu Cho[4], Kai Han[5], Yann LeCun[6], Patrick Pérez[3] and Jean Ponce[1,2]

[1]Département d'informatique de l'ENS, ENS, CNRS, PSL University, Paris, France
[2]INRIA, Paris, France [3]Valeo.ai [4]POSTECH [5]University of Oxford [6]New York University

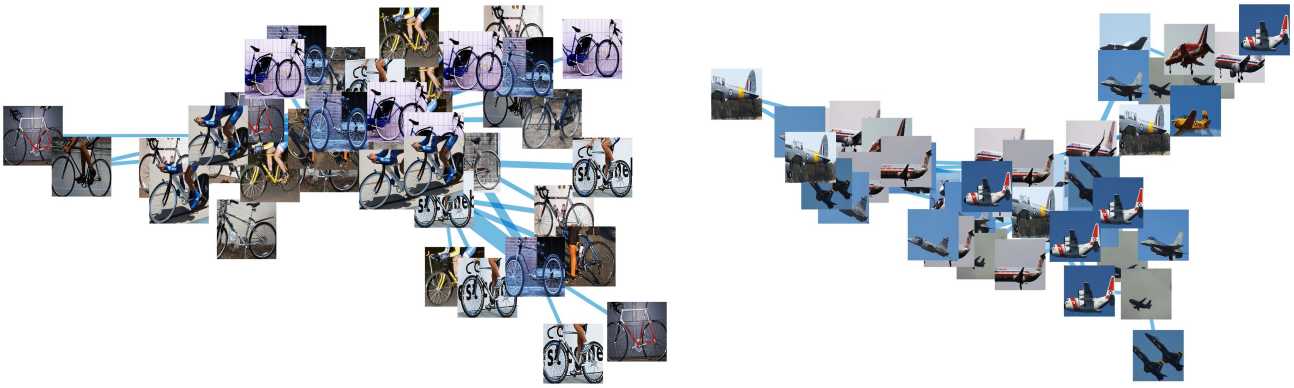

Figure 1: The proposed optimization-based method automatically discovers links between images that depict similar objects. This figure shows two image clusters that emerge as a by-product of this approach on the VOC_6x2 object recognition dataset that mixes 6 classes under two viewpoints. See text for details.

Abstract

Learning with complete or partial supervision is powerful but relies on ever-growing human annotation efforts. As a way to mitigate this serious problem, as well as to serve specific applications, unsupervised learning has emerged as an important field of research. In computer vision, unsupervised learning comes in various guises. We focus here on the unsupervised discovery and matching of object categories among images in a collection, following the work of Cho et al. *[12]. We show that the original approach can be reformulated and solved as a proper optimization problem. Experiments on several benchmarks establish the merit of our approach.*

1. Introduction

Remarkable progress has been achieved in visual tasks such as image categorization, object detection, or semantic segmentation, typically using fully supervised algorithms and vast amount of manually annotated data (e.g., [17, 20, 21, 27, 29, 38, 40]). With the advent of crowd-sourcing, large corporations and, to a lesser extent, academic units can launch the corresponding massive annotation efforts for *specific* projects that may involve millions images [40].

But handling Internet-scale image (or video) repositories or the continuous learning scenarios associated with personal assistants or autonomous cars demands approaches less hungry for manual annotation. Several alternatives are possible, including *weakly supervised* approaches that rely on readily available meta-data [2, 9] or image-level labels [14, 23, 24, 25, 39, 45] instead of more complex annotations such as bounding boxes [17, 38] or object masks [20] as supervisory signal; *semi supervised* methods [6, 26] that exploit a relatively small number of fully annotated pictures, together with a larger set of unlabelled images; and *self supervised* algorithms that take advantage of the internal regularities of image parts [15, 37] or video subsequences [1, 34, 48] to construct image models that can be further fine-tuned in fully supervised settings.

We address here the even more challenging problem of discovering both the structure of image collections – that is, which images depict similar objects (or textures, scenes, actions, etc.), and the objects in question, in a *fully unsupervised* setting [8, 11, 16, 30, 39, 41, 43]. Although weakly, semi, and self supervised methods may provide a

more *practical* foundation for large-scale visual recognition, the fully unsupervised construction of image models is a *fundamental* scientific problem in computer vision, and it should be studied. In addition, any reasonable solution to this problem will facilitate subsequent human labelling (by presenting discovered groups to the operator) and scaling through automatic label propagation, help interactive query-based visual search by linking ahead of time fragments of potential interest, and provide a way to learn visual models for subsequent recognition.

1.1. The implicit structure of image collections

Any collection of images, say, those found on the Internet, or more modestly, in a dataset such as Pascal VOC'07, admits a natural graph representation, where nodes are the pictures themselves, and edges link pairs of images with similar visual content. In *supervised* image categorization (e.g., [27, 29]) or object detection (e.g., [17, 20, 38]) tasks, both the graph structure and the visual content are clearly defined: Annotators typically sort the images into bags, each one intended to represent some "object", "scene" or, say, "action" class ("horse", "forest", "playing tennis", etc.). Two nodes are linked by an edge when they are associated with the same bag, and each class is empirically defined by the images (or some manually-defined rectangular regions within) in the corresponding connected component of the graph. In *weakly supervised* cosegmentation [23, 25, 39] or colocalization [14, 24, 45] tasks, on the other hand, the graph is fully connected, and all images are supposed to contain instances of the (few) same object categories, say, "horse", "grass", "sky", "background". Manual intervention is reduced to selecting which images to put into a single bag, and the visual content, in the form of regions defined by pixel-level symbolic labels or bounding boxes associated with one of the predefined categories, is discovered using a clustering algorithm. [1]

