Image Retrieval with Structured Object Queries Using Latent Ranking SVM

Tian Lan[1], Weilong Yang[1], Yang Wang[2], and Greg Mori[1]

[1] Simon Fraser University
[2] University of Manitoba
{tla58,wya16,mori}@cs.sfu.ca, ywang@cs.umanitoba.ca

Abstract. We consider image retrieval with *structured object queries* – queries that specify the objects that should be present in the scene, and their spatial relations. An example of such queries is "car on the road". Existing image retrieval systems typically consider queries consisting of object classes (i.e. keywords). They train a separate classifier for each object class and combine the output heuristically. In contrast, we develop a learning framework to jointly consider object classes and their relations. Our method considers not only the objects in the query ("car" and "road" in the above example), but also related object categories can be useful for retrieval. Since we do not have ground-truth labeling of object bounding boxes on the test image, we represent them as latent variables in our model. Our learning method is an extension of the ranking SVM with latent variables, which we call *latent ranking SVM*. We demonstrate image retrieval and ranking results on a dataset with more than a hundred of object classes.

1 Introduction

How can we do image retrieval with complex queries that involve objects in certain relations, such as "car on road"? Most current keyword-based image retrieval systems will simply search for images containing both "car" and "road" while ignoring their relations (in this case "on"). In this paper we present an approach for retrieving images for *structured object queries* – queries specifying the objects that should be in a scene, and their relations. As an example, Fig. 1 illustrates retrieval from a query asking for an image containing a house, trees, with a road beside grass, and showing sky above trees.

In image understanding and retrieval, there are two related, but inverse grand goals. (1) Image to sentence: given an image, generate sentences describing the content of the image. (2) Sentence to image: given a sentence, retrieve images with content relevant to the sentence. Both tasks are extremely challenging. In the computer vision literature, researchers have tried to simplify the problem by replacing "sentences" with "words", in particular words corresponding to object names. Standard object classification/detection can be seen as the simplest example of generating a word description for an image. Beyond classification/detection, the "words and pictures" work of Barnard et al. [1] developed probabilistic models

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part VI, LNCS 7577, pp. 129–142, 2012.

for matching image regions with words. These models could facilitate both grand goals, though at a word (more specifically, object category) level.

Recent research in image understanding and retrieval has attempted to step past an object-level representation. Li and Fei-Fei [2] develop generative models of images for queries involving *what, where, and who* – events, scenes and objects. Gupta et al. [3] use the AND-OR graph formalism to represent spatio-temporal relations among objects and actions in videos. Sadeghi and Farhadi [4] examine the scale of unit at which to categorize objects, and develop a notion of visual phrases for jointly recognizing co-occurring objects. Farhadi et al. [5] develop image models that indicate the presence of *object, action, scene* triplets. These can again be used both for image retrieval and generating short descriptive sentences given an image. Kulkarni et al. [6] push this work a step further by generating more complex, natural language descriptions which are able to describe multiple objects, their attributes, and their spatial relations. Our work can be considered as a reverse process of this line of work. Instead of generating a complex descriptive sentence, users are asked to provide a set of short phrases specifying several objects and their relations. Our algorithm learns to retrieve relevant images according to these phrases, taking into account spatial relations of objects.

There are numerous challenges in developing such an algorithm. For instance, some objects are easier to detect than others – sky or grass detectors tend to work well compared to chair or lamp detectors. On the other hand, some of these objects might be very frequent – sky and grass might appear in many images. Further, the relations between these objects, and the ability to localize them accurately, will vary over different pairs of objects. We employ a latent variable ranking learning model in an attempt to address these challenges. Our learning framework will attempt to learn which object detectors and relations are reliable for image retrieval with structured object queries.

Image retrieval is a large, active area of research. Datta et al. [7] provide a recent survey. Learning to rank has proven extremely successful in retrieval, where a ranking function is learnt given different relevance levels of the training data with respect to the query. Our work follows in the ranking SVM [8] formalism, but extends it with latent variables [9,10]. One limitation of most previous approaches is that they learn a separate ranking function for each query term (single word), and combine the output of single-word queries heuristically for retrieving multi-word queries. Recently, Siddiquie et al. [11] built a framework for multi-attribute retrieval which models the co-occurrence between the attributes. In contrast, our work considers a more detailed phrase-level representation of queries, and explicitly models the spatial layout of objects. Moreover, we do not assume the existence of reliable detectors as in [11]. Instead, the object locations are treated as latent variables that are implicitly inferred simultaneously with image retrieval.

