Unsupervised Feature Optimization (UFO): simultaneous selection of multiple features with their detection parameters

Leonid Karlinsky Michael Dinerstein Shimon Ullman
{leonid.karlinsky,michael.dinerstein,shimon.ullman}@weizmann.ac.il
Weizmann Institute of Science, Rehovot 76100, Israel

Abstract

Class learning, both supervised and unsupervised, requires feature selection, which includes two main components. The first is the selection of a discriminative subset of features from a larger pool. The second is the selection of detection parameters for each feature to optimize classification performance. In this paper we present a method for the discovery of multiple classification features, their detection parameters and their consistent configurations, in the fully unsupervised setting. This is achieved by a global optimization of joint consistency between the features as a function of the detection parameters, without assuming any prior parametric model. We demonstrate how the proposed framework can be applied for learning different types of feature parameters, such as detection thresholds and geometric relations, resulting in the unsupervised discovery of informative configurations of objects parts. We test our approach on a wide range of classes and show good results. We also demonstrate how the approach can be used to unsupervisedly separate and learn visually similar subclasses of a single category, such as facial views or hand poses. We use the approach to compare various criteria for feature consistency, including Mutual Information, Suspicious Coincidence, L2 and Jaccard index. Finally, we compare our approach to a parametric consistency optimization technique such as pLSA and show significantly better performance.

1. Introduction

In this paper we consider the problem of unsupervised selection of multiple classification features together with the optimization of their detection parameters and discovery of consistent feature configurations. The input is a mixed set of unlabeled images S, only a small subset of which (20% or lower) contains instances of an unknown class category (which may be uncropped, unaligned and of small size relative to the background), and a large pool of candidate features $\mathcal{F}$ (is of size 1000-3000 in our experiments), which is measured on each image. The feature pool may also be generated directly from all the images in the unsupervised training set (e.g. by applying the method of [9]). As in many classification schemes, each feature $F_i \in \mathcal{F}$ is associated with some parameters θ_i that need to be set. When θ_i are fixed, the feature $F_i(I;\theta_i)$ is a function mapping image I to some discrete (usually binary) value. For example, the parameter θ_i may be the similarity threshold and the feature F_i be a binary function with $F_i = 1$ iff the maximal similarity between all descriptors extracted from image I and a descriptor associated with F_i exceeds θ_i. Another example is that the feature is a pair (F_i, F_j) (where F_i and F_j are "maximal similarity + threshold" features from the previous example) and θ_i is an expected spatial offset between the detected locations of the two members of the pair. The combined feature is detected only if the observed offset is close to θ_i. Finally, θ_i may be the linear coefficients for combining similarity measures of several different types of descriptors for a given image patch associated with F_i. Such de-

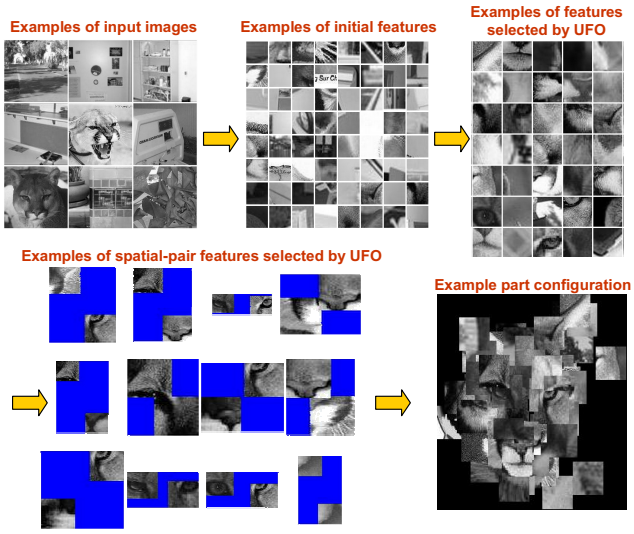


Figure 1. An example of the application of our method to cougar class from Caltech-101.

scriptor combinations were successfully used in [21]. The main goal of the algorithm is to identify a subset of the candidate features that are most discriminative for detecting the category instances together with optimizing parameters of each feature to find their most discriminative values. This is challenging since the image class labels are unknown and cannot be used to identify useful features. Another goal is to use the discovered subset of optimized features to detect and localize a reliable subset of class instances present in the unsupervised training set. The detected class examples can then be used for the subsequent category learning (e.g. as done in [14, 10]). Figure 1 illustrates the input, the various outputs and the intermediate stages of our method.