We address in this paper the much more difficult problem of *fully unsupervised* image matching and object discovery, where both the graph structure and a model of visual content in the form of object bounding boxes must be extracted from the native data without *any* manual intervention. This problem has been addressed in various forms, e.g., clustering [16][2], image matching [39] or topic discovery [41, 43] (see also [8, 11], where "pseudo-object" labels are learned in an unsupervised manner). In this presentation, we build directly on the work of Cho *et al.* [12] (see [28] for related work): Given an image and its neighbors, assumed to contain the same object, a robust matching technique exploits both appearance and geometric consistency constraints to assign confidence and saliency ("stand-out") scores to region proposals in this image. The overall discovery algorithm alternates between *localization* steps where the neighbors are fixed and the regions with top saliency scores are selected as potential objects, and *retrieval* steps where the confidence of the regions within potential objects are used to find the nearest neighbors of each image. After a fixed number of steps, the region with top saliency in each image is declared to be the object it contains. Empirically, this method has been shown in [12] to give good results. However, it does not formulate image matching and object discovery as a proper optimization problem, and there is no guarantee that successive iterations will improve some objective measure of performance. The aim of this paper is to remedy this situation.

[1] In both the cases of supervised image categorization/object detection and weakly supervised cosegmentation/colocalization, once the graph structure and the visual content have been identified at *training time*, these can be used to learn a model of the different object classes and add nodes, edges, and possibly additional bounding boxes at *test time*.

[2] Note that plain unsupervised clustering, whether classic, spectral, discriminative or deep [4, 22, 32, 36], focuses on data partitioning and not on the discovery of subsets of matching items within a cluttered collection.

2. Proposed approach

2.1. Problem statement

Let us consider a set of n images, each containing p_i rectangular region proposals, with i in $\{1 \dots n\}$. We assume that the images are equipped with some implicit graph structure, where there is a link between two images when the second image contains at least one object from a category depicted in the first one, and our aim is to discover this structure, that is, find the links and the corresponding objects. To model this problem, let us define an indicator variable x_i^k, whose value is 1 when region number k of image i corresponds to a "foreground object" (visible in large part and from a category that occurs multiple times in the image collection), and 0 otherwise. We collect all the variables x_i^k associated with image i into an element x_i of $\{0,1\}^{p_i}$, and concatenate all the variables x_i into an element x of $\{0,1\}^{\sum_{i=1}^n p_i}$. Likewise, let us define an indicator variable e_{ij}, whose value is 1 if image j contains an object also occurring in image i, with $1 \leq i,j \leq n$ and $j \neq i$, and 0 otherwise, collect all the variables e_{ij} associated with image i into an element e_i of $\{0,1\}^n$, and concatenate all the variables e_i into an $n \times n$ matrix e with rows e_i^T. Note that we can use e to define a neighborhood for each image in the set: Image j is a neighbor of the image i if $e_{ij} = 1$. By definition, e defines an undirected graph if e is symmetric and a directed one otherwise. Let us also denote by S_{ij}^{kl} the similarity between regions k and l of images i and j, and by S_{ij} the $p_i \times p_j$ matrix with entries S_{ij}^{kl}.

We propose to maximize with respect to x and e the objective function

$$S(x,e) = \sum_{\substack{i,j=1 \\ j \neq i}}^{n} e_{ij} \sum_{\substack{1 \leq k \leq p_i \\ 1 \leq l \leq p_j}} S_{ij}^{kl} x_i^k x_j^l = \sum_{\substack{i,j=1 \\ j \neq i}}^{n} x_i^T [e_{ij} S_{ij}] x_j. \quad (1)$$

Intuitively maximizing $S(x,e)$ encourages building edges between images i and j that contain regions k and l with a strong similarity S_{ij}^{kl}. Of course we would like to impose certain constraints on the x and e variables. The following cardinality constraints are rather natural:

• An image should not contain more than a prededined number of objects, say ν,

$$\forall\ i \in 1 \dots n,\ \ x_i \cdot \mathbb{1}_{p_i} \leq \nu, \tag{2}$$

where $\mathbb{1}_{p_i}$ is the element of $\mathbb{R}^{p_i}$ with all entries equal to one.

• An image should not match more than a predefined number of other images, say τ,

$$\forall\ i \in 1 \dots n,\ \ e_i \cdot \mathbb{1}_n \leq \tau. \tag{3}$$

Assumptions. We will suppose from now on that S_{ij} is elementwise nonnegative, but not necessarily symmetric (the similarity model we explore in Section 3 is asymmetrical). Likewise, we will assume that the matrix e has a zero diagonal but is not necessarily symmetric.

Under these assumptions, the cubic pseudo-Boolean function S is supermodular [10]. Without constraints, this type of functions can be maximized in polynomial time using a max-flow algorithm [7] (in the case of $S(x,e)$, which does not involve linear and quadratic terms, the solution is of course trivial without constraints, and amounts to setting all x_i^k and e_{ij} with $i \neq j$ to 1). When the cardinality constraints (2-3) are added, this is not the case anymore, and we have to resort to a gradient ascent algorithm as explained next.