We highlight the main contributions of this paper. 1) We consider complex and descriptive structured object queries. There is work (e.g. [12,13]) that considers complex relations in object detection. This paper is the first that attempts to

address the inverse problem – given a query involving objects and their relations, retrieve images that are relevant. 2) We develop a latent variable framework for image retrieval that integrates structured prediction into a ranking model. We demonstrate experimentally that the rich structures explicitly given by queries as well as those inherent in images are important for the image retrieval task.

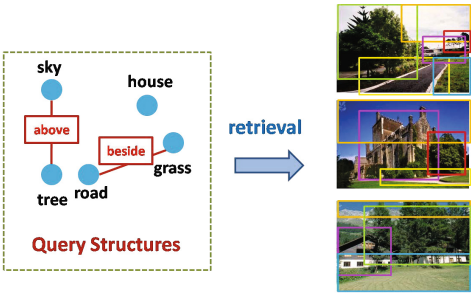

Fig. 1. An example of image retrieval with structured object queries. A structured object query is represented as a graph $\mathcal{G} = \{V_\mathcal{G}, E_\mathcal{G}\}$. The vertices $V_\mathcal{G}$ in the graph denote object classes and the edges $E_\mathcal{G}$ denote spatial relations between pairs of objects. Here the query is $\{house, sky - above - tree, road - beside - grass\}$. The goal of this paper is to explore the rich structures in the queries to retrieve the relevant images the query is asking for while localizing the objects specified by the query.

2 Structured Object Query Model

Here we describe our approach for image retrieval with structured object queries. We are given an object vocabulary $\mathcal{X}$ consisting of K object classes, i.e. $|\mathcal{X}| = K$. A structured object query consists of a subset of object classes, and the relations between certain pairs of objects. Figure 1 shows an example of structured object queries. The query involves four objects (house, sky, road, grass) and two relations ("above" relation between sky and tree, "beside" relation between road and grass). We can represent a structured object query as a graph $\mathcal{G} = \{V_\mathcal{G}, E_\mathcal{G}\}$. The vertices $V_\mathcal{G}$ in the graph denote object classes and the edges $E_\mathcal{G}$ denote spatial relations between pairs of objects, as shown in Fig. 1. Our goal is to learn a retrieval model that jointly consider the objects and the relations mentioned in the query, as well as objects not mentioned in the query but can be useful for retrieval.

At the training stage, we are given a set of training images $\mathcal{I}_{train} = \{I_1, I_2, ..., I_N\}$. The ground-truth bounding boxes for all the objects are available on the training images. During testing, we are given a new set $\mathcal{I}_{test}$ of images without object bounding boxes. For a given query $\mathcal{Q}$, our goal is to predict a subset Y from the test image set $\mathcal{I}_{test}$ (i.e. $Y \subset \mathcal{I}_{test}$) such that Y is relevant to the query $\mathcal{Q}$. Different from most previous image retrieval methods, we build a model that retrieves images based on not only objects in the query, but also considers remaining objects in $\mathcal{X}$ that are not in the query. For example, consider the query "car on road". Both "car" and "road" are objects in the query. A

standard image retrieval method will try to retrieve images in which both "car" and "road" detectors have high responses. The limitation of this approach is that other objects not in the query can often provide additional useful information. For example, if an image has high responses from a "pedestrian" detector, it is more likely to be relevant to the query, even though "pedestrian" is not an object in the query. This is because pedestrians tend to appear together with cars and roads. On the other hand, a high response from a "sofa" detector might suggest that an image is less likely to be relevant to this query, since sofas usually do not co-occur with cars/roads. To retrieve images given a query, our model considers the evidence from all object detectors, not just detectors for objects in the query. Similar ideas have been used in [11] for multi-attribute image retrieval. The difference from [11] is that, in addition to object names, our query also considers certain relations between objects. In the previous example "car on road", we are not just interested in images containing cars and roads. Instead we also prefer a specific relation (i.e. "on") between these two objects.

Our model is motivated by the following observations. In many image retrieval applications, object names are not enough to accurately capture users' intentions. For example, architects might want to retrieve images containing buildings and roads in some specific layout. It is not enough to retrieve images containing the objects ("building" and "road"). Instead, we need to incorporate the relation between "building" and "road" into the query. Chen et al.[14] address this issue by allowing users to draw a sketch as the query. However this form of query might not be suitable for users with little practice in drawing. In this paper, we introduce *structured object queries*. In addition to encoding object names as standard keyword-based image retrieval systems, our method also encodes certain relations among objects. Compared with sketches as queries, the structured object queries are easier to use since they do not require any drawing skills from users. For example, a structured object query can be $\mathcal{Q} = \{$"car on road"$\}$. See Fig. 2 for more examples of structured queries.