The problem of unsupervised learning of an unknown category has recently attracted substantial attention [2, 4, 5, 6, 7, 8, 10, 20, 15, 12, 13, 14, 19, 17, 18, 23]. The unsupervised approaches may be further subdivided into weakly supervised - ones that assume that all training images contain uncropped and unaligned instances of the learned object category [6, 15, 18, 23], and fully unsupervised - ones for which the category instances may appear only in a small portion of the training images [8, 20, 7, 5, 19, 4, 17, 12, 2, 13, 14, 10]. The approach proposed in this paper belongs to the latter, fully unsupervised, category. The general idea in all of these approaches is to compensate for the lack of supervision by searching for consistency between the measured features, assuming that this consistency arises primarily from the presence of a class instance in the image. The approaches may be further categorized in terms of how they search for this inter-feature consistency. Some approaches assume a prior parametric model for the features and their detection parameters, and the consistency is detected via unsupervised training of the parameters of this model. The models used by these approaches include the constellation model [6] used by [6, 7], pLSA used by [20, 5, 19, 12, 14], Spatial-LTM [4], ISM [11] used by [8], Semantic-Shift [13], TSI-pLSA [5] and UCA [10]. Other approaches are non-parametric in a sense that no prior model is assumed for the inter-feature consistency and the consistency is detected by some form of a bottom-up agglomerative process. This process builds upon local, usually pairwise (for a given level of agglomeration), consistency relations between the features. Examples of the latter approaches are efficient mining used by [17, 18], joining frequent feature triplets with high suspicious coincidence measure by [23] and combining similar image segments by [2]. Our approach belongs to the latter non-parametric type. However, it does not perform an iterative agglomeration based on the local consistency of the features. Instead, a global optimization is performed, optimizing the consistency between all the potential features as a function of the feature parameters. The output of the global optimization is a consistent setting of feature parameters, maximizing the sum of pairwise consistency values

Figure 2. Examples of part configurations learned by UFO for various object classes. Each configuration displays a set of patch features automatically selected and arranged by the algorithm for each category.

between the relevant features. Clustering the resulting consistency graph between the features gives rise to consistent feature clusters, which are discriminative for the unknown learned category (or several categories in case of a multi-class), as demonstrated in our experiments.

An important consideration is that if we want to compute consistency of feature F_1 with two different features F_2 and F_3, then the parameter (e.g. threshold) of F_1 that attains maximal consistency with F_2 may be different from the one needed for maximal consistency with F_3. Consequently, in order to have maximal joint consistency of many features, the parameter optimization needs to be global. An agglomerative process that optimizes consistency only locally by merging a few (usually 2 or 3) features at a time, may arrive at lower consistency at the higher levels of agglomeration or become inconsistent in the setting of parameters.

As in previous approaches, the proposed framework can be used with different types of consistency measures. Therefore, in our experiments we used the proposed method to compare different consistency measures, such as Suspicious Coincidence (SC), Mutual Information (MI), Jaccard Index (JI) and L2 distance. Surprisingly, the standard SC consistency measure attained lower results in our comparison, significantly inferior to MI and JI measures. In addition, we compare our approach to pLSA - probably the most widely used parametric learning method based on consis-

tency, and show significantly better performance.

Unsupervised learning can be particularly useful in the weakly supervised setting, when all the images contain instances of a given class, but it is beneficial to divide the class into a set of visual sub-classes (such as different object views or different poses), to obtain better recognition of each sub-class individually, instead of recognizing a mixture of sub-classes by a single model. In the experiments below, we demonstrate how the proposed method can be used to obtain unsupervised separation between visually similar sub-classes, such as separating face views and separating hand poses.

Finally, we exploit the fact that the proposed framework may be used with any types of features and their parameters, and demonstrate how our approach can be used in a pipeline that starts from a generic set of quantized image patch descriptors and a set of unlabeled images, and finally arrives at discovering spatial configurations of object parts, which are both discriminative and visually plausible (see Figure 2). Within this pipeline, the method is applied twice: first for learning detection thresholds, and second for learning spatial offset parameters of feature pairs, finally leading to the discovery of full $2D$ consistent feature configurations.

The rest of the paper is organized as follows, section 2 describes the proposed approach, section 3 summarizes the experimental results, and section 4 provides summary and discussion.

2. Method

In this section we describe the Unsupervised Feature Optimization (UFO) algorithm (section 2.1), discuss and analyze the pair-wise consistency measures that can be efficiently used within the algorithm (section 2.2), and show how the UFO algorithm can be applied to select appearance and geometric features, including their respective detection parameters, in order to discover objects and consistent geometric configurations of their parts (section 2.3).

The UFO algorithm receives as input a set of unlabeled images and a feature pool from which it performs the selection. However, in practical applications, such as described in section 2.3, it is possible to work with either an external feature pool provided by the user, or with a feature pool internally generated by the system from all the unsupervised training images (e.g. by applying the method by [9]). A schematic diagram describing the proposed method is given in figure 3 and explained further in section 2.3.

2.1. UFO algorithm

Input: A set of images $S = \{I_n\}_{n=1}^{N}$ and a (large) pool of features together with their respective detection parameters $\mathcal{F}_0 = \{\langle F_i, \theta_i \rangle\}_{i=1}^{M}$, such that each feature is a function: $\langle F_i, \theta_i \rangle : S \rightarrow V$, here V is a discrete set of possible feature values (we use binary values in our experiments), and each θ_i takes a value from a discrete set of possible values that may be specific to each feature $\theta_i \in W_i$. For simplicity, we will use the notation $F_i(\theta_i)$ to indicate a row vector of N values from V, the n-s entry of which will be $\langle F_i, \theta_i \rangle$ measured on image I_n. We will also refer to the feature itself as F_i.