2.2. Relaxing the problem

Let us first note that, for binary variables x_i^k, x_j^l and e_{ij}, we have

$$S(x,e) = \sum_{\substack{i,j=1 \\ j\neq i}}^{n} \sum_{\substack{1\leq k\leq p_i \\ 1\leq l\leq p_j}} S_{ij}^{kl} \min(e_{ij}, x_i^k, x_j^l), \tag{4}$$

with $S_{ij}^{kl} \geq 0$. Relaxing our problem so all variables are allowed to take values in $[0,1]$, our objective becomes a sum of concave functions, and thus is itself a concave function, defined over the convex set (hyperrectangle) $[0,1]^N$, where N is the total number of variables. This is the standard tight concave continuous relaxation of supermodular functions.

The Lagrangian associated with our relaxed problem is

$$K(x,e;\lambda,\mu) = S(x,e) - \sum_{i=1}^{n} [\lambda_i(x_i \cdot \mathbb{1}_{p_i} - \nu) + \mu_i(e_i \cdot \mathbb{1}_n - \tau)], \tag{5}$$

where $\lambda = (\lambda_1, \dots, \lambda_n)^T$ and $\mu = (\mu_1, \dots, \mu_n)^T$ are *positive* Lagrange multipliers. The function $S(x,e)$ is concave and the primal problem is strictly feasible; hence Slater's conditions [44] hold, and we have the following equivalent primal and dual versions of our problem

$$\begin{cases} \max_{(x,e)\in D} \inf_{\lambda,\mu\geq 0} K(x,e;\lambda,\mu), \\ \min_{\lambda,\mu\geq 0} \sup_{(x,e)\in D} K(x,e;\lambda,\mu), \end{cases} \tag{6}$$

where the domain D is the Cartesian product of $[0,1]^{\sum_i p_i}$ and the space of $n \times n$ matrices with entries in $[0,1]$ and a zero diagonal. With slight abuse we denote it $D = [0,1]^N$, with $N = \sum_i p_i + n(n-1)$.

2.3. Solving the dual problem

We propose to solve the dual problem with a subgradient descent approach. Starting from some initial values for λ^0 and μ^0, we use the update rule

$$\begin{cases} \lambda_i^{t+1} = [\lambda_i^t + \alpha(x_i^t \cdot \mathbb{1}_{p_i} - \nu)]_+, \\ \mu_i^{t+1} = [\mu_i^t + \beta(e_i^t \cdot \mathbb{1}_n - \tau)]_+, \end{cases} \tag{7}$$

where $[\cdot]_+$ denotes positive part, $k \geq 0$, α and β are fixed step sizes, $x_i^t \cdot \mathbb{1}_{p_i} - \nu$ and $e_i^t \cdot \mathbb{1}_n - \tau$ are respectively the negative of the subgradients of the Lagrangian with respect to λ_i and μ_i in λ_i^t and μ_i^t, and

$$(x^t, e^t) \in \operatorname{argmax}_{(x,e)\in[0,1]^N} K(x,e;\lambda^t,\mu^t). \tag{8}$$

As shown in Appendix, for fixed values of λ and μ, our Lagrangian is a *supermodular* pseudo-Boolean function of binary variables sets x and e. This allows us to take advantage of the following direct corollary of [3, Prop. 3.7].

Proposition 2.1. *Let f denote some supermodular pseudo-Boolean function of n variables. We have*

$$\max_{x\in\{0,1\}^n} f(x) = \max_{x\in[0,1]^n} f(x), \tag{9}$$

and the set of maximizers of $f(x)$ in $[0,1]^n$ is the convex hull of the set of maximizers of f on $\{0,1\}^n$.

In particular, we can take

$$(x^t, e^t) \in \operatorname{argmax}_{(x,e)\in\{0,1\}^N} K(x,e;\lambda^t,\mu^t). \tag{10}$$

As shown in [7, 10], the corresponding supermodular cubic pseudo-Boolean function optimization problem is equivalent to a maximum stable set problem in a bipartite *conflict graph*, which can itself be reduced to a maximum-flow problem. See Appendix for details.

Note that the size of the min-cut/max-flow problems that have to be solved is conditioned by the number of nonzero S_{ij}^{kl} entries, which is upper-bounded by n^2p^2 when the matrices S_{ij} are dense (denoting $p = \max\{p_i\}$). This is prohibitively high given that, in practice, p is between 1000 and 4000. To make the computations manageable, we set all but between 100 and 1000 (depending on the dataset's size) of the largest entries in S_{ij} to zero in our implementation.

2.4. Solving the primal problem

Once the dual problem is solved, as argued by Nedić & Ozdaglar [35] and Bach [3], an approximate solution of the primal problem can be found as a running average of the primal sequence (x^t, e^t) generated as a by-product of the sub-gradient method:

$$\hat{x} = \frac{1}{T}\sum_{t=0}^{T-1} x^t, \quad \hat{e} = \frac{1}{T}\sum_{t=0}^{T-1} e^t \tag{11}$$

after some number T of iterations. Note the scalars $\hat{x}_i^k$ and $\hat{e}_{ij}$ lie in $[0,1]$ but do not necessarily verify the constraints (2) and (3). Theoretical guarantees on these values can be found under additional assumptions in [3, 35].

