Multi-class Object Layout with Unsupervised Image Classification and Object Localization

Ser-Nam Lim[1], Gianfranco Doretto[2], and Jens Rittscher[1]

[1] Computer Vision Lab, GE Global Research, Niskayuna, NY 12309
[2] Dept. of CS & EE, West Virginia University, Morgantown, WV 26506

Abstract. Recognizing the presence of object classes in an image, or image classification, has become an increasingly important topic of interest. Equally important, however, is also the capability to locate these object classes in the image. We consider in this paper an approach to these two related problems with the primary goal of minimizing the training requirements so as to allow for ease of adding new object classes, as opposed to approaches that favor training a suite of object-specific classifiers. To this end, we provide the analysis of an exemplar-based approach that leverages unsupervised clustering for classification purpose, and sliding window matching for localization. While such exemplar based approach by itself is brittle towards intraclass and viewpoint variations, we achieve robustness by introducing a novel Conditional Random Field model that facilitates a straightforward accept/reject decision of the localized object classes. Performance of our approach on the PASCAL Visual Object Challenge 2007 dataset demonstrates its efficacy.

1 Introduction

In recent years, the integration of the tasks of *image classification* and *object localization* has generated a great amount of interest [1,2,3,4,5,6,7,8,9,10,11]. The first task is concerned with labeling an image with tags describing the object classes depicted in it. The second task is concerned with localizing, typically by a bounding box, the objects described by such tags in the image. The rationale for combining these two tasks is that solving the first one would improve the solution to the second one and vice versa [3]. Additionally, it is of great common interest to not only know the presence of certain object classes in the image, but also their locations. Despite all the progress in image classification [12,13,14] and object detection [15,16,17,18,19], localizing and tagging objects in images are still challenging due to large intraclass variations, illumination and viewpoint changes, partial occlusions, as well as deformations. In light of these challenges, the sliding window approach (e.g., [3]) has been shown to be one of the more promising approaches towards solving such a *multi-class object layout* problem. The sliding window approach entails the design and use of a suite of trained binary classifiers, each for classifying a specific object class, follow by applying these classifiers to the image with a sliding window approach.

Training a suite of object-specific classifiers is, however, a tedious process that typically requires many training images per class and a lengthy optimization procedure. For this reason also, adding new object class becomes non-trivial. To this end, the

G. Bebis et al. (Eds.): ISVC 2011, Part I, LNCS 6938, pp. 577–589, 2011.

first contribution of this work comes from relaxing this requirement. That is, this work adopts unsupervised clustering to build a dictionary, from which a ranked list of object exemplars, each representing an object class, can be efficiently retrieved from a large database [20,21,22,23,24] when given a test image. By adopting an *unsupervised* learning approach, one immediate advantage is that the training efforts disappear, allowing for the ability to quickly add new object classes to be detected, which simply involves adding the exemplar images of a new class to the database, and running the clustering procedure. Additionally, the unsupervised clustering technique that we have adopted [22] allows for building a dictionary very quickly even with a large database. While not fully evaluated in this work, this ease of adding new object classes and dictionary construction makes the approach scalable with respect to the number of classes that can be represented. It is important to note that the use of such unsupervised clustering need not be restricted to retrieving exemplars, but in our case the retrieved object exemplars would allow one to determine an *approximate* object location simply by matching sliding windows with the retrieved exemplars at runtime.

Given the exemplars and their approximate locations, the final object layout is obtained when the proposed exemplars in error are removed and the final locations of the remaining ones are known. Addressing this issue, which is the second contribution of the paper, is non-trivial because the use of exemplars faces the challenge of intraclass and viewpoint changes, and partial occlusion. A mitigating factor is the representation of each object class with a large number of exemplars [7,25]. We further address this by introducing a regularizing model based on a Conditional Random Field (CRF) [26,27]. At the core of it is a new PCA-based shape representation, called *shape ordering* that drives the CRF conditional distributions. The model can cope with occlusion, false positives, and missing features, and provides a straightforward way to declare an accept or reject decision on the localized object classes based on shape agreement. It is worthwhile to point out that even though we are fully aware of recent state-of-the-art approaches based on modeling and learning contextual object interactions [1,2,28,29], this work leaves that route for future developments, and concentrates here on designing an approach that requires *minimal training and supervision* (unlike work such as [3,29,30]).

The third contribution of the paper is state-of-the-art performance on the PASCAL 2007 dataset [31]. This confirms that it is possible to address intraclass variations and viewpoint changes by representing a class with image exemplars, and by using a CRF to get rid of the resulting sensitivity due to the use of these exemplars, all while avoiding extensive classification and training.

2 Problem Setup

Given a test, or query image I depicting N objects, where the object o belongs to a class c_o, is located at position $\mathbf{p}_o$, and has size s_o in the image, the *object layout* problem aims at identifying the N object classes present in the image, as well as the positions and sizes of the objects.