Output: K disjoint subsets of features $\bigcup_{1 \le k \le K} \hat{\mathcal{F}}_k = \mathcal{F}_0$, and the respective optimal parameter settings for each feature: $\left\{\hat{\theta}_i\right\}_{i=1}^{M}$. The goal is that one or several of these subsets will be associated with the classes present in the images of set S.

The flow of the algorithm:

1. Compute maximal consistency for each pair of features F_i and F_j: $C_{ij} = \max_{\theta_i, \theta_j} Cons\left[F_i(\theta_i); F_j(\theta_j)\right]$, where the consistency measure, $Cons$, is a function measuring pairwise consistency between two vectors from V^N: $Cons : V^N \times V^N \rightarrow \mathbb{R}$. The consistency measures that we have compared within our algorithm, as well as methods to efficiently compute $Cons\left[F_i(\theta_i); F_j(\theta_j)\right]$ for all i and j and all values of θ_i and θ_j, are described in section 2.2.

2. Transform the resulting $M \times M$ matrix $C = \{C_{ij}\}_{i,j=1}^{M}$ into a graph adjacency matrix A. This is obtained by setting all except the largest L entries of C to zero and the largest L entries to one (C and A are symmetric, we refer to L entries above the diagonal and keep their L respective reflections as well). The resulting graph has M nodes, one node for each feature, and L edges. Throughout all our experiments we used $L = 10,000$. In principle a large value for L is preferred, the only reason to limit its value is to lower the running time of the graphical model based optimization that will be described next. For each edge described by non-zero entry A_{ij} we compute a potential function: $w_{ij} : W_i \times W_j \rightarrow \mathbb{R}$ such that: $w_{ij}(\theta_i, \theta_j) = Cons\left[F_i(\theta_i); F_j(\theta_j)\right]$. The potentials are then used to define an energy function E being the sum of the pairwise consistency measures:

$$E(\theta_1, \ldots, \theta_M) = \sum_{i,j:A_{ij}=1} w_{ij}(\theta_i, \theta_j) \qquad (1)$$

Energy function E is a sparse approximation of the full pairwise consistency measure between all the features.

3. Apply a Loopy-Belief-Propagation (LBP) algorithm to compute (approximate) the maximal assignment to E:

$$\left\{\hat{\theta}_i\right\}_{i=1}^{M} \underset{LBP}{\approx} \arg\max E(\theta_1, \ldots, \theta_M) \qquad (2)$$

4. Finally, compute: $\hat{C}_{ij} = Cons\left[F_i\left(\hat{\theta}_i\right); F_j\left(\hat{\theta}_j\right)\right]$ and cluster the resulting $M \times M$ affinity matrix $\hat{C} = \left\{\hat{C}_{ij}\right\}_{i,j=1}^{M}$ into K clusters (we use $K = 7$ in all our experiments). In our implementation of the UFO algorithm, we use spectral clustering [22] for the last step.

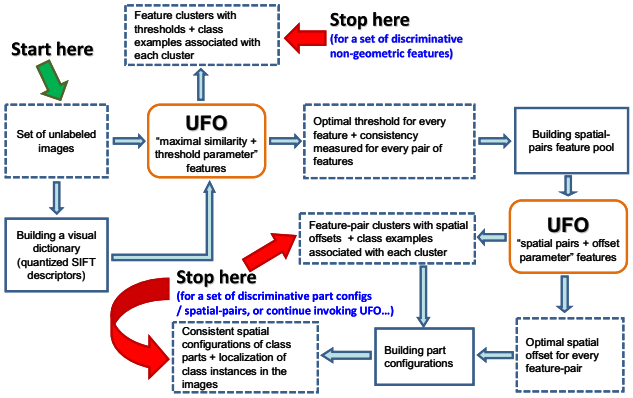


Figure 3. A schematic diagram showing the flow of the method. Details are explained in section 2.3. The UFO algorithm, explained in section 2.1, is applied twice, first for the non-geometric "maximum-similarity" features, and second for the geometric "spatial-pair" features constructed using the results of the first UFO application. The dashed boxes describe input, output and intermediate results. The solid boxes describe assisting algorithms explained in the section 2.3. Entry and optional exit points of the flow are indicated by green and red arrows respectively.

Intuition - Class and feature consistency: As will be demonstrated in the results section 3, the feature clusters obtained by the method are closely associated with the unknown classes. The intuition behind the UFO algorithm is that the main source of consistency between the features is their association with specific classes. For features with high enough consistency with the same class (or sub-class), there will also be high consistency between the features themselves. This is intuitive, but also follows from the lower bounds on feature-to-feature consistency derived analytically for all the consistency measures in the next section 2.2. The bounds become tighter as the consistency between the features and the class (for the optimal setting of detection parameters) increases. Therefore, keeping graph edges with high consistency C_{ij} has a high likelihood to form edges between features consistent with the same class. Moreover, choosing detection parameters $\left\{\hat{\theta}_i\right\}_{i=1}^{M}$ that maximize consistency between the features, will also contribute to increasing the feature-to-class consistency.