2.5. Rounding the solution and greedy ascent

Note that two problems remain to be solved: The solution $(\hat{x}, \hat{e})$ found now belongs to $[0,1]^N$ instead of $\{0,1\}^N$, and it may not satisfy the original constraints. Note, however, that because of the form of the function S, given some i in $\{1,\dots,n\}$ and fixed values for e and all x_j with $j \neq i$, the maximum value of S given the constraints is obtained by setting to 1 exactly the ν entries of x_i corresponding to the ν largest entries of the vector $\sum_{j\neq i}(e_{ij}S_{ij} + e_{ji}S_{ji}^T)x_j$. Likewise, for some fixed value of x, the maximum value of S is reached by setting to 1, for all i in $\{1,\dots,n\}$, exactly the τ entries of e_i corresponding to the τ largest scalars $x_i^T S_{ij} x_j$ for $j \neq i$ in $\{1\dots n\}$. This suggests the following approach to rounding up the solution, where the variables x_i are updated sequentially in an order specified by some random permutation σ of $\{1,\dots,n\}$, before the variables e_i are updated in parallel. Given the permutation σ, the algorithm below turns the running average $(\hat{x}, \hat{e})$ of the primal sequence into a discrete solution (x, e) that satisfies the conditions (2) and (3):

Initialize $x = \hat{x}$, $e = \hat{e}$.
For $i = 1$ to n do
  Compute the indices k_1 to k_ν of the ν largest elements of the vector $\sum_{j\neq\sigma(i)}^{n}(e_{\sigma(i)j}S_{\sigma(i)j} + e_{j\sigma(i)}S_{j\sigma(i)}^T)x_j$.
  $x_{\sigma(i)} \leftarrow 0$.
  For $t = 1$ to ν do $x_{\sigma(i)}^{k_t} \leftarrow 1$.
For $i = 1$ to n do
  Compute the indices j_1 to j_τ of the τ largest scalars $x_i^T S_{ij} x_j$.
  $e_i \leftarrow 0$.
  For $t = 1$ to τ do $e_{ij_t} \leftarrow 1$.
Return x, e.

Note that there is no preferred order for the image indices. This actually suggests repeating this procedure with different random permutations until the variables x and e do not change anymore or some limit on the number of iterations is reached. This iterative procedure can be seen as a greedy ascent procedure over the discrete variables of interest. Note that by construction the terms in the left and right sides of (2) and (3) are equal at the optimum.

2.6. Ensemble post processing

The parameter ν can be seen from two different viewpoints: (1) as the maximum number of objects that may be depicted in an image, or (2) as an upper bound on the total number of object region *candidates* that are under consideration in a picture. Both viewpoints are equally valid but, following Cho *et al.* [12], we focus in the rest of this presentation on the second one, and present in this section a simple heuristic for selecting one final object region among these candidates. Concretely, since using random permutations during greedy ascent provides a different solution for each run of our method, we propose to apply an *ensemble method* to stabilize the results and boost performance in this selection process, itself viewed as a post-processing stage separate from the optimization part.

Let us suppose that after L independent executions of the greedy ascent step, we obtain L solutions $(x(l), e(l))$, $1 \leq l \leq L$. We start by combining these solutions into a single discrete pair $(\bar{x}, \bar{e})$ where $\bar{x}$ and $\bar{e}$ satisfy

- $\bar{x}_i^k = 1$ if $\exists\, l, 1 \leq l \leq L$ such that $x_i^k(l) = 1$,
- $\bar{e}_{ij} = 1$ if $\exists\, l, 1 \leq l \leq L$ such that $e_{ij}(l) = 1$.

This way of combining the individual solutions can be seen as a *max pooling* procedure. We have also tried average pooling but found it less effective. Note that after this intermediate step, an image might violate any of the two constraints (2-3). This is not a problem in this postprocessing stage of our method. Indeed, we next show how to use $\bar{x}$ and $\bar{e}$ to select a *single* object proposal for each image.

We choose a single proposal for each image out of those retained in $\bar{x}$ (proposals (i,k) s.t. $\bar{x}_i^k = 1$). To this end, we rank the proposals in image i according to a score u_i^k defined for each proposal (i,k) as

$$u_i^k = \bar{x}_i^k \sum_{j\in\mathcal{N}(i,k)} \max_{l|\bar{x}_j^l=1} S_{ij}^{kl}, \tag{12}$$

where $\mathcal{N}(i,k)$ is composed of the τ images represented by the $1s$ in $\bar{e}_i$ which have the largest similarity to (i,k) as measured by $\max_{l|\bar{x}_j^l=1} S_{ij}^{kl}$. Finally, we choose the proposal in image i with maximum score u_i^k as the final object region. Note that the graph of images corresponding to these final object regions can be retrieved by computing e that maximizes the objective function given the value of x defined by these regions as in the greedy ascent. Also, the method above can be generalized to more than one proposal per image using the defined ranking.

3. Similarity model

Let us now get back to the definition of the similarity function S_{ij}. As advocated by Cho *et al.* [12], a rectangular region which is a tight fit for a compact object (the *foreground*) should better model this object than a larger region, since it contains less *background*, or than a smaller region (a *part*) since it contains more foreground. Cho *et al.* [12] only implement the first constraint, in the form of a *stand-out* score. We discuss in this section how to implement these ideas in the optimization context of this work.

3.1. Similarity score

Following [12], the similarity score between proposal k of image i and proposal l of image j can be defined as

$$s_{ij}^{kl} = a_{ij}^{kl} \sum_{o \in O} g(r_i^k, r_j^l, o) \sum_{\substack{1 \le k' \le p_i \\ 1 \le l' \le p_j}} g(r_i^{k'}, r_j^{l'}, o) a_{ij}^{k'l'}, \quad (13)$$

where a_{ij}^{kl} is a similarity term based on appearance alone, using the WHO descriptor (whiten HOG) [13, 19] in our case, r_i^k and r_j^l denote the image rectangles associated with the two proposals, o is a discretized offset (translation plus two scale factors) taking values in O, and $g(r, s, o)$ measures the geometric compatibility between o and the rectangles r and s. Intuitively, s_{ij}^{kl} scales the appearance-only score a_{ij}^{kl} by a geometric-consistency term akin to a generalized Hough transform [5], see [12] for details.