The proposed approach makes the assumption that it is possible to capture a good chunk of the intraclass and viewpoint variability by collecting a sufficient number of exemplar images containing tightly cropped objects of the same class. After collecting all the exemplars in a database, an unsupervised clustering method [7,21,22,23,24] can be used to build a dictionary from which exemplars similar to the image I can be retrieved. In this work, we use an implementation of the hierarchical vocabulary tree in [22]. This produces a ranked list of M exemplar images. M can be conveniently set to a large enough number, such that there is enough confidence that $M \geq N$.

Rather than being forced to use trained object classifiers during the localization of the objects in the image, the availability of the exemplars is suggesting a minimalistic approach. More precisely, one might attempt to leave with an *approximate* localization, easily obtainable with a form of template matching. Here we rely on a multi-scale sliding window approach where at every scale and image location an image region descriptor is computed, and compared with the corresponding region descriptor of the exemplar. The scale and location corresponding to the best descriptor match constitute the approximate location and size of an object in the image I. Here we have used a fast implementation of the HOG descriptor [16].

After retrieving the exemplar images, a corresponding list of M candidate object classes $\hat{c}_1, \cdots, \hat{c}_M$ is obtained, and after locating where each exemplar might fit within the image I, a list of approximate object positions $\hat{\mathbf{p}}_1, \cdots, \hat{\mathbf{p}}_M$, and sizes $\hat{s}_1, \cdots, \hat{s}_M$ is available. Therefore, we have obtained a candidate object layout $\hat{L}_M = \{(\hat{c}_o, \hat{\mathbf{p}}_o, \hat{s}_o)\}_{o=1,\cdots,M}$.

Obviously, if one of the classes in the test image is not represented by any of the retrieved exemplars, it will never be recovered[1]. Also, the exemplars of interest may not be the first N in the proposed ranking. Needless to say that even N is not known at this point. Finally, even the location of the objects may be rather approximate. All these issues need to be addressed in order to go from the candidate object layout $\hat{L}_M$, to the actual layout $L_N = \{(c_o, \mathbf{p}_o, s_o)\}_{o=1,\cdots,N}$ that is contained by the test image I. To do so, the search for L_N should have been narrowed down to the point that the model in charge of estimating it is constrained enough to (a) suppress the $M - N$ outlier object classes; (b) handle the remaining appearance and shape differences between exemplars and the objects in the image I, due to intraclass and viewpoint variations; (c) compute a refined estimation of the object positions, and scales. All of the above is achieved with a new Conditional Random Field (CRF), which is described in Sec. 3.

2.1 Shape and Appearance Representations

In order to estimate the layout L_N we use a representation for the test image I, and for the candidate exemplar images that is based on SIFT features [32]. In particular, the appearance of I is given by its set of SIFT descriptors $\{f_i\}$, located at image positions $\{\mathbf{f}_i\}$, identified by the SIFT detector. Similarly, the appearance of an exemplar image representing the candidate object o is given by its SIFT descriptors $\{g_u^o\}$, located at the exemplar image positions $\{\mathbf{g}_u^o\}$.

We also introduce a set of planar transformations. More precisely, associated to every candidate object o, the layout L_M carries the information of position $\hat{\mathbf{p}}_o$, and size $\hat{s}_o$ of

[1] Sect. 5 shows how important is to set M big enough for good performance. Conversely, M too big forces the approach to work too hard for rejecting unwanted exemplars, with a reduction in overall performance.

the object in the test image. Based on these parameters, the transformation ϕ_o defines a mapping of the feature locations $\{\mathbf{g}_u^o\}$ from the coordinate system of the exemplar image onto the coordinate system of the test image I.

Given the test image I, represented by $\{(f_i, \mathbf{f}_i)\}$, and given the candidate layout L_M, represented by $\{(g_u^o, \phi_o(\mathbf{g}_u^o))\}_{o=1,\cdots,M}$, the actual layout L_N will be estimated in two steps: 1) Attempt to associate each of the test image features to one of the candidate layout features; 2) Based on the associations above accept or reject outlier object classes, and refine the object positions and sizes. Sec. 3 addresses step 1, whereas Sec. 4 addresses step 2.

3 Conditional Random Field Model

Given the feature representations of the test image and of the candidate layout, we are interested in attempting to match a test image feature f_i, located at $\mathbf{f}_i$, to a feature g_u^o, located at $\mathbf{g}_u^o$, in the exemplar representing the candidate object o. We indicate this matching with the variable assignment $X_i = (o, u)$. The goal then is to estimate the quantity $X \doteq (X_1, X_2, \cdots)$, given the observables $Y \doteq \{(f_i, \mathbf{f}_i)\} \cup \{(g_u^o, \phi_o(\mathbf{g}_u^o))\}_{o=1,\cdots,M}$.

Obviously, one might establish a match $X_i = (o, u)$ by observing that the feature descriptors are similar enough (f_i similar to g_u^o), and that the feature positions are close enough ($\mathbf{f}_i$ close to $\phi_o(\mathbf{g}_u^o)$). Limiting ourselves to such a rule for the matching assignment would leave completely open the possibility that close-by test image features be assigned to close-by exemplar features that belong to different candidate objects. Since these are not our expectations, one would have to apply a form of regularization that involves, for instance, the use of a Markov Random Field (MRF) to model the pairwise matching assignments.