Fig. 2. Some examples of structured queries. Note that the queries involve not only object classes (such as "sky", "building", etc.), but also the relations (such as "above", "on", etc.) between certain pairs of objects.

The key of our method is to learn a real-valued function $H_\Theta(I, \mathcal{Q})$ (here Θ are parameters of this function) that measures the compatibility of an image I and a query $\mathcal{Q}$. Ideally $H_\Theta(I, \mathcal{Q})$ will have a high value if the image I is relevant to the query $\mathcal{Q}$. If the image I comes with ground-truth object bounding boxes, $H_\Theta(I, \mathcal{Q})$ will be trivial to define. We can simply examine the object bounding boxes and assign a high value if they are consistent with the query $\mathcal{Q}$. But the challenge is that we do not have access to ground-truth bounding boxes on the test image set. One possible solution is to run all the object detectors on the test image set and pretend the detector outputs to be ground-truth. But this is not a reliable solution unless we have near perfect detectors for all the objects in $\mathcal{X}$.

Instead of relying on ground-truth object bounding boxes, we treat the bounding boxes for all the objects as latent variables and infer them implicitly in our model. For an image I, we assume it is associated with a "hypothesized" configuration of all object classes. We denote this "hypothesized" configuration as $L = (l_1, l_2, ..., l_K)$, where l_i represents the location and scale of object i. Note that we only consider one instance of an object. For example, if $i =$ "car'' we assume l_i only encodes one location/scale of cars in the image. For some queries where an object class appears multiple times, e.g. {"car-beside-car"}, then i and j will represent the same object class (car), and we do not constrain l_i and l_j to the same location/scale. Also note that we are considering all object classes in an image, not just those in the query. This is in sharp contrast with standard object detection tasks. In standard object detection, an object class with low scores simply means this object class does not appear in an image. But for image retrieval tasks, as we mentioned earlier, an object class with low scores actually can provide useful information on whether this image is relevant to a query. So in our representation, we assume every object has a "hypothesized" location/scale in an image. We will build a model that consolidate the detector scores from all object classes for image retrieval.

Given a tuple $(I, \mathcal{Q}, L)$ consisting of an image I, a query $\mathcal{Q}$, and a putative object configuration L, we define a real-valued function F_Θ (Θ are the parameters of this function) that scores the tuple. Then the score $H_\Theta(I, \mathcal{Q})$ between an image I and a query $\mathcal{Q}$ can be obtained by maximizing over L (similar to latent SVM [15]): $H_\Theta(I, \mathcal{Q}) = \underset{L}{\text{argmax}} F_\Theta(L, \mathcal{Q}, I)$. The score $H_\Theta(Y, \mathcal{Q})$ between a query $\mathcal{Q}$ and an image set Y can be defined as the cumulative score of all the images in Y, i.e. $H_\Theta(Y, \mathcal{Q}) = \sum_{I \in Y} H_\Theta(I, \mathcal{Q})$.

For a given query $\mathcal{Q}$, the optimal image set Y^* can be obtained by choosing the image set with the maximum score:

$$Y^* = \underset{Y \subset \mathcal{Y}}{\text{argmax}} H_\Theta(Y, \mathcal{Q}) = \underset{Y \subset \mathcal{Y}}{\text{argmax}} \Big(\sum_{I \in Y} H_\Theta(I, \mathcal{Q}) \Big) \tag{1}$$

We assume the model parameters have five components $\Theta = \{\alpha, \beta, \gamma, \eta, \zeta\}$ and define $F_\Theta(L, \mathcal{Q}, I)$ as follows:

$$F_\Theta(\mathcal{Q}, L, I) = \Theta^\top \Psi(\mathcal{Q}, L, I) \tag{2a}$$

$$= \alpha^\top \phi(\mathcal{Q}, L, I) + \beta^\top \psi(\mathcal{Q}, L, I) + \gamma^\top \omega(\mathcal{Q}, L, I) + \eta^\top \varphi(\mathcal{Q}, L, I) + \zeta^\top \Omega(\mathcal{Q}, I) \tag{2b}$$

In the following, we describe in detail each term in Eq. 2.