Note that we can rewrite Eq. (13) as

$$s_{ij}^{kl} = b_{ij}^{kl} \cdot c_{ij}, \quad (14)$$

where b_{ij}^{kl} is the vector of dimension $|O|$ with entries $a_{ij}^{kl} g(r_i^k, r_j^l, o)$, and $c_{ij} = \sum_{k',l'=1}^{p} b_{ij}^{k'l'}$. The $p_i p_j$ vectors b_{ij}^{kl} and the vector c_{ij} can be precomputed with time and storage cost of $\mathcal{O}(p^2|O|)$. Each term s_{ij}^{kl} can then be computed in $\mathcal{O}(|O|)$ time, and the matrix S_{ij} can thus be computed with a total time and space complexity of $\mathcal{O}(p^2|O|)$.

Note that the score s_{ij}^{kl} defined by Eq. (13) depends on the number of region proposals per images, which may introduce a bias for edges between images that contain many region proposals. It may thus be desirable to *normalize* this score by defining it instead as

$$s_{ij}^{kl} = \frac{1}{p_i p_j} b_{ij}^{kl} \cdot c_{ij}. \quad (15)$$

3.2. Stand-out score

Let us identify the region proposals contained in some image i with their index k, and define P_i^k as the set of regions that are *parts* of that region (that is, they are included, with some tolerance, within k). Let us also define B_i^k as the set of regions that form the *background* for k (that is, k is included, with some tolerance, within these regions). Let r_i^k denote the actual rectangular image region associated with proposal k in image i, and let $A(r)$ denote the area of some rectangle r. A plausible definition for P_i^k is

$$P_i^k = \{l \,:\, A(r_i^k \cap r_i^l) > \rho A(r_i^l)\}, \quad (16)$$

for some reasonable value of ρ, e.g., 0.5. Likewise, a plausible definition for B_i^k is

$$B_i^k = \{l : A(r_i^k \cap r_i^l) > \delta A(r_i^k) \text{ and } A(r_i^l) > \gamma A(r_i^k)\}, \quad (17)$$

for reasonable values of δ and γ, e.g., 0.8 and 2. Following [12], we define the *stand-out score* of a match (k, l) as

$$S_{ij}^{kl} = s_{ij}^{kl} - v_{ij}^{kl}, \text{ where } v_{ij}^{kl} = \max_{(k',l') \in B_i^k \times B_j^l} s_{ij}^{k'l'}. \quad (18)$$

With this definition, S_{ij}^{kl} may be negative. In our implementation, we threshold these scores so they are nonnegative.

When B_i^k and B_j^l are large, which is generally the case when the regions r_i^k and r_j^l are small, a brute-force computation of v_{ij}^{kl} may be very slow. We propose below instead a simple heuristic that greatly speeds up calculations.

Let Q_{ij} denote the set formed by the q matches (k, l) with highest scores s_{ij}^{kl}, sorted in increasing order, which can be computed in $\mathcal{O}(p^2 \log p)$. The stand-out scores can be computed efficiently by the following procedure:

Initialize all v_{ij}^{kl} to 0.
For each match (k', l') in Q_{ij} do
  For each match (k, l) in $P_i^{k'} \times P_j^{l'}$ do $v_{ij}^{kl} = s_{ij}^{k'l'}$.
For $k = 1$ to p_i and $l = 1$ to p_j do
  If $s_{ij}^{kl} > 0$ and $v_{ij}^{kl} = 0$ then $v_{ij}^{kl} = \max_{(k',l') \in B_i^k \times B_j^l} s_{ij}^{k'l'}$.

The idea is that relatively few high-confidence matches (k', l') in Q_{ij} can be used to efficiently compute many stand-out scores. There is a trade-off between the cost of this step, $\mathcal{O}(\sum_{(k',l') \in Q_{ij}} |P_i^{k'}||P_j^{l'}|)$, and the number of variables v_{ij}^{kl} it assigns a value to, $\mathcal{O}(|\cup_{(k',l') \in Q_{ij}} P_i^{k'} \times P_j^{l'}|)$. In practice, we have found that taking $q = 10,000$ is a good compromise, with only about 5% of the stand-out scores being computed in a brute-force manner, and a significant speed-up factor of over 10.

4. Experiments and results

Datasets, proposals and metric. For our experiments we use the same datasets (*ObjectDiscovery* [OD], *VOC_6x2* and *VOC_all*) and region proposals (obtained by the *randomized Prim's algorithm* [RP] [33]) as Cho *et al.* [12]. OD consists of pictures of three object classes (*airplane*, *horse* and *car*) with outliers not containing any object instance. There are 100 images per category, with 18, 7 and 11 outliers respectively (containing no object instance). VOC_all

Method			OD	VOC_6x2
Cho *et al.*			84.2	67.7
Cho *et al.*, our version			84.2	67.6
w/o EM	w/o CO	w/o NS	81.9 ± 0.9	65.9 ± 1.0
		w NS	83.1 ± 0.8	67.2 ± 1.0
	w CO	w/o NS	82.9 ± 0.8	66.6 ± 0.7
		w NS	84.4 ± 0.8	68.1 ± 0.9
w EM	w/o CO	w/o NS	84.4 ± 0.0	68.8 ± 0.4
		w NS	85.6 ± 0.3	68.7 ± 0.5
	w CO	w/o NS	83.8 ± 0.2	67.4 ± 0.4
		w NS	**85.8 ± 0.6**	**69.4 ± 0.3**

Table 1: Performance of different configurations of our algorithm compared to the results of Cho *et al.* on Object Discovery and VOC_6x2 datasets in the separate setting.

is a subset of the PASCAL VOC2007 train+val dataset obtained by eliminating all images containing only objects marked as *difficult* or *truncated*. Finally, VOC_6x2 is a subset of VOC_all containing only images of 6 classes – *aeroplane*, *bicycle*, *boat*, *bus*, *horse* – and *motorbike* from two different views, *left* and *right*.