The following section describes several natural consistency measures that can be efficiently used within the proposed UFO algorithm, and provides lower bounds for feature-to-feature consistency in terms of the respective feature-to-class consistency for each of the measures.

2.2. Consistency measures and efficient implementation

In our experiments with the UFO algorithm we tested and compared four consistency measures (denoted $Cons$ in the description of the algorithm), namely: Jaccard Index (JI), Mutual Information (MI), Suspicious Coincidence (SC) and Euclidian Distance (L2). As explained in section 2.3, in our experiments we focused on binary features and their parameters. In order to implement the UFO algorithm efficiently, we need to efficiently estimate $Cons\left[F_i\left(\theta_i\right);F_j\left(\theta_j\right)\right]$ for all i and j and all values of θ_i and θ_j. Fortunately, in the binary feature case, the sufficient statistics for computing $Cons\left[F_i\left(\theta_i\right);F_j\left(\theta_j\right)\right]$ for all the listed measures are only five numbers: the number of images for which both $F_i\left(\theta_i\right)$ and $F_j\left(\theta_j\right)$ are equal to one (the inner product $F_i\left(\theta_i\right)\cdot F_j\left(\theta_j\right)$), the number of images for which they are both zero $\bar{F}_i\left(\theta_i\right)\cdot\bar{F}_j\left(\theta_j\right)$, the $\bar{F}_i\left(\theta_i\right)\cdot F_j\left(\theta_j\right)$ (needed for MI), and the numbers of ones in $F_i\left(\theta_i\right)$ and $F_j\left(\theta_j\right)$ respectively ($\left|F_i\left(\theta_i\right)\right|_1$ and $\left|F_j\left(\theta_j\right)\right|_1$). W.l.o.g., assume that the sets of possible values for the detection parameters: W_i are of the same cardinality $|W_i|=T$ for all i. Then the complexity of the full computation of all the required sufficient statistics is $2\cdot T^2$, and in case of MI $3\cdot T^2$, times the complexity of binary matrix multiplication.

In the following text we describe all the tested consistency measures and for each of them provide a lower bound on feature-to-feature consistency $Cons\left[F_i\left(\theta_i\right);F_j\left(\theta_j\right)\right]$ (we will write $Cons\left[F_i;F_j\right]$ for brevity) in terms of respective feature-to-class consistencies: $Cons\left[F_i;C\right]$ and $Cons\left[F_j;C\right]$. The bounds are monotonically increasing in both $Cons\left[F_i;C\right]$ and $Cons\left[F_j;C\right]$, and become tighter as the feature-to-class consistencies approach their maximum. A notable observation regarding these bounds is that they hold for any class labeling C. Therefore, even though two features might have low consistency with the entire class (e.g. faces), they might be very consistent with the same sub-class (e.g. profile faces) and thus have high lower bound on their feature-to-feature consistency (using this sub-class labeling as C). Supporting this claim, we have successfully applied our method for unsupervised separation of visually similar sub-classes of a larger class, such as separating face views and separating hand poses (see section 3 for more details).

Jaccard Index (JI): The JI measures the amount of consistency between two sets by computing the ratio between their intersection and their union. The JI between two binary vectors X and Y is defined as:

$$JI\left(X,Y\right)=\frac{X\cdot Y}{X+Y-X\cdot Y} \tag{3}$$

where $X\cdot Y$ is the dot product. This is exactly the classical JI for sets of indices of ones in the two vectors. In our experiments we use the following symmetric variant of JI:

$$\widehat{JI}\left(X,Y\right)=0.5\cdot\left[JI\left(X,Y\right)+JI\left(\bar{X},\bar{Y}\right)\right] \tag{4}$$

where $\bar{X}=1-X$. This variant is used in order to prevent an artificial tendency for larger consistency due to greater number of ones in the binary vectors.

Claim 2.1

$$JI\left[F_i;F_j\right] \geq \frac{JI\left[F_i;C\right]+JI\left[F_j;C\right]-1}{1/JI\left[F_i;C\right]+1/JI\left[F_j;C\right]-1} \quad (5)$$

similar result holds for $\widehat{JI}\left[F_i;F_j\right]$.

The proofs of this and following claims are given in the supplementary material. This shows that $\widehat{JI}\left[F_i;F_j\right]$ is increasing in both the JIs: $JI\left[F_i;C\right]$ and $JI\left[F_j;C\right]$, and the bound (5) becomes tight when the JIs attain their maximal values of 1.

Mutual Information (MI): The MI measures the decrease in the entropy of one random variable conditioned on another random variable. For two binary vectors X and Y:

$$MI\left[X;Y\right] = H\left(X\right)+H\left(Y\right)-H\left(X,Y\right) \quad (6)$$

where H is the Shannon's entropy of a random variable.