Although certainly beneficial, using a MRF for regularization does not allow to directly exploit the observable data to influence the pairwise assignments. A model that enables that is the Conditional Random Field [26,27] (CRF), here characterized by the following energy function

$$E(X, Y) = \sum_i D(X_i, Y) + \sum_i \sum_{j \in \mathcal{N}_i} W(X_i, X_j, Y) \;, \tag{1}$$

where D and W are the unary and pairwise energy terms. $\mathcal{N}_i$ indicates the neighborhood of X_i, and comprises of all the X_j's such that $|\mathbf{f}_i - \mathbf{f}_j|$ is less than a given threshold.

3.1 Unary Energy Term

This term accounts for the cost of matching f_i at position $\mathbf{f}_i$ in the test image, with the feature g_u^o at position $\mathbf{g}_u^o$ in the exemplar representing the candidate object o. It should penalize a matching assignment between features located far apart and/or dissimilar in appearance. Therefore, the data cost is defined as

$$D(X_i, Y) \doteq \left(1 - e^{-\frac{|\mathbf{f}_i - \phi_o(\mathbf{g}_u^o)|^2}{2\mu^2}}\right) + \nu \Delta(f_i, g_u^o) \;, \tag{2}$$

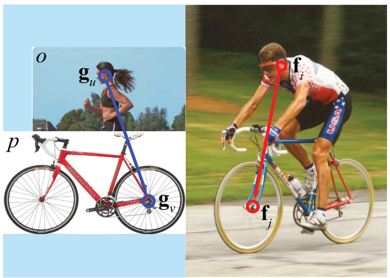

Fig. 1. CRF pairwise assignments. Left image shows the exemplar images of a person head and torso (object o), and of a bicycle (object p). The objects are in the same configuration as they appear in the test image I, on the right. Such configuration is defined by the transformations ϕ_o, and ϕ_p. See also text in Sec. 3.2.

where the constant μ allows to calibrate the spatial weighting of the first term, and ν allows to calibrate the relative weight between the first and the second term. Δ computes the Bhattacharryya distance between f_i and g_u^o. Therefore, the first term introduces penalty based on the distance between $\mathbf{f}_i$ and $\phi_o(\mathbf{g}_u^o)$ (proximity among features). The second term introduces penalty based on the appearance dissimilarity between f_i and g_u^o.

3.2 Pairwise Energy Term

This term accounts for the cost of assigning the matches $X_i = (o, u)$, and $X_j = (p, v)$. This means that f_i is assigned to g_u^o, and f_j to g_v^p. Since the unary term takes care of assigning the matches based on the appearance, this term can simply focus on the shape.

To better understand the meaning of the pairwise term, we consider the example depicted in Fig. 1. Let us consider two features in the test image on the right, f_i and f_j, located at $\mathbf{f}_i$ and $\mathbf{f}_j$. f_i describes the head of a person riding a bike, and f_j describes the center of the front wheel of the bicycle. Let us also consider matching f_i and f_j with two features, g_u^o and g_v^p, from the head of a person exemplar o, and from the center of the rear wheel of a bicycle exemplar p, respectively, which are located at $\mathbf{g}_u^o$ and $\mathbf{g}_v^p$. In such a case, the appearance similarity of the features would tend to assign a match between f_i and g_u^o, and between f_j and g_v^p. Note that this would not be a desirable outcome. To better get a geometric intuition of why this is the case, we draw a segment $\overline{\mathbf{f}_i\mathbf{f}_j}$ from the head to the front wheel in the test image, and a segment $\overline{\mathbf{g}_u^o\mathbf{g}_v^o}$ from the head to the rear wheel in the combined exemplar images on the left (we will later expand on such combination). Clearly, these two segments are not representative of homologous regions of the two compound person-bicycle objects. Therefore, they should not be matched. The pairwise energy is meant to achieve exactly this goal, through the use of the idea of *shape ordering*, which we now introduce.

Shape Ordering. Given a point cloud on the plane, let us consider two specific points, $\mathbf{a}$ and $\mathbf{b}$. The shape ordering of the pair $(\mathbf{a}, \mathbf{b})$ establishes an ordering of such points with respect to the cloud. Such cloud might be representative for the shape of an imaged object. In particular, let us consider the cloud of points $\{\mathbf{g}_u^o\}$, given by the feature locations of the exemplar image representing object o. With respect to this cloud, the *shape ordering* of the pair $(\mathbf{g}_u^o, \mathbf{g}_v^o)$ is a 2×1 matrix, computed according to the following expression

$$\sigma_{uv}^o \doteq \operatorname{sign}\left(V^{oT}(\mathbf{g}_v^o - \mathbf{g}_u^o)\right) \ , \tag{3}$$

where $V^o = [v_1^o, v_2^o] \in \mathbb{R}^{2\times 2}$ are the principal components of the point cloud $\{\mathbf{g}_u^o\}$, and sign is the usual function that is $+1$, 0, or -1 when the argument is positive, 0, or negative, respectively.