For evaluation, we use the standard *CorLoc* measure, the percentage of images correctly localized. It is a proxy metric in the case of unsupervised discovery. An image is "correctly localized" when the intersection over union (IoU) between one of the ground-truth regions and the predicted one is greater than 0.5. Following [12], we evaluate our algorithm in "separate" and "mixed" settings. In the former case, the class-wise performance is averaged over classes. In the latter, a single performance is computed over all classes jointly. In our experiments, we use $\nu = 5$, $\tau = 10$ and standout matrices with 1000 non-zero entries unless mentioned otherwise.

Separate setting. We firstly evaluate different settings of our algorithm on the two smaller datasets, OD and VOC_6x2. The performance is governed by three design choices: (1) using the normalized stand-out score (*NS)* or its unnormalized version, (2) using continuous optimization (*CO*) or variables x and e with all entries equal to one to initialize the greedy ascent procedure, and (3) using the ensemble method (*EM*) or not. In total, we thus have eight configurations to test.

The results are shown in Table 1. We have found a small bug in the publicly available code of Cho *et al*. [12] and report both the results from [12] and those we obtained after correction. We observe that the normalized standout score always gives comparable or better results than its unnormalized counterpart while the ensemble method also improves both the score and the stability (lower variance) of our solution. Combining the normalized standout score, the ensemble method, and the continuous optimization initialization to greedy ascent yields the best performance. Our best results outperform [12] by small but statistically significant margins: 1.6% for OD and 1.8% for VOC_6x2. Finally, to assess the merit of the continuous optimization, we have measured its duality gap on OD and VOC_6x2: it ranges from 1.5% to 8.7% of the energy, with an average of 5.2% and 3.9% on the two datasets respectively.

Method		VOC_all
Cho *et al.*		36.6
Cho *et al.*, our execution		37.6
w/o CO	w/o EM	36.4 ± 0.3
	w EM	39.0 ± 0.2
w CO	w/o EM	37.8 ± 0.3
	w EM	39.2 ± 0.2
Li *et al.* [31]		40.0
Wei *et al.* [49]		**46.9**

Table 2: Performance on VOC_all in separate setting with different configurations.

We now evaluate our algorithm on VOC_all. As the complexity of solving the max flow problem grows very fast with the number of images, for configurations with continuous optimization, we reduce the number of non-zero entries in each standout matrix such that the total number of nodes in the graph is around 2×10^7. These standout matrices are then used in rounding the continuous solution, but in the greedy ascent procedure we switch to standout matrices with 1000 non-zero entries. For configurations without the continuous optimization, we always use the standout matrices with 1000 non-zero entries. Also, to reduce the memory footprint of our method, we prefilter the set of potential neighbors of each image for the class *person* that contains 1023 pictures. Pre-filtering is done by marking 100 nearest neighbors of each image in terms of Euclidean distance between GIST [46] descriptors as potential neighbors. In the separate setting, we only apply the pre-filtering on the class *person* which has 1023 images. The other classes are sufficiently small for not resorting to the prefiltering procedure.

Table 2 shows the CorLoc values obtained by our method with different configurations compared to Cho *et al*. It can be seen that the ensemble postprocessing and the continuous optimization are also helpful on this dataset. We obtain the best result with the configuration that includes both of them, which is 1.6% better than Cho *et al*. However, our performance is still inferior to state of the art in image colocalization [31, 49] which employ deep features from convolutional neural networks trained for image classification and explicitly exploits the single-class assumption.

Mixed setting. We now compare in Table 3 the performance of our algorithm to Cho *et al*. in the mixed setting (none of the other methods is applicable to this case). It can be seen that our algorithm without the continuous optimization has the best performance among those in consideration. Compared to Cho *et al*., it gives a CorLoc 0.8% better on OD dataset, 4.3% better on VOC_6x2 and 2.3% better on VOC_all. The decrease in performance of our method when using the continuous optimization is likely due to the fact that we use standout matrices with only 200 non-zero entries on OD, 100 non-zero entries on VOC_6x2 and 100

Method	OD	VOC_6x2	VOC_all
Cho *et al.*	-	-	37.6
Cho *et al.*, our execution	82.2	55.9	37.5
w/o CO	**83.0 ± 0.4**	**60.2 ± 0.4**	**39.8 ± 0.2**
w CO	80.8 ± 0.5	59.3 ± 0.4	38.5 ± 0.2

Table 3: Performance on the datasets in mixed setting.

Method			VOC_6x2
$\nu = 1$	w/o CO	w/o EM	63.5 ± 1.2
		w EM	67.7 ± 0.8
	w CO	w/o EM	65.8 ± 0.8
		w EM	**68.1 ± 0.7**
$\nu = 5$	w/o CO	w/o EM	67.2 ± 1.0
		w EM	68.7 ± 0.5
	w CO	w/o EM	68.1 ± 0.9
		w EM	**69.4 ± 0.3**
$\nu = 10$	w/o CO	w/o EM	68.6 ± 1.0
		w EM	69.1 ± 0.3
	w CO	w/o EM	68.9 ± 0.7
		w EM	**70.0 ± 0.3**

Table 4: Performance of different configurations of our algorithm with $\nu = 1$, $\nu = 5$ and $\nu = 10$.

non-zero entries on VOC_all (due to the limit on the number of nodes of the bipartite graphs) in the configuration with the continuous optimization while we use standout matrices with 1000 non-zero entries in the configuration without the continuous optimization.