Query Object Detector Model $\alpha^\top \phi(\mathcal{Q}, L, I)$: For an object label i, we can imagine α_i to be a template for detecting object i in an image. For example, if an image patch is represented by HOG descriptors, α_i can be the weights associated with each HOG cell which are obtained from a linear SVM classifier. Then we parameterize this potential function as:

$$\alpha^\top \phi(\mathcal{Q}, L, I) = \sum_{i \in V_\mathcal{Q}} \alpha_i^\top f(I(l_i)) \tag{3}$$

Here $f(I(l_i))$ is a feature vector extracted at location/scale l_i of the image I. The parameters α are simply concatenations of $\{\alpha_i : \forall i \in \mathcal{X}\}$. Notice that if the query $\mathcal{Q}$ does not involve an object label k (i.e. $k \notin V_\mathcal{Q}$), the detector for object k will not appear in Eq. 3. The intuition behind Eq. 3 is as follows. If the detectors for those objects in the query have high responses at certain location/scale in an image, Eq. 3 will have a large value which in turn means the image is more likely to be relevant to the query. Here we adopt a simpler approach by setting $f(I(l_i))$ to be the output score of an independently trained object detector.

Query Object Spatial Relation Model $\beta^\top \psi(\mathcal{Q}, L, I)$: This potential function captures the importance of the spatial relation between certain pairs of objects defined in the query. Fig. 1 shows a typical query structure, with nodes representing the object classes and edges representing the spatial relations between certain pairs of object classes. Here we consider 3 different spatial relations: *above*, *below* and *overlap*. Given two object labels (i, j) and their configurations (l_i, l_j), suppose that k is the spatial relation between i and j defined in the query. We define $d_\mathcal{Q}(l_i, l_j, k) = 1$ if the spatial relation between l_i and l_j is consistent with the spatial relation k in the query $\mathcal{Q}$, and $d_\mathcal{Q}(l_i, l_j, k) = 0$ otherwise. Let $E_\mathcal{Q}$ denote the set of object pairs and their spatial relations defined in the query $\mathcal{Q}$, this potential function is then parameterized as follows:

$$\beta^\top \psi(\mathcal{Q}, L) = \sum_{(i,j,k) \in E_\mathcal{Q}} \beta_{ijk} \cdot d_\mathcal{Q}(l_i, l_j, k) \tag{4}$$

where β_{ijk} is a scalar parameter that weights the spatial relation k between object labels (i, j) defined in the query.

Non-query Object Detector Model $\gamma^\top \omega(\mathcal{Q}, L, I)$: The first two potential functions (Eq. 3 and Eq. 4) only consider object classes that are in the query. For image retrieval, object classes not in the query can also be informative. For example, if the query is "car" and the "road" detector has a high response in the image, this image is likely to be relevant even if the "car" detector does not fire. This potential function tries to capture this co-occurrence contextual information between object classes. If we denote the set of object classes not in the query $\mathcal{Q}$ as $\mathcal{X} \setminus V_\mathcal{Q}$, this potential function is parameterized as

$$\gamma^\top \omega(\mathcal{Q}, L, I) = \sum_{i \in V_\mathcal{Q}} \sum_{j \in \mathcal{X} \setminus V_\mathcal{Q}} \gamma_{ij} f(I(l_j)) \tag{5}$$

where γ_{ij} is a parameter encodes the co-occurrence between object class i in the query $\mathcal{Q}$ and object class j in the context. $f(I(l_j))$ can be interpreted as the confidence of this co-occurrence. γ_{ij} is large if object class j in the context supports object i in the query.

Non-query Object Spatial Relation Model $\eta^\top \varphi(\mathcal{Q}, L, I)$: This potential function captures the spatial arrangements between object classes in the query and those not in the query. It is parameterized as:

$$\eta^\top \varphi(\mathcal{Q}, L, I) = \sum_{i \in V_\mathcal{Q}} \sum_{j \in \mathcal{X} \setminus V_\mathcal{Q}} \eta_{ij}^\top d(l_i, l_j) \tag{6}$$

where $d(\cdot)$ is a spatial relation feature that bins the relative location of l_i and l_j into one of the three canonical relations (similar to [13]): *above*, *below* and *overlap*. Hence $d(l_i, l_j)$ is a sparse vector of all zeros with a single one for the bin occupied by the spatial relation between l_i and l_j. η_{ij} is a vector of parameters that favors certain spatial relation between object classes i and j.

Global Query Model $\zeta^\top \Omega(\mathcal{Q}, I)$: This potential function captures how likely the image I contains the objects described by query $\mathcal{Q}$ based on global image features of I. It is parameterized as:

$$\zeta^\top \Omega(\mathcal{Q}, I) = \sum_{i \in V_\mathcal{Q}} \zeta_i^\top g(I) \tag{7}$$

Here $g(I)$ is the GIST feature of the whole image. The parameter ζ_i is a template for the object class i. Intuitively, ζ_i captures the global scene properties (e.g. highway scene) typically associated with certain objects (e.g. cars). In our implementation, rather than using the raw gist descriptor, we pretrain the gist descriptor using logistic regression and obtain a score for each object category. In this way, $g(I)$ is the output score of a single object category.