To understand the meaning of σ_{uv}^o, let us focus on its first component, and the projections of $\mathbf{g}_u^o$ and $\mathbf{g}_v^o$ onto the first principal axis v_1^o. If the first component is $+1$ then $\mathbf{g}_u^o$ precedes $\mathbf{g}_v^o$ along v_1^o. We indicate this fact with the notation $\mathbf{g}_u^o \prec_1 \mathbf{g}_v^o$. Similarly, if the first component is -1 means that $\mathbf{g}_u^o$ succeeds $\mathbf{g}_v^o$ along v_1^o (i.e. $\mathbf{g}_u^o \succ_1 \mathbf{g}_v^o$). Finally, if the first component is 0 means that $\mathbf{g}_u^o$ and $\mathbf{g}_v^o$ are the same when projected onto v_1^o (i.e. $\mathbf{g}_u^o =_1 \mathbf{g}_v^0$). Analogous reasoning holds with respect to the second component of the shape ordering. More precisely we have $+1 \Leftrightarrow \mathbf{g}_u^o \prec_2 \mathbf{g}_v^o$, $-1 \Leftrightarrow \mathbf{g}_u^o \succ_2 \mathbf{g}_v^o$, and $0 \Leftrightarrow \mathbf{g}_u^0 =_2 \mathbf{g}_v^o$.

When a pair of feature locations $(\mathbf{g}_u^o, \mathbf{g}_v^p)$, belongs to two different candidate objects, o and p, the shape ordering is defined as follows. The objects are related according to the candidate layout geometry, defined by the transformations ϕ_o, and ϕ_p, therefore, the reference point cloud becomes $\{\mathbf{g}_u^o\} \cup \phi_o^{-1}(\phi_p(\{\mathbf{g}_v^p\}))$. If V^{op} represents the principal components of such cloud, then the shape ordering is computed as $\sigma_{uv}^{op} \doteq \operatorname{sign}\left(V^{opT}(\phi_o^{-1}(\phi_p(\mathbf{g}_v^p)) - \mathbf{g}_u^o)\right]$.

The idea of shape ordering can be applied to the feature locations of the test image as well, and it is done in this way. First, we need to identify the point cloud with respect to which we want to compute the ordering. This is done by picking a candidate object o, and by using the transformation ϕ_o that maps the borders of the exemplar image into a bounding box B_o, in the test image I. Therefore, the shape ordering σ_{ij}^o of two feature locations $(\mathbf{f}_i, \mathbf{f}_j)$, is computed with respect to the point cloud given by the feature locations inside the bounding box B_o, i.e $\{\mathbf{f}_i | \mathbf{f}_i \subset B_o\}$. The expression is given by $\sigma_{ij}^o \doteq \operatorname{sign}\left(U^{oT}(\mathbf{f}_i - \mathbf{f}_j)\right)$, where U^o are the principal components of the set $\{\mathbf{f}_i | \mathbf{f}_i \subset B_o\}$.

The shape ordering of the pair of feature locations $(\mathbf{f}_i, \mathbf{f}_j)$ can be computed also with respect to a point cloud induced by multiple candidate objects. The case of two objects, o and p, is of particular interest to us. In this case the reference cloud is given by the feature locations that are either inside B_o or B_p, i.e. $\{\mathbf{f}_i | \mathbf{f}_i \subset B_o \cup B_p\}$. If U^{op} represents the principal components of such cloud, then the shape ordering is computed as $\sigma_{ij}^{op} \doteq \operatorname{sign}\left(U^{opT}(\mathbf{f}_i - \mathbf{f}_j)\right)$.

Mutual Shape Ordering. Going back to the example of Fig. 1, let us consider the cloud of SIFT locations from the person exemplar $\{\mathbf{g}_u^o\}$, and from the bicycle exemplar $\{\mathbf{g}_v^p\}$, and further assume that their candidate positions in the test image are defined through ϕ_o, and ϕ_p. The shape ordering of the pair of features $(\mathbf{g}_u^o, \mathbf{g}_v^p)$, located on the head, and on the center of the rear wheel is σ_{uv}^{op}. At the same time, let us consider the point cloud defined by the test image feature locations inside B_o and B_p. The shape ordering of the pair $(\mathbf{f}_i, \mathbf{f}_j)$, located on the head, and on the center of the front wheel is σ_{ij}^{op}. If the two clouds are coming from the same type of object, and if the segments $\overline{\mathbf{f}_i \mathbf{f}_j}$ and $\overline{\mathbf{g}_u^o \mathbf{g}_v^o}$ represent homologous regions of the point cloud we should expect similar shape orderings. The *mutual shape ordering* aims at measuring this similarity by computing $\delta(\sigma_{ij}^{op}, \sigma_{uv}^{op})$, which is simply the Kronecker delta between the first, and between the second components of σ_{ij}^{op} and σ_{uv}^{op}. Therefore, when $\delta(\sigma_{ij}^{op}, \sigma_{uv}^{op}) = [0, 0]^T$ there is maximum shape agreement, and when $\delta(\sigma_{ij}^{op}, \sigma_{uv}^{op}) = [1, 1]^T$ there is minimum agreement.