Claim 2.2 *Assuming that F_i and F_j are conditionally independent given class C, then:*

$$MI\left[F_i;F_j\right] \geq MI\left[F_i;C\right]+MI\left[F_j;C\right]-H\left(C\right) \quad (7)$$

Because the maximal value that $MI\left[F_i;F_j\right]$ can attain is: $\min\left[H(F_i),H(F_j)\right]$, maximizing $MI\left[F_i;F_j\right]$ will tend to prefer F_i and F_j that are correlated and non-sparse in terms of both zeros and ones (with higher entropy). Unlike MI, JI can have high values for correlated features that appear very infrequently (these features will have low MI scores, because of their low entropy).

Suspicious Coincidence (SC): The SC measures the ratio between the probability of two events happening simultaneously and an estimate of this probability assuming the two events are independent. For binary vectors X and Y of length N:

$$SC\left[X;Y\right] = \frac{N\cdot\left(X\cdot Y\right)}{\left|X\right|_1\cdot\left|Y\right|_1} \quad (8)$$

Claim 2.3 *Assume the events $F_i=1$ and $F_j=1$ are conditionally independent given the event $C=1$, then:*

$$SC\left[F_i;F_j\right] \geq SC\left[F_i;C\right]\cdot SC\left[F_j;C\right]\cdot\left|C\right|_1/N \quad (9)$$

The SC is a commonly used consistency measure, e.g. in [23] it was used for weakly supervised learning of object contours. A drawback of the SC is that taken by itself it may prefer very infrequent events. The reason is that if X has few ones in it, then $SC\left[X;X\right]=1/\left|X\right|_1$ is very large. Therefore, SC is usually used in conjunction with a threshold on minimal frequency of the events (i.e. demanding that $\left|X\right|_1$ is large enough). However, this is problematic in our fully unsupervised setup, since the class instances themselves are infrequent (there may be 20% or less class instances in a set). This inherent drawback of the SC is supported by our experiments (section 3), where SC performs significantly worse then MI or JI.

Euclidian Distance (L2): The bounds for L2 are provided by the triangular inequality. A drawback of L2 is that it is dominated by the larger of $X\cdot Y$ and $\bar{X}\cdot\bar{Y}$ (X and Y defined as above), and hence may disregard the lack of consistency between either zeros or ones in the vectors X and Y, depending which (zeros or ones) are less frequent.

2.3. Implementation Details

The flow of the proposed method is schematically shown in figure 3. In this section we describe the auxiliary steps of the method in more detail. All the constant numbers appearing in the description are fixed parameters of the method and are used throughout the experiments.

Initial feature pool: we apply the method of [9] on the entire image set S (both class and non-class, since labeling is unknown) to build an initial feature pool $\mathcal{F}_0$ of $M_0=1000$ quantized SIFT descriptors of 40×40 image patches. For every image in the set S, we compute the similarity of each feature from the pool at all locations on a dense grid. For each feature $F_i\in\mathcal{F}_0$ and image $I_n\in S$, the five highest similarity local maxima locations are retained. Denote their respective similarity scores by $0\leq\alpha_{i,n}^k\leq 1$ (in decreasing order for each feature) and image locations by $L_{i,n}^k\in\mathbb{R}^2$, where $k\in\{1,\ldots,5\}$.

Unsupervised optimization of threshold parameters: we apply the UFO algorithm (section 2.1) to learn individual optimal threshold parameters θ_i^{thr} for each $F_i\in\mathcal{F}_0$. We use only the maximal similarity scores $\alpha_{i,n}^1$ in this optimization. For each of the optimized threshold parameters θ_i^{thr} we set a range of 20 candidate values W_i^{thr}, by taking 20 values with equal spacing in the index from the sorted list $sort\left(\alpha_{i,1}^1,\alpha_{i,2}^1,\ldots,\alpha_{i,N}^1\right)$. This ensures that each feature has an adequate set of candidate values for its threshold for any density of its continuous similarity values. As explained in section 2.1, following threshold setting, the UFO also produces clusters of features, which are tested for their classification performance in the results section 3.

Building spatial-pairs feature pool: an additional output of the UFO (section 2.1) is the feature-to-feature consistency matrix $\hat{C}=\left\{\hat{C}_{ij}\right\}_{i,j=1}^{M_0}$ computed for the chosen optimal threshold parameters. We next build spatial-pair features from $M_1=3000$ pairs of $\mathcal{F}_0$ features that have maximal pair-wise consistency (i.e. maximal entries in the matrix $\hat{C}$). The spatial-pair features are pairs of $\mathcal{F}_0$ features that have a Gaussian model for the spatial offset between their detected locations. For each spatial-pair feature in the resulting feature pool, denoted $\mathcal{F}_1$, we train a separate Gaussian Mixture Model (GMM) producing five competing Gaussian spatial offset models. The GMM for a spatial-pair feature $(F_i,F_j)\in\mathcal{F}_1$ is trained on a set $O_{i,j}$ of offsets between F_i and F_j in all their detected locations (from all the images)