3 Inference

During testing, we are given the model parameters $\Theta = \{\alpha, \beta, \gamma, \eta, \zeta\}$, a query $\mathcal{Q}$, and a collection of test images $\mathcal{I}_{test}$ without object bounding boxes. For each image $I \in \mathcal{I}_{test}$, we need to compute the score $H_\Theta(I, \mathcal{Q})$. The final retrieval results are simply the set of images with high scores.

The key computational bottleneck of the testing is to solve the following inference problem for a given query $\mathcal{Q}$ and image I:

$$H_\Theta(I, \mathcal{Q}) = \underset{L}{\text{argmax}} F_\Theta(L, \mathcal{Q}, I) = \underset{L}{\text{argmax}} \Theta^\top \Psi(L, \mathcal{Q}, I) \tag{8}$$

The inference problem in Eq. 8 is hard because we need to search over all the possible configurations (i.e. locations and scales) for each object class, and find the complete configurations for all the object classes that jointly maximize Eq. 8. If we only consider the first component of the model $\alpha^\top \phi(\mathcal{Q}, L, I)$, this amounts to running each object detector separately in a sliding window manner and

picking the optimal configuration for each object class independently. With our full model defined in Eq. 2, the inference problem in Eq. 8 is computationally infeasible.

To address this computational issue, we use several approximation strategies. First of all, we reduce the search space for an object. This is achieved by running an independently learned detector on all locations/scales in the image I in a standard sliding window fashion, then doing a non-maximum suppression to obtain a set $\mathcal{L}_i$ of candidates configurations for this object class. In our experiments, the number of candidate configurations for each object class is reduced to around five. When solving the inference problem in Eq. 8, we restrict the configuration l_i of this object class to be one of its corresponding candidate configurations in $\mathcal{L}_i$, i.e.

$$H_{\Theta}(I, \mathcal{Q}) = \underset{L: l_i \in \mathcal{L}_i, \forall i \in \mathcal{X}}{\operatorname{argmax}} \Theta^{\top} \Psi(L, \mathcal{Q}, I) \tag{9}$$

It is easy to show that the inference problem in Eq. 9 is equivalent to the MAP estimation in a Markov Random Field with K nodes. Each vertex in the MRF corresponds to an object class, and each node i has $|\mathcal{L}_i|$ possible states. An edge (i, j) in the MRF corresponds to the relation between objects i and j. The optimization problem in Eq. 9 is still hard if we have to consider the relation between all pairs of object classes, i.e. when the relation between object classes is represented by a complete graph. To further speed-up, we prune the graph by removing the edge between i and j if these two objects rarely co-occur. For example, it is not semantically meaningful to learn the spatial relations between "car" and "sofa" that almost do not occur together. In our implementation, we enforce the graph sparsity by setting the degree of vertex that corresponds to an object class in the query to 10. The inference takes around 0.05 sec per image in MATLAB on a 2.8GHZ CPU 8GB RAM PC.

4 Learning with Latent Ranking SVM

We learn our model in a latent ranking SVM framework. We assume we are given training images annotated with ground truth object locations. However, learning using these object location annotations can be problematic – we do not have access to these labels at test time, and object detectors for many classes are notoriously unreliable. Hence, we develop a modified learning approach to handle the fact that these will be noisy object detector responses at test time.

More precisely, our training dataset $\mathbf{D}$ consists of a set of triples $\mathbf{D} = \{(\mathcal{Q}^t, I_p^t, I_q^t)\}_t$. Here $\mathcal{Q}^t$ is a structured object query. I_p^t and I_q^t are two images where I_p^t is more relevant to the query $\mathcal{Q}^t$ than I_q^t. The training images are also labeled with ground-truth object locations. We use L_p^t and L_q^t to denote the ground-truth object configurations for I_p^t and I_q^t, respectively. The goal of our learning method is to set the model parameters Θ that tend to score I_p^t higher than I_q^t for the query $\mathcal{Q}^t$. One natural way of learning the parameters is to use the rank SVM formulation as follows:

$$\min_{\theta,\xi\geq 0} \ \frac{1}{2}||\Theta||^2 + C\sum_t \xi_t \tag{10a}$$

$$\text{s.t.} \quad F_\Theta(L_p^t, I_p^t, \mathcal{Q}^t) - F_\Theta(L_q^t, I_q^t, \mathcal{Q}^t) \geq 1 - \xi_t \ \forall t \tag{10b}$$

The intuition of Eq. 10 is to set the score of (I_p^t, L_p^t) higher than (I_q^t, L_q^t) by a margin of 1. Similar to standard SVMs, a slack variable ξ_t is used to handle soft margin, and C is a parameter controlling the tradeoff between training accuracy and regularization.