Energy Term. Finally, in order to assign the matches $X_i = (o, u)$, and $X_j = (p, v)$ (i.e. f_i is assigned to g_u^o, and f_j to g_v^p), the pairwise energy term is defined as

$$W(X_i, X_j, Y) \doteq 1 - e^{-\beta |(\mathbf{f}_j - \mathbf{f}_i)^T U^{op} | \delta(\sigma_{ij}^{op}, \sigma_{uv}^{op})} , \tag{4}$$

where $\beta > 0$ is a scaling factor, and the notation $|\cdot|$ in the exponent is intended to take the absolute values of each of the components of the vector inside. Essentially, we are penalizing mismatch in the mutual shape ordering by the amount of distance that has to be made up between $\mathbf{f}_i$ and $\mathbf{f}_j$ on the principal axes in order to obtain shape ordering agreement. Inference on the CRF is done with the Belief Propagation approach described in [33].

Notice that the shape ordering and the mutual shape ordering are quantities that are very robust to noise (see also Fig. 2). This is a very desirable property that enables the model to handle a fair amount of variations in shape due to intraclass variations and viewpoint changes.

4 Final Object Layout

Given the matching assignments we now wish to reject the $M - N$ outlier objects to obtain the final layout. For a given object o, let us consider the set of matches $\{\mathbf{f}_i \leftrightarrow \mathbf{g}_u^o\}$, such that every $\mathbf{f}_i$ is located inside B_o. In order to establish whether o is an outlier, from the set of matches we build two sets: one containing only the feature locations in the test image, and the other one containing only the feature locations in the exemplar object. By using the shape ordering within each set as a comparison criterion between a pair of feature locations, it is possible to comparison sort the feature locations of each set along each of the principal axis. This leads to the formation of four ordered sequences F_1, F_2, G_1, G_2. For example, with reference to Fig. 2: $F_1 = \mathbf{f}_7 \prec_1 \mathbf{f}_5 \prec_1 \mathbf{f}_8 \prec_1 \mathbf{f}_6 \prec_1 \mathbf{f}_2 \prec_1 \mathbf{f}_4 \prec_1 \mathbf{f}_3 \prec_1 \mathbf{f}_1$; $F_2 = \mathbf{f}_5 \prec_2 \mathbf{f}_2 \prec_2 \mathbf{f}_3 \prec_2 \mathbf{f}_6 \prec_2 \mathbf{f}_1 \prec_2 \mathbf{f}_4 \prec_2 \mathbf{f}_7 \prec_2 \mathbf{f}_8$; $G_1 = \mathbf{g}_7 \prec_1 \mathbf{g}_5 \prec_1 \mathbf{g}_8 \prec_1 \mathbf{g}_6 \prec_1 \mathbf{g}_3 \prec_1 \mathbf{g}_4 \prec_1 \mathbf{g}_2 \prec_1 \mathbf{g}_1$; $G_2 = \mathbf{g}_5 \prec_2 \mathbf{g}_3 \prec_2 \mathbf{g}_2 \prec_2 \mathbf{g}_6 \prec_2 \mathbf{g}_4 \prec_2 \mathbf{g}_7 \prec_2 \mathbf{g}_1 \prec_2 \mathbf{g}_8$. We exploit these four sequences to perform a final rejection of the matches that do not show full shape agreement, and/or might come from background clutter. This is done by computing the longest increasing subsequence [34] between F_1 and G_1, and between F_2 and G_2. In our example they become $7 \prec_1 5 \prec_1 8 \prec_1 6 \prec_1 3 \prec_1 1$, and $5 \prec_2 3 \prec_2 6 \prec_2 4 \prec_2 7 \prec_2 8$, respectively. The intersection between the two resulting subsequences yields the final set of estimated matches. In our example this is given by $\mathcal{M} \doteq \{\mathbf{f}_7 \leftrightarrow \mathbf{g}_7, \mathbf{f}_5 \leftrightarrow \mathbf{g}_5, \mathbf{f}_8 \leftrightarrow \mathbf{g}_8, \mathbf{f}_6 \leftrightarrow \mathbf{g}_6, \mathbf{f}_3 \leftrightarrow \mathbf{g}_3\}$. Finally, we say that object o is part of the layout L_N if the number of matches $|\mathcal{M}| \geq \gamma$,