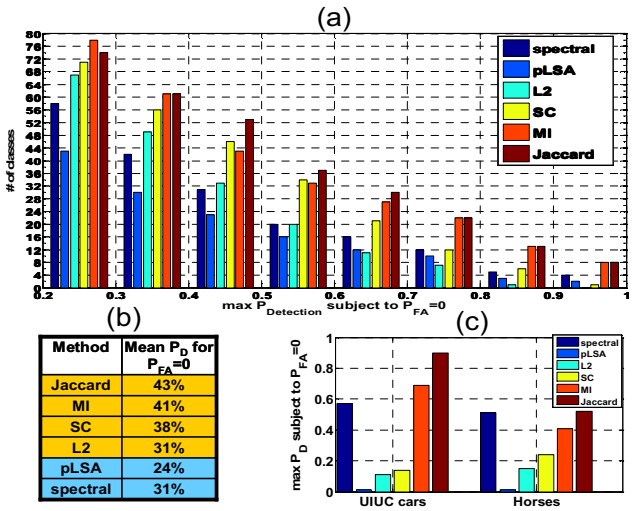
that passed the optimal thresholds (computed by UFO) $\hat{\theta}_i^{thr}$ and $\hat{\theta}_j^{thr}$ respectively: $O_{i,j} = \left\{ L_{i,n}^{k_1} - L_{j,n}^{k_2} \mid 1 \le n \le N, \right.$ $\left. 1 \le k_1, k_2 \le 5,\ \alpha_{i,n}^{k_1} > \hat{\theta}_i^{thr},\ \alpha_{j,n}^{k_2} > \hat{\theta}_j^{thr} \right\}$. Denote the Gaussian components of the learned mixture: $G_{i,j}^k = N\left(\mu_{i,j}^k, \Sigma_{i,j}^k\right)$, where $k \in \{1, \ldots, 5\}$. The detection parameter for the spatial-pair feature $(F_i, F_j) \in \mathcal{F}_1$ is $\theta_{i,j}^{offs} \in \{1, \ldots, 5\}$, and the pair-feature is considered detected in image $I_n \in S$ with $\theta_{i,j}^{offs} = k$, iff there exists an offset in $O_{i,j}$ that originates from image I_n and for that offset the $G_{i,j}^k$ component has maximum likelihood among all the GMM components.

Unsupervised optimization of offset parameters: we apply the UFO algorithm again, this time for choosing the correct Gaussian offset model among the five individual candidates for each spatial-pair feature. The outputs of the algorithm are the optimal offset models $\hat{G}_{i,j} = N\left(\hat{\mu}_{i,j}, \hat{\Sigma}_{i,j}\right)$ for the spatial-pair features $(F_i, F_j) \in \mathcal{F}_1$, clusters of these features, and pair-wise consistency measured between each two spatial-pair features.

Building consistent configurations of object parts: The next goal is to derive for each cluster of the spatial-pair features (computed by the UFO) 2-D feature configurations that will be as consistent as possible with the pair-wise offsets found in the previous step. Example configurations are shown in figure 2. They are usually visually plausible in the sense that they correspond to a repeating part configuration in the object. Since each feature in a configuration is associated with some object part, we call them "part configurations". Let $\{(F_{i_1}, F_{j_1}), \ldots, (F_{i_L}, F_{j_L})\} \subset \mathcal{F}_1$ be a cluster of spatial-pair features, and let $\left\{\hat{G}_{i_1,j_1}, \ldots, \hat{G}_{i_L,j_L}\right\}$ their respective optimal offset models selected by the UFO. First, we solve a 2D placement problem, where we define unknowns (x_i, y_i) for each feature F_i belonging to at least one of the pairs in the cluster, and solve the following system of linear equations, two equations for each pair in the cluster:

$$x_{i_l} - x_{j_l} = \hat{\mu}_{i_l,j_l}(1), \quad y_{i_l} - y_{j_l} = \hat{\mu}_{i_l,j_l}(2) \qquad (10)$$

where the $\hat{\mu}_{i_l,j_l}(1)$ and $\hat{\mu}_{i_l,j_l}(2)$ are the x and y components of the mean. This linear system is solved by weighted least squares with outlier rejection. The two equations of each pair are weighted by the amount of uncertainty of the respective optimal offset model, measured by the area of the covariance: $\left|\hat{\Sigma}_{i_l,j_l}\right|$. We also add two auxiliary equations $x_{i_1} = 0$ and $y_{i_1} = 0$ with large weight, as the system (10) is independent of the choice of the origin. The outlier rejection is implemented by removing pairs of equations corresponding to maximal error, until the bound of at most one pixel error is reached. The part configurations are obtained as connected components (CC) of a graph whose nodes are the features belonging to at least one of the spatial-pairs from the cluster and whose edges are the pairs themselves. The resulting configurations may then by used as 'hyper-features' for either detecting the learned objects in new images (e.g. using the ISM voting scheme proposed by [11]) or as input to subsequent invocations of the UFO in order to build even more complex features.