Although seemingly natural, the formulation in Eq. 10 has a problem – it uses the ground-truth object configuration L_p^t and L_q^t during learning. Since we do not have access to the ground-truth object configuration during testing, this means the learning does not mimic the situation at testing.

To address this issue, we modify the formulation in Eq. 10 as follows. For an object class i that appears in an image I, we use $\mathcal{S}_I(l_i)$ to denote the set of locations/scales that significantly overlap with one of the bounding boxes of object i in image I. If the object i does not appear in an image I, we simply define $\mathcal{S}_I(l_i)$ to be the set of all locations/scales in the image. We define $\mathcal{S}_I(L)$ as the Cartesian product of $\mathcal{S}_I(l_i)$ $(i = 1, 2, ..., K)$, i.e. $\mathcal{S}_I(L) = \mathcal{S}_I(l_1) \times \mathcal{S}_I(l_2) \times \cdots \times \mathcal{S}_I(l_K)$. Then we modify the formulation in Eq. 10 as follows:

$$\min_{\theta,\xi\geq 0} \ \frac{1}{2}||\Theta||^2 + C\sum_t \xi_t \tag{11a}$$

$$\text{s.t.} \quad \max_{L\in\mathcal{S}_{I_p^t}(L_p^t)} F_\Theta(L, I_p^t, \mathcal{Q}^t) - \max_{L'\in\mathcal{S}_{I_q^t}(L_q^t)} F_\Theta(L', I_q^t, \mathcal{Q}^t) \geq 1 - \xi_t \ \forall t \tag{11b}$$

The intuition behind Eq. 11 is that we allow the object configuration to move a bit around its ground-truth locations (if ground-truth bounding boxes exist for this object) for robustness. Similar ideas have been used in [15]. If ground-truth bounding boxes do not exist for an object (i.e. the object does not appear in the image), we simply consider every possible location/scale in the image for this object. We call the formulation in Eq. 11 the *latent ranking SVM*. It extends the latent SVMs in [15] to the case of learning ranking functions. Similar to latent SVMs, we can use an iterative strategy to solve Eq. 11:

- Holding L and L' fixed, learn the parameters Θ by solving a standard ranking SVM. We use the algorithm described in [16] to solve this problem.
- Holding Θ fixed, infer L and L' by solving the following optimization problems, which can be solved by loopy belief propagation:

$$L^* = \operatorname*{argmax}_{L\in\mathcal{S}_{I_p^t}(L_p^t)} \Theta^\top \Psi(L, \mathcal{Q}, I_p^t) \tag{12a}$$

$$L'^* = \operatorname*{argmax}_{L'\in\mathcal{S}_{I_q^t}(L_q^t)} \Theta^\top \Psi(L', \mathcal{Q}, I_q^t) \tag{12b}$$

These two steps are repeated until convergence.

5 Experiments

To the best of our knowledge, there are no public datasets available for the task of image retrieval with structured object queries. Instead, we create our own dataset based on the SUN 09 dataset [17] originally collected for the study of scene recognition and object detection. The SUN 09 dataset consists of $12,000$ annotated images with more than a hundred of object categories and $152,000$ annotated object instances. A typical SUN 09 image has around 14 object instances in 7 object categories. Similar to [17], we use 4367 images for training and 4317 images for testing. We create structured object queries (details below) from the ground-truth labeling of object classes on the dataset.

We generate the query set from the ground-truth bounding boxes on the training set, and each query is associated with a list of images from the training set. Due to the large number of object categories of the SUN 09 dataset, it is impossible to consider all the possible combinations of objects and relations as queries. Instead, we generate queries as follows. First, we only consider the queries with five different structures as shown in Fig. 3. For example, the query with the structure of Fig.3 (a) is a phrase with only two object labels, such as {"car on road"} {"sky above building"}. To create the final query set used in our experiments, we consider the combinations of objects and relations appearing in the training set, then randomly sample a subset from it. In the end, we obtain 1926 structured object queries. We train a separate model for each of the five query structures.