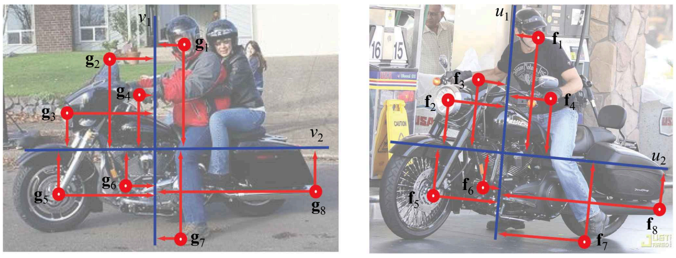

Fig. 2. Shape ordering. Illustration of the notion of shape ordering with a synthetic example. The left image depicts the exemplar of an object o. v_1 and v_2 are the principal axes for o. The right image shows only the portion of the test image inside the bounding box B_o. u_1 and u_2 are the principal axes of the test image features located inside B_o, and the indices indicate matches (e.g., $\mathbf{g}_1$ would be matched with $\mathbf{f}_1$). For the pair $(\mathbf{g}_4, \mathbf{g}_8)$ the shape ordering is $\sigma^o_{48} = [-1, +1]^T$. Similarly, the corresponding features in the test image $(\mathbf{f}_4, \mathbf{f}_8)$ have the same ordering $[-1, +1]^T$. Therefore, the mutual shape ordering between them is a perfect match of $[0, 0]^T$. This is despite that the motorbikes are taken from different viewpoints, illustrating how the shape ordering can tolerate a certain amount of viewpoint and intraclass variation.

where γ is a pre-specified required number of matches. While the optimal value of γ can certainly be obtained by training, we have found that setting γ between $8-10$ yields very good performance. This can be understood by the fact that a sufficient number of matches represents a fairly low possibility of shape disagreement in rigid objects. For a good illustration of such a procedure, refer to Fig. 2, which purposely contains two "slight" mismatches, i.e., $\mathbf{f}_2 \leftrightarrow \mathbf{g}_2$, and $\mathbf{f}_3 \leftrightarrow \mathbf{g}_3$, of which one can be filtered by the procedure.

Lastly, we fit a 2D affine transformation to the final set of matches $\mathcal{M}$, mapping the features from the exemplar domain to the test image domain. This allows to refine the estimation of the position $\mathbf{p}_o$, and size s_o, of the object o.

5 Experiments

In order to be comparative, we evaluate our approach on the PASCAL Visual Object Challenge (VOC) 2007 dataset [31], on which many state-of-the-art methods have been evaluated on. These include in particular the work in [2], which tries to solve the same multi-class object layout problem by modeling the object interactions.

The PASCAL 2007 dataset is widely agreed to be a very challenging dataset, consisting of about 10000 images of 20 different object classes. Occlusions are common in these images, making it especially challenging. Roughly half of the images are used for training while the other half are used for testing. For the training set, there are approximately 16000 objects in about 5000 images. We generate exemplars out of them by cropping out the 16000 objects from the images, based on the bounding box annotations that came along with the PASCAL dataset, and added them to our exemplar

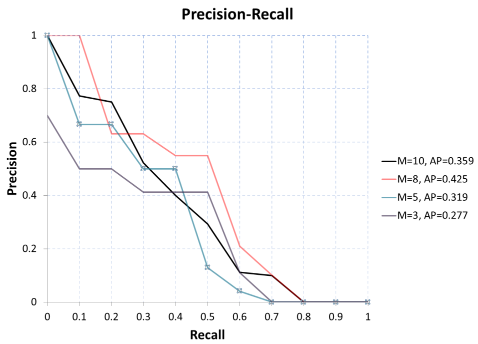

Fig. 3. Multi-class results. PR curves for four different runs with different number, M, of retrieved exemplar objects. Our AP scores of 0.277, 0.319, 0.425 and 0.359 for $M = 3, 5, 8, 10$ outperform that of [2], which reports an AP of 0.272.

Table 1. Per-class results. The best per-class results among our approach, the approach in [2], and the approach in [3] are in bold. The mAP of our approach shows significant improvement. In general, it can be observed that object classes with reach shape context, as described by the shape ordering descriptor, do very well.

Class	Approach [3]	Approach [2]	Our approach
Aeroplane	0.338	0.288	**0.508**
Bicycle	0.430	**0.562**	0.410
Bird	0.097	0.032	**0.560**
Boat	0.096	0.142	**0.333**
Bottle	0.187	0.294	**0.395**
Bus	**0.419**	0.387	0.310
Car	0.504	0.487	**0.625**
Cat	0.150	0.124	**0.250**
Chair	0.146	0.160	**0.180**
Cow	0.239	0.177	**0.367**
Diningtable	0.151	**0.240**	0.227
Dog	0.154	0.117	**0.273**
Horse	0.482	0.450	**0.667**
Motorbike	**0.417**	0.394	0.381
Person	0.202	0.355	**0.520**
Pottedplant	0.161	0.152	**0.537**
Sheep	0.212	0.161	**0.581**
Sofa	0.203	0.201	**0.300**
Train	0.291	0.342	**0.506**
Tvmonitor	0.382	0.354	**0.455**
mAP	0.263	0.253	**0.419**

database. Following the PASCAL evaluation protocol, a detection is considered correct if its bounding box overlaps 50% or more with the groundtruth bounding box.

Multi-Class Results. In four different runs, we retrieved $M = 3, 5, 8, 10$ objects from the database, based on the vocabulary tree. Together with the bounding boxes localized on the test image, they form the candidate object layout on which inference is performed.