Method	Mean P_D for P_{FA}=0
Jaccard	43%
MI	41%
SC	38%
L2	31%
pLSA	24%
spectral	31%

Figure 4. **(a)** Summary of the comparison of UFO and baseline methods (pLSA, spectral) on the entire Caltech-101 dataset. For each method the bar-plots show a cumulative histogram of detection probability P_D at the point $P_{FA} = 0$ (no False Alarms) on the ROC. For example, bars placed between 0.2 and 0.3 count the number of classes for which the tested methods achieved at least 20% detection with no false alarms; the pLSA achieved this for 42 of the 101 classes, compared with 78 classes for UFO with MI consistency measure. We also analyzed the 27 classes with $P_D < 20\%$ for the JI consistency measure (the rightmost dark-red bar). Only 13 classes produced < 5 class examples as top scoring ones (getting 5 top examples at random in our 20% class setup has probability < 0.0003); **(b)** Mean P_D for $P_{FA} = 0$ on all Caltech-101 classes for each method; **(c)** Comparison on UIUC cars and horses datasets.

3. Experimental Results

To test the proposed approach we applied it on a wide range of classes from several datasets, including the entire Caltech-101, UIUC cars [1], horses [3], hand poses [16], a dataset of similar face views, and a dataset obtained from querying Google's image search. For every class from the Caltech-101, UIUC cars, and horses datasets, the experimental protocol was to take all the class images mixed with random selection of four times more background images from the Caltech backgrounds set. Thus in every unsupervised set there were only 20% class images. Figure 4(a,b) summarizes the comparison of the UFO algorithm with baseline approaches on the entire Caltech-101 dataset. The comparison of the different consistency measures dis-

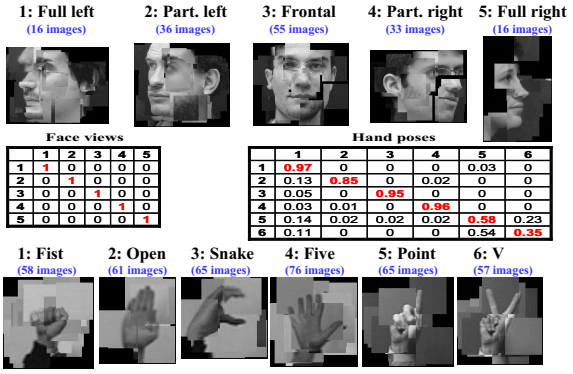


Figure 5. Summary of the multi-sub-class unsupervised separation experiment. The proposed method was applied for unsupervised multi-class separation of face views and hand poses datasets. The pLSA and spectral clustering baseline approaches performed significantly worse with at least 20% difference in mean accuracy. The images show examples of part configurations obtained for each sub-class.

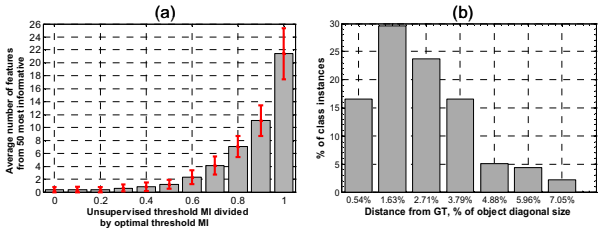


Figure 6. **(a)** Comparison of unsupervisedly learned parameters (using MI consistency measure) with optimal parameters obtained in supervised setting with respect to MI. The mean histogram was computed for 50 most informative features in 78 Caltech-101 classes for which the UFO succeeded ($P_D > 20\%$ for $P_{FA} = 0$). The rightmost bin corresponds to perfect match. The density remains almost unchanged when the histogram is computed for either 2, 10, and 100 most informative features; **(b)** Evaluation of car localization using the part configurations learned on UIUC cars dataset. The histogram shows the error distribution of a detected reference point with respect to manually marked ground truth.

cussed in section 2.2 is also included in the figure 4(a,b). The baseline methods are pLSA with 7 topics, and spectral clustering (to 7 clusters) of the normalized cross correlation of feature responses without parameter optimization. In all the experiments UFO also computed 7 clusters. To make a fair comparison, the UFO was applied with the selection of threshold parameters but not geometry, since both baseline approaches do not use geometry. Each of the compared approaches generates feature clusters (pLSA topics are soft feature clusters). The test for a good unsupervised feature cluster is the extent to which it corresponds to the unknown class. The score we have chosen to compare the different approaches, is the detection probability P_D at the point $P_{FA} = 0$ (no False Alarms) on the ROC. The reason is that applications that will use the proposed approach (or a baseline) for unsupervised extraction of class examples, will take top scoring examples and any errors in them will hinder subsequent performance. We also got very similar results in terms of the relative differences between the methods for the point of 80% precision. For each method and each class, the cluster with highest P_D for $P_{FA} = 0$ was taken into the comparison. The ROC for each pLSA topic was generated according to the topic probability in each image. For both the UFO and the spectral approach, the ROC for each cluster was generated by simply summing up the response vectors of features belonging to the cluster. For UFO, the response vectors were thresholded using the computed optimal detection thresholds (one for each feature) prior to the summation.

On figure 4c we show the results of the same comparison on the UIUC cars and the horses datasets. Only the 170 uncropped and unaligned test images from the UIUC cars dataset (mixed with random background images) were used in the experiment. Figure 6b illustrates the performance of car localization using the part configurations learned on the UIUC cars dataset. The part configurations were combined using ISM [11] and voted for a single reference point.