Baselines: We compare our full model with several baseline methods. All methods are evaluated using exactly the same experimental setup. For the first baseline method (called "part-based detector" in Table 1), we ignore the object relations in the query and heuristically combine detector scores for the retrieval. Given a query, we simply use the sum of maximum response scores from each object detector to indicate the relevance of the testing image. For example, given a query of $\mathcal{Q} = \{$"car on road"$\}$, we first apply both car and road detector on the testing images. Then, a testing image will be scored by the sum of the maximum car detector score and maximum road detector score on this image. Note that the detector scores across all object categories have been normalized to the same scale by logistic regression. Here we use the precomputed set of detector outputs provided by [17], where the part-based detector [15] is used. The second baseline (called "MARR" in Table 1) is the method in [11]. The original implementation of the MARR model is not available so we re-implemented it. We define another two baseline methods by considering some components of our full model, including: *global model* (Eq. 7) and a model without the global potential, which we call *structure model* (Eq. 3 - 6). For a fair comparison, we use the same rankSVM [16] solver for all the methods.

Evaluation: We use the Mean Average Precision (mean AP) across all queries to measure the performance. This measurement has been widely adopted in the area of image retrieval. Note that some queries we used in training are associated with only a few testing instances, we therefore discard the queries that appear

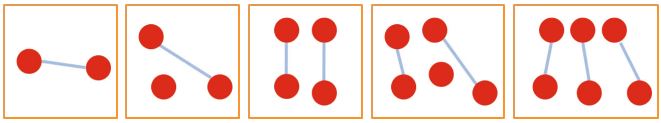

Fig. 3. Different query structures we used to evaluate the model. Nodes denote object classes and edges denote spatial relations between pairs of objects in the query. For example, a query {"car on road, sky above building, pedestrians"} is represented with structure (d).

Table 1. Mean Average Precision (averaged across all queries) of different methods for image retrieval. We show the mean average precision for each of the five query structure types (see Fig. 3) and the overall mean average precision.

Method	Structure (a)	Structure (b)	Structure (c)	Structure (d)	Structure (e)
part-based detector	7.76	5.45	5.52	5.76	5.95
MARR [11]	10.10	6.95	6.35	6.51	6.28
global model	8.01	7.30	6.79	7.92	6.41
structure model	11.16	7.80	8.76	8.17	8.49
full model	**12.67**	**8.80**	**9.31**	**9.93**	**9.64**

less than 30 times in the test set and compute the mean AP on the remaining 1573 queries. The results are summarized in Table 1. We can see that our method significantly outperforms all the baseline methods. By looking at the Mean APs of different components of the model, we can conclude that exploring the structures of queries as well as contextual objects (*structure model*) is beneficial for image retrieval, modeling the scene (global model) further improves the performance. Note that our model still outperforms the MARR model [11] after removing the global potential (see the Mean AP of *structure model*), which confirms that the performance boost is not simply due to including gist features by the global model. We can visualize some of the learned parameters to get some insights about our approach. For example, the parameters γ capture the co-occurrence of objects. If we fix one object (say "car"), we can use γ to visualize which other objects are likely to co-occur with cars. In Fig. 4, we show several such examples. We can observe some intuitive co-occurrence patterns, such as car with road; stove with dishwasher; etc. This suggests that when retrieving images for "car", the presence of "road" is a good indicator of a relevant image.

Fig. 5 shows the top ranked images for two example queries: {"sky above mountain, fence"} and {"curtain above bed, wall above floor"}. We can see that our method works particularly well for objects that are hard to detect, such as "fence" and "curtain". This is because our learning algorithm utilizes rich structured information such as spatial layout of objects as well as co-occurrence between certain pairs of objects. The structured information will constrain the "hard-to-detect" objects to appear at certain locations given the context of other easier objects.

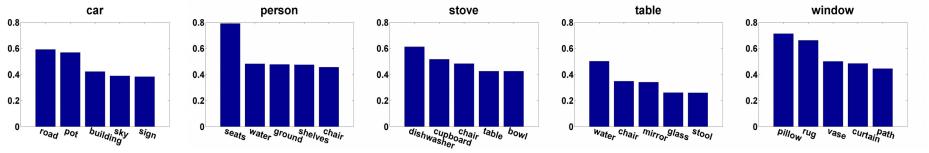

Fig. 4. Visualization of learnt γ parameters (Eq. 5). Each plot corresponds to an object category i, we show the weights of the top five components of γ_i (y-axis) and the corresponding object classes (x-axis).