Sensitivity to ν. We compare the performance of our method when using different values of ν on the VOC_6x2 dataset.[3] Table 4 shows the CorLoc obtained by different configurations of our algorithm, all with normalized standout. The performance consistently increases with the value of ν on this dataset. In all other experiments however, we set $\nu = 5$ to ease comparisons to [12].

Using deep features. Since activations from deep neural networks trained for image classification (deep features) are known to be better image representations than handcrafted features in various tasks, we have also experimented with such descriptors. We have replaced WHO [19] by activations from different layers in VGG16 [42], when computing the appearance similarity between regions. In this case, the similarity between two regions is simply the scalar product of the corresponding deep features (normalized or not). As a preliminary experiment to evaluate the effectiveness of deep features, we have run our algorithm without the continuous optimization with the standout score computed using layers *conv4_3*, *conv5_3* and *fc6* in VGG16. Table 5 shows the results of these experiments. Surprisingly, most of the deep features tested give worse results than WHO. This may be due to the fact that our matching task is more akin to image retrieval than classification, for which deep features are typically trained. Among those tested, only a variant of the features extracted from the layer *conv5_3* of VGG16 gives an improvement (about 2%) compared to the result obtained by using WHO.

[3]Note that we have also tried the interpretation of ν as the maximum number of objects per image, without satisfying results so far.

Features			Average
WHO			68.8 ± 0.5
conv4_3	warping + center cropping	unnormalized	64.2 ± 0.2
		normalized	57.1 ± 0.6
	ROI pooling [18]	unnormalized	63.1 ± 0.2
		normalized	63.4 ± 0.4
conv5_3	warping + center cropping	unnormalized	64.9 ± 0.2
		normalized	64.1 ± 0.4
	ROI pooling [18]	unnormalized	**70.7 ± 0.2**
		normalized	68.2 ± 0.3
fc6	warping + center cropping	unnormalized	61.3 ± 0.2
		normalized	61.0 ± 0.4

Table 5: Performance of our algorithm with deep features on VOC_6x2 in the separate setting.

Unsupervised initial proposals. It should be noted that, although our algorithm like that of Cho *et al*. [12] is totally unsupervised once *given the region proposals*, the randomized Prim's algorithm itself is supervised [33]. To study the effect of this built-in supervision, we have also tested the unsupervised *selective search* algorithm [47] for choosing region proposals. We have conducted experiments on VOC_6x2 dataset with the three different settings of selective search (*fast*, *medium* and *quality*). As one might expect, the *fast* mode gives the smallest number of proposals and of *positive* ones (proposals whose IoU with one ground truth box is greater than 0.5); the *quality* mode outputs the largest set of proposals and of positive ones, the *medium* mode lies in-between. To compare with [12], we also run their public software with each mode of selective search.

Proposal algorithm		Cho *et al.*	Ours
selective search	*fast*	23.3	41.4 ± 0.5
	medium	20.6	48.4 ± 0.5
	quality	32.6	62.8 ± 0.6
randomized Prim's		67.6	69.4 ± 0.4

Table 6: Object discovery on VOC_6x2 with selective search and randomized Prim's as region proposal algorithms.

The results are shown in Table 6. It can be seen that the performance of both Cho *et al*.'s method and ours drop significantly when using selective search. This may be due to the fact that the percentage of positive proposals found by selective search is much smaller than that of RP. However, we see that with the *quality* mode of selective search, our method gives results quite close to those of RP, whereas the method in [12] fails badly. This suggests that our method is more robust.

Visualization. In order to gain insight into the structures discovered by our approach, we derive from its output a graph of image regions and visualize its main connected components. The nodes of this graph are the image regions that have been finally retained. Two regions (i, k) and (j, l) are connected if the images containing them are neighbors in the discovered undirected image graph (e_{ij} or $e_{ji} = 1$)

Figure 2: Visualization of VOC_6x2 in the mixed setting. The figure shows the third component in the graph of regions, corresponding roughly to class *motorbike*. The two first components are shown in Fig.1.

and the standout score between them, S_{ij}^{kl}, is greater than a certain threshold.

Choosing the threshold to get a sufficient number of large enough components for visualization purpose has proven difficult. We used instead an iterative procedure: the graph is first constructed with a high threshold to produce a small number of connected components of reasonable size, which are removed from the graph. On the remaining graph, a new, suitable threshold is found to get new components of sufficient size. This is repeated until a target number of components is reached.

When applied to our results in the mixed setting on VOC_6x2 dataset, this visualization procedure yields clusters that roughly match object categories. In Figure 1, we show sub-sampled graphs (for visualization purpose) of the two first components, which roughly correspond to classes *bicycle* and *aeroplane*. The third component is shown in Figure 2. Although containing also images of other classes, it is by far dominated by *motorbike* images. The visualization suggests that our model does extract meaningful semantic structures from the image collections and regions they contain.

5. Conclusion

We have presented an optimization-based approach to fully unsupervised image matching and object discovery and demonstrated its promise on several standard benchmarks. In its current form, our algorithm is limited to relatively small datasets. We are exploring several paths for scaling up its performance, including better mechanisms based on deep features and the PHM algorithm for prefiltering image neighbors and selecting regions proposals. Future work will also be dedicated to developing effective ensemble methods for discovering multiple objects in images, further investigating a symmetric version of the proposed approach using an undirected graph, understanding why deep features do not give better results in our context, and improving our continuous optimization approach so as to handle large datasets in a mixed setting, perhaps through some form of variable clustering.