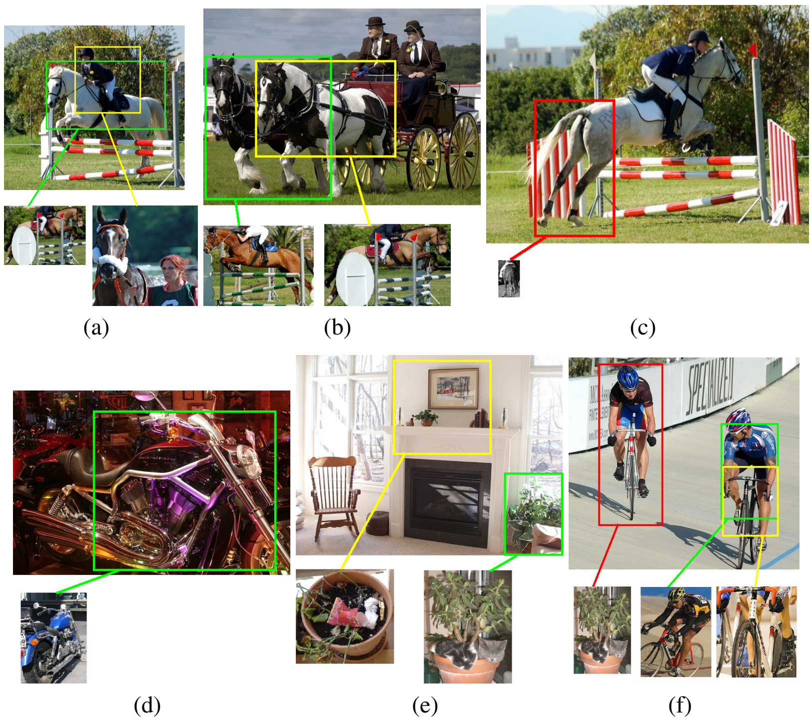

Fig. 4. Qualitative evaluation. In each subfigure, the upper image is the test image while the lower images are the retrieved exemplars. **(a)** The first horse exemplar retrieved shows the right side of a horse but was used to successfully detect the left side of a horse in the test image. The SIFT descriptors used during classification are invariant to such "geometrical flip", which means that for a somewhat symmetrical object such as a horse, where the left and right side are similar to each other, both sides could have been retrieved. Even so, the shape ordering descriptor, being isotropic to geometrical flip, was able to do the right thing. In the second exemplar retrieved, the cropping, meant to generate a horse exemplar, has to include the whole horse, and consequently the person in a frontal pose. The person is actually successfully matched with the rider in the test image. **(b)** Another example of the shape ordering descriptor's invariance to geometrical flip in the detection of the horses. The people in the test image were not detected, constituting false negatives. **(c)** To what degree can the shape ordering descriptor deal with viewpoint variations? Here, the retrieved exemplar is the backside of a horse, which proves to have too much viewpoint variation with respect to the right view of the backside of the horse in the test image, triggering the CRF to declare a no-match decision. **(d)** The viewpoint variation was overcome by the CRF, and with the bounding box intersecting more than 50% of the groundtruth bounding box, this was declared a true positive. **(e)** Although the localized bounding box for the first plant exemplar was too big (shown as the yellow bounding box in the test image before refinement), the CRF was able to discard feature matchings that do not agree with the shape of the exemplar, and correctly produce a true positive. The second exemplar was used to correctly detect the plant in the lower right corner of the test image. This is in spite of the plant in the test image being partially occluded because the shape ordering descriptors agree enough to trigger a true positive. **(f)** This is probably one of the best demonstrations of how a good exemplar retrieved, together with good localization, make it fairly straightforward for the CRF to declare correctly a true positive for the rider and bike on the right side of the test image. Unfortunately, the shape of the rider on the left side of the test image was confused with a plant exemplar.

For each run, we follow the multi-class scoring methodology in [2] by pooling detections across classes, and computing the Average Precision (AP) according to the VOC protocol. We show the precision-recall (PR) curves in Fig. 3, with AP scores of 0.277, 0.319, 0.425 and 0.359 for each of the four runs respectively. For the PASCAL 2007 dataset, our algorithm peaks at $M = 8$ with an AP of 0.425, which is a great improvement over an AP of 0.272 for the multi-class experiments conducted in [2].

Per-Class Results. We also provide the per-class AP results in Table. 1. Our results show very good performance with all but four classes lower in AP than [2]. The mean Average Precision (mAP) over these classes is 0.419, which is a considerable improvement over an mAP of 0.253, achieved by [2].

Moreover, these results help to answer the question of how relevant the CRF is on top of the image classification and localization stage. We first compare our results with those reported in [3]. Our mAP of 0.419 is a significant improvement over the best mAP result of approximately 0.263 obtained over a different number of sliding windows by the work in [3]. Contrasting our results with those reported in [3] is important. The reason is that [3], being a state-of-the-art sliding window approach based on detectors trained on HOG and appearance features, bears resemblance to our HOG based localization stage. Similarly, we borrow the per-class results reported by [35] on the PASCAL 2007 dataset for the same unsupervised image classification technique [22] we use. The best reported mAP is approximately 0.3, which represents the performance of a detector that was trained based on descriptors extracted from a hierarchical vocabulary tree. This is in contrast to the mAP of 0.419 our approach is able to achieve.