In addition, we have compared the percent of features originating in class images in the feature set returned by the UFO with the percent of such features in codebooks returned by the standard unsupervised feature selection and quantization methods, namely k-means and [9], that are extensively used in many current image classification approaches. The comparison showed that on our mix of 20% class 80% background images, the standard methods produce codebooks containing on average $22 \pm 8\%$ class features, while the UFO produces a set of features with $68 \pm 15\%$ class features (statistics computed on all Caltech-101 classes). Moreover, as can be seen in figure 2, the features selected by the UFO mostly come from the class object region of the class images, while it is likely that it is not so for the standard algorithms (mostly the class objects occupy less then 50% of the class images).

On two datasets, the hands (382 images, 6 poses, proposed by [16]) and the faces (156 images, 5 views), the proposed method was applied in an unsupervised multi-class manner in order to simultaneously separate and learn all the sub-classes. From the hand poses dataset only the test images were used. The resulting confusion matrices and the computed part configurations for the relevant clusters are shown in figure 5, showing perfect separation for face views and high performance for hand poses.

Figure 6a shows comparison of the parameters learned in the fully unsupervised setting by UFO with MI consistency measure, with ground truth most informative (maximal MI) parameters obtained with supervision. The comparison is

Figure 7. Summary of the experiment on 200 first images returned by "jaguar" Google's image search query. 30 of them contained (jaguar) animals the rest (jaguar) cars and noise. All processing was done in grayscale. All images were normalized to same vertical size. The green and the red boxes display first 20 images of two of the clusters returned by the UFO. The cluster associated with animals (green box), had 67% recall with 95% precision, and 83% recall with 72% precision in animal jaguar detection.

performed on the 78 Caltech-101 classes for which the UFO (with MI) exceeded 20% detection with no false alarms. Surprisingly, the unsupervised method learns globally optimal parameters for more then 40%, and near optimal parameters (MI ratio of $\geq$ 0.8) for about 80%, of the most discriminative features. Finally, figure 7 summarizes the results of an experiment of unsupervised clustering of images obtained from a query "jaguar" to Google's image search.

4. Discussion

The main novelty of the proposed approach is the discovery of multiple classification features and their detection parameters in the fully unsupervised setting by a global optimization of their joint consistency as a function of the detection parameters. The method can be applied to both unsupervised and (weakly) supervised learning tasks, as a method for reducing the dimensionality of the feature space by selecting and optimizing most discriminative features. It also proved useful as a method for unsupervised sub-class discovery. Automatic separation into meaningful sub-classes is an important tool for boosting object recognition performance, as it allows to model each sub-class individually. Future work includes extending the proposed method for optimizing descriptor combination parameters, testing additional consistency measures, and extending to the temporal domain for unsupervised discovery of consistent motion patterns, that could be applied to both action recognition and motion based object recognition.

Acknowledgment: This work was supported by EU IST Grant FP6-2005-015803.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 2004.

[2] N. Ahuja and S. Todorovic. Discovering hierarchical taxonomy of categories and shared subcategories in images. *ICCV*, 2007.

[3] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. *ECCV*, 2002.

[4] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. *ICCV*, 2007.

[5] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. *ICCV*, pages 1816–1823, 2005.

[6] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR (2)*, pages 264–271, 2003.

[7] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. *ECCV*, 2004.

[8] M. Fritz and B. Schiele. Towards unsupervised discovery of visual categories. *DAGM*, 2006.

[9] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. *ICCV*, 2005.

[10] L. Karlinsky, M. Dinerstein, D. Levi, and S. Ullman. Unsupervised classification and part localization by consistency amplification. *ECCV*, 2008.

[11] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. *ECCV*, 2004.

[12] L.-J. Li, G. Wang, and L. Fei-Fei. Optimol: automatic object picture collection via incremental model learning. *CVPR*, 2007.

[13] D. Liu and T. Chen. Semantic-shift for unsupervised object detection. *CVPR Workshop*, 2006.

[14] D. Liu and T.Chen. Unsupervised image categorization and object localization using topic models and correspondences between images. *ICCV*, 2007.

[15] N. Loeff, H. Arora, A.Sorokin, and D. Forsyth. Efficient unsupervised learning for localization and detection in object categories. *NIPS*, 2005.

[16] S. Marcel. Hand posture recognition in a body-face centered space. *Conference on Human Factors in Computer Systems (CHI)*, 1999.

[17] S. Nowozin, K. Tsuda, T. Uno, T. Kudo, and G. H. Bakir. Weighted substructure mining for image analysis. *CVPR*, 2007.

[18] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient mining of frequent and distinctive feature configurations. *ICCV*, 2007.

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

[20] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their localization in images. *ICCV*, pages 370–377, 2005.

[21] M. Varma and D. Ray. Learning the discriminative powerinvariance trade-off. *ICCV*, 2007.

[22] Y. Weiss. Segmentation using eigenvectors: A unifying view. *ICCV*, 1999.

[23] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. *ECCV*, 2008.