Fig. 5. Examples of retrieval results showing top five ranked images using our method and the MARR model [11]. The top two rows and the last two rows are the top ranked images w.r.t. the structured queries {sky-above-mountain, fence} and {curtain-above-bed, wall-above-floor} respectively. The color of the bounding box encircles the image indicates whether the image contains the corresponding query (green) or not (red). The predicted locations of the objects in the query are also shown.

5.1 Image Ranking

It is straightforward to extend our image retrieval model to image ranking. In image retrieval, given a query $\mathcal{Q}$ the output is an image set $\mathcal{Y}$ that are considered as relevant to $\mathcal{Q}$. In the case of image ranking, given a query $\mathcal{Q}$, our goal is to rank the images according to their relevance to $\mathcal{Q}$.

Different from many previous ranking methods which simply use a binary rank (relevant and irrelevant), we utilize multiple levels of relevance. Images containing all the words/phrases in the query $\mathcal{Q}$ are assigned a relevance $|\mathcal{Q}|$, images containing $|\mathcal{Q}| - 1$ words/phrases in the query are assigned a relevance $|\mathcal{Q}| - 1$, and so on, relevance 0 is assigned to the images containing none of the words/phrases in the query.

We use the Normalized Discounted Cumulative Gains (NDCG) [18] to measure the performance of our image ranking approach. We report NDCG values at five different truncation levels. Since queries with simple structures such as Fig. 3 (a) do not require the use of multiple relevance levels, we only evaluate our method on the queries with the most complex structures (Fig. 3 (e)). The comparison of our method and the baselines is shown in Table 2. Since for the image ranking task, the MARR model [11] uses a complex loss that directly optimizes the NDCG score, which is different from our learning criterion, we do not compare with it in this task.

Table 2. Ranking performances for different methods. We use NDCG at different truncation levels to measure the performance. Here we only consider the queries of structure type shown in Fig. 3(e).

Method	NDCG@20	NDCG@40	NDCG@60	NDCG@80	NDCG@100
global model	0.32	0.38	0.47	0.47	0.51
structure model	0.64	0.69	0.79	0.82	0.83
full model	**0.66**	**0.72**	**0.82**	**0.84**	**0.85**

6 Conclusion

We have presented a learning approach for image retrieval with structured object queries. In contrast to previous work that only considers queries consisting of object classes, the structured object queries in our method can encode both object classes and their relations. In our experiment, we have consider three spatial relations. But we would like to point out that our proposed model is general and can be used to encode other object relations. We have demonstrated the effectiveness of our method on a challenging dataset involving more than a hundred of object classes. Our ultimate goal is to build image retrieval systems that allow users to express their queries in natural languages. As future work, we plan to incorporate other aspects of natural languages, such as verbs, into our model.

References

1. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. Journal of Machine Learning Research 3, 1107–1135 (2003)
2. Li, L.J., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
3. Gupta, A., Srinivasan, P., Shi, J., Davis, L.S.: Understanding videos, constructing plots: Learning a visually grounded storyline model from annotated videos. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009)
4. Sadeghi, M.A., Farhadi, A.: Recognition using visual phrases. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2011)
5. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every Picture Tells a Story: Generating Sentences from Images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 15–29. Springer, Heidelberg (2010)
6. Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Baby talk: Understanding and generating simple image descriptions. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2011)
7. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Survey 40, 1–60 (2008)
8. Joachims, T.: Optimizing search engines using clickthrough data. In: ACM SIGKDD (2002)
9. Yu, C.N., Joachims, T.: Learning structural SVMs with latent variables. In: International Conference on Machine Learning (2009)
10. Blaschko, M.B., Vedaldi, A., Zisserman, A.: Simultaneous object detection and ranking with weak supervision. In: NIPS (2010)
11. Siddiquie, B., Feris, R.S., Davis, L.S.: Image ranking and retrieval based on multi-attribute queries. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2011)
12. Parikh, D., Zitnick, C.L., Chen, T.: From appearance to context-based recognition: Dense labeling in small images. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009)
13. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: IEEE International Conference on Computer Vision (2009)
14. Chen, T., Cheng, M.M., Tan, P., Shamir, A., Hu, S.M.: Sketch2Photo: Internet image montage. ACM Transactions on Graphics (2009)
15. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)
16. Joachims, T.: Training linear SVMs in linear time. In: SIGKDD (2006)
17. Choi, M.J., Lim, J.J., Torralba, A., Willsky, A.S.: Exploiting hierarchical context on a large database of object categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010)
18. Chapelle, O., Le, Q., Smola, A.: Large margin optimization of ranking measures. In: NIPS Workshop on Learning to Rank (2007)