In both comparisons, even though the results in those works pertain to trained detectors, they give a good indication that the CRF is able to boost performance significantly on top of the image classification and localization stage.

Qualitative Evaluation. Finally, we show and explain some of the interesting detection results qualitatively. Fig. 4 shows different examples that are meant to illustrate the following: (i) the robustness of the shape ordering descriptor to viewpoint variations, (ii) what might be too much viewpoint variation for the shape ordering descriptor, (iii) how tolerant is the CRF model to inaccuracy in the localization, and (iv) cases of false positives and false negatives, due also to shape confusion. Please refer to the caption of Fig. 4 for details.

6 Conclusions

This work proposes an alternative solution to the object classification and localization problem. The solution avoids the effort involves in training a suite of object classifiers by adopting unsupervised clustering for building a dictionary from which proposed object classes given a test image can be retrieved efficiently. The proposed object classes, together with their locations, are evaluated robustly with a CRF model that is based on a newly proposed shape ordering. As a consequence of utilizing such a shape ordering, it becomes fairly straightforward to decide whether to accept or reject the list of proposed object locations.

The resulting approach has minimum training requirements, and is able to produce a remarkable performance improvement on the challenging PASCAL 2007 dataset, showing that it is able to cope effectively with very large intraclass and viewpoint changes. This confirms the utility of such approach for practical purposes.

Acknowledgements. This report was prepared by GE GRC as an account of work sponsored by Lockheed Martin Corporation. Information contained in this report constitutes technical information which is the property of Lockheed Martin Corporation. Neither GE nor Lockheed Martin Corporation, nor any person acting on behalf of either; a. Makes any warranty or representation, expressed or implied, with respect to the use of any information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or b. Assume any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method or process disclosed in this report.

References

1. Choi, M., Lim, J., Torralba, A., Willsky, A.: Exploiting hierarchical context on a large database of object categories. In: CVPR (2010)
2. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV, pp. 229–236 (2009)
3. Harzallah, H., Jurie, F., Schmid, C.: Combining efficient object localization and image classification. In: ICCV, pp. 237–244 (2009)
4. Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 30–43. Springer, Heidelberg (2008)
5. Ladicky, L., Sturgess, P., Alahari, K., Russel, C., Torr, P.H.S.: What, where and how many? Combining object detectors and cRFs. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 424–437. Springer, Heidelberg (2010)
6. Li, L.J., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: ICCV (2007)
7. Lim, S.N., Doretto, G., Rittscher, J.: Object constellations: Scalable, simultaneous detection and recognition of multiple specific objects. In: Workshop on Cognitive Vision in conjunction with ECCV (2010)
8. Mutch, J., Lowe, D.: Object class recognition and localization using sparse features with limited receptive fields. IJCV 80 (2008)
9. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
10. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: CVPR (2010)
11. Yeh, T., Lee, J., Darrell, T.: Fast concurrent object localization and recognition. In: CVPR, pp. 280–287 (2009)
12. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)
13. Perronin, F.: Universal and adapted vocabularies for generic visual categorization. IEEE TPAMI 30, 1243–1256 (2008)
14. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. IJCV 73, 213–238 (2007)

15. Bourdev, L., Maji, S., Brox, T., Malik, J.: Detecting people using mutually consistent poselet activations. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 168–181. Springer, Heidelberg (2010)
16. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: ICCV, vol. 1, pp. 886–893 (2005)
17. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE TPAMI 32, 1627–1645 (2010)
18. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on riemannian manifolds. In: CVPR, pp. 1–8 (2007)
19. Viola, P., Jones, M.J.: Robust real-time face detection. IJCV 57, 137–154 (2004)
20. Berg, T., Forsyth, D.: Animals on the web. In: CVPR (2006)
21. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
22. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR, vol. 2, pp. 2161–2168 (2006)
23. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
24. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR (2008)
25. Gammeter, S., Bossard, L., Quack, T., van Gool, L.: I know what you did last summer: Object-level auto-annotation of holiday snaps. In: ICCV, pp. 614–621 (2009)
26. Kumar, S., Hebert, M.: Discriminative random fields. IJCV 68, 179–202 (2006)
27. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML, pp. 282–289 (2001)
28. Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A.A., Hebert, M.: An empirical study of context in object detection. In: CVPR (2009)
29. Galleguillos, C., McFee, B., Belongie, S., Lanckriet, G.: Multi-class object localization by combining local contextual interactions. In: CVPR, pp. 113–120 (2010)
30. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR (2008)
31. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results (2007), `http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html`
32. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
33. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. IJCV 70, 41–54 (2006)
34. Fredman, M.L.: On computing the length of the longest increasing subsequences. Discrete Mathematics 11, 29–35 (1975)
35. Bengio, S., Pereira, F., Singer, Y., Strelow, D.: Group sparse coding. In: NIPS (2009)