Appendix: Maximization of supermodular cubic pseudo-Boolean functions

An immediate corollary of [7, Lemma 1] is that a cubic pseudo-Boolean function with nonegative trinary coefficients and no binary terms is supermodular. For fixed λ and μ, this is obviously the case for the Lagrangian K in (5).

In addition, the unary terms in K are nonpositive, and the Langragian can thus be rewritten, up to some constant additive term, in the form

$$f(x_1, \ldots, x_n) = \sum_{i \in U} c_i \bar{x}_i + \sum_{(i,j,k) \in T} c_{ijk} x_i x_j x_k, \quad (19)$$

where $\bar{x}_i = 1 - x_i$ (the *complement* of x_i), $U \subset \{1, \ldots, n\}$, $T \subset \{1, \ldots, n\}^2$, and all coefficients c_i and c_{ijk} are positive. We specialize in the rest of this section the general maximization method of [7] to functions of this form.

The *conflict graph* [7, 10] $G(f)$ associated with such a function f has as a set of nodes $X(f) = V \cup W$, where the elements of V correspond to linear terms, those of W correspond to cubic terms, and an edge links to nodes when one of the corresponding terms contains a variable, and the other one its complement. By construction $G(f)$ is a bipartite graph, with edges joining only elements of V to elements of W.

As shown in [7] maximizing f amounts to finding a maximum weight stable set in $G(f)$, where the nodes of V are assigned weights c_i and the nodes of W are assigned weights c_{ijk}, which in turn reduces to computing a maximum flow between nodes s and t in the network deducted from $G(f)$ by (1) adding a source node and edges with upper capacity bound c_i between s and the corresponding elements of V; (2) adding a sink node t and edges with upper capacity bound c_{ijk} between the corresponding elements of W and t; (3) assigning to all edges (from V to W) in $G(f)$ an upper capacity bound of $+\infty$.

Let $[A, \bar{A}]$ denote the minimum cut obtained by computing the maximum flow in this graph, where s is an element of A and t is an element of $\bar{A} = X(f) \setminus A$. The maximum weight stable set is then $S = (A \cap V) \cup (\bar{A} \cap W)$. The monomials $\bar{x}_i$ and $x_i x_j x_k$ associated with elements of S are set to 1, from which the values of all variables are easily deduced.

Acknowledgments. This work was supported in part by the Inria/NYU collaboration agreement, the Louis Vuitton/ENS chair on artificial intellgence and the EPSRC Programme Grant Seebibyte EP/M013774/1. We also thank Simon Lacoste-Julien for his valuable comments and suggestions.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *ICCV*, 2015.

[2] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Learning from narrated instruction videos. *PAMI*, 40(9):2194–2208, 2018.

[3] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

[4] F. Bach and Z. Harchaoui. DIFFRAC : a discriminative and flexible framework for clustering. In *Proc. Neural Info. Proc. Systems*, 2007.

[5] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 1981.

[6] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *International Conference on Computational Learning Theory*, pages 624–638. Springer, 2004.

[7] A. Billionnet and M. Minoux. Maximizing a supermodular pseudoboolean function: a polynomial algorithm for supermodular cubic functions. *Discrete Applied Mathematics*, 12:1–11, 1985.

[8] P. Bojanowski and A. Joulin. Unsupervised learning by predicting noise. In *ICML*, 2017.

[9] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce, and C. Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015.

[10] E. Boros and P. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[11] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[12] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, 2015.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[14] T. Deselaers, B. Alexe, and V. Ferrari. Localizing Objects While Learning Their Appearance. In *ECCV*, 2010.

[15] C. Doersch, A. Gupta, and A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[16] A. Faktor and M. Irani. Clustering by composition–unsupervised discovery of image categories. In *ECCV*, 2012.

[17] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010.

[18] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[19] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012.

[20] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[22] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. 2016.

[23] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[24] A. Joulin, K. Tang, and L. Fei-Fei. Efficient Image and Video Co-localization with Frank-Wolfe Algorithm. In *ECCV*, 2014.

[25] G. Kim and E. Xing. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, 2011.

[26] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[28] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised object discovery and tracking in video collections. In *ICCV*, 2015.

[29] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[30] Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *CVPR*, 2010.

[31] Y. Li, L. Liu, C. Shen, and A. Hengel. Image co-localization by mimicking a good detector's confidence score distribution. In *ECCV*, 2016.

[32] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[33] S. Manen, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized Prim's algorithm. In *ICCV*, 2013.

[34] M. Matthieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.

[35] A. Nedić and A. Ozdaglar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4), 2009.

[36] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[37] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2106.

[38] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. Neural Info. Proc. Systems*, 2015.

[39] M. Rubinstein and A. Joulin. Unsupervised Joint Object Discovery and Segmentation in Internet Images. In *CVPR*, 2013.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[41] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[43] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, 2008.

[44] M. Slater. Lagrange multipliers revisited. *Cowles Commission Discussion Paper No. 403*, 1950.

[45] K. Tang, A. Joulin, and L.-j. Li. Co-localization in Real-World Images. In *CVPR*, 2014.

[46] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *CVPR*, 2008.

[47] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013.

[48] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[49] X. Wei, C. Zhang, Y. Li, C. Xie, J. Wu, C. Shen, and Z. Zhou. Deep descriptor transforming for image co-localization. In *IJCAI*, 2017.