Boosting Object Retrieval With Group Queries

Yanzhi Chen, Xi Li, Anthony Dick, and Anton van den Hengel

***Abstract*—Given a query image of an object, object retrieval aims to return all images from a corpus that depict the same object. Inevitably, the accuracy of the result depends strongly on the quality of the query image. Several measures have been taken to improve retrieval result quality, including the addition of a bounding box to the query, the mining of highly ranked results for more views of the object, and spatial consistency re-ranking. In this letter, we propose a discriminative criterion for improving result quality. This criterion lends itself to the addition of extra query data, and we show that multiple query images can be combined to produce enhanced results. Experiments compare the performance of the method to state-of-the-art in object retrieval, and show how performance is lifted by the inclusion of further query images.**

***Index Terms*—Discriminative ranking function, group query, object retrieval.**

I. Introduction

OVER the past decade, the problem of image based object retrieval has been actively researched, with significant improvements to both accuracy and scale. Many works have used the bag-of-words (BoW) model, where images are represented as term frequency inverse document frequency (tf-idf) weight vectors and ranked by their dot product similarity to a query image. To improve retrieval accuracy, methods built on the BoW model make use of several auxiliary steps either to improve the image representation [1]–[3] or post-process the query result list [4]–[6]. Despite encouraging results, these methods have difficulty in capturing the diverse distribution of possible appearances of the query object, leading to a strong dependence on query image quality. As shown in Fig. 1, the performance when retrieving the same building object varies dramatically: the retrieval case with front viewpoint (I) in Fig. 1 has significantly higher accuracy than the other cases (the right side viewpoint (II) and left side viewpoint (III) in Fig. 1). In the case of (II) and (III), query expansion [5] usually fails because there are not enough true positives for spatial verification, as investigated in [7].

Several standard object retrieval datasets contain bounding boxes specifying the image region occupied by the object of interest. These boxes are exploited by most of the methods listed above, to improve retrieval result quality. Less common is the use of multiple query images to specify a single object, even though this information is more likely to be available in real cases for object retrieval. For example, image sharing websites such as Flickr or Facebook group images into communities containing the same or similar subjects. Typically, each community contains images of the same object from varying viewpoints. One of the few previous works to exploit this information is [8], where users input multiple query images as positive samples of an object class, along with negative images that do not contain the object. Using these positive and negative samples, a discriminative classification model is learned to rank all images in the dataset. Alternatively, target object in the dataset can be matched by a discriminative relevance evaluation, where positive and negative queries are used to obtain the mutual information score [9]. A discriminative ranking criterion is well suited to the use of multiple query images as it models the set of positive samples non-parametrically, and can therefore accommodate a diverse set of image views. It also naturally benefits from the addition of extra positive and negative samples.

Manuscript received July 03, 2012; revised August 19, 2012; accepted August 22, 2012. Date of publication September 10, 2012; date of current version September 21, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. H. Vicky Zhao.

The authors are with the School of Computer Science, The University of Adelaide, Adelaide SA 5005, Australia (e-mail: yanzhi.chen@adelaide.edu.au; xi.li03@adelaide.edu.au; anthony.dick@adelaide.edu.au; anton.vandenhengel@adelaide.edu.au).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Digital Object Identifier 10.1109/LSP.2012.2216875

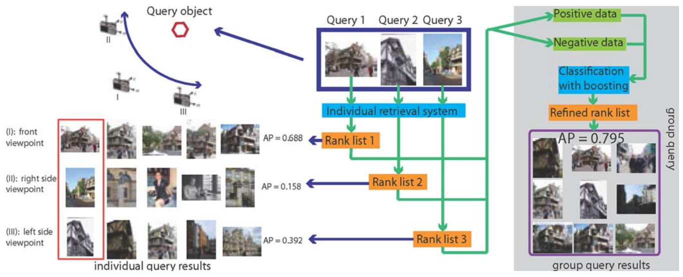

Fig. 1. Illustration of our object retrieval method, using 3 query images of the same object from different viewpoints. Individual query results are mined for positive and negative examples, which are then used to train a boosted classifier. The classification function is then used to rank image results. Typically, it obtains higher accuracy than is possible from any individual query.

Contribution: We introduce a group query-based object retrieval method (illustrated in Fig. 1), which uses a small group of images of a query object to reflect its appearance variation (e.g., different viewpoints). We propose a novel discriminative ranking criterion for multiple queries that builds on previous methods [8], [10]. In contrast to [8], our method can work even with a single query image, but exploits the extra information in multiple queries when available. These query instances are used to automatically collect a set of query-dependent positive and negative data samples as used by discriminative query expansion (DQE) [10]. In contrast to average query expansion (AQE) [5], which uses positive samples to improve query recall, DQE re-ranks the dataset images by a data-dependent weight vector learnt from both positive and negative samples. Unlike the linear ranking models used in [8], [10], [11], our proposed method constructs a nonlinear ranking model using an ensemble of linear support vector machines (SVM) that are adaptively weighted by boosting. The constructed ranking model features both nonlinear inter-class separation and efficient ranking prediction.

II. Discriminative Object Retrieval

A. Overview

We define a group query as the set $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^M$, where each instance $\mathbf{q}_i$ is represented as a tf-idf vector and M is the number of query instances. We let $\mathcal{R} = \{\mathbf{r}_i\}_{i=1}^M$ denote the corresponding set of ranking lists, where each ranking list $\mathbf{r}_i$ is obtained by performing pairwise dot product-based image matching between the query $\mathbf{q}_i$ and the database images. The goal of our object retrieval method is to design an effective and efficient ranking function for capturing the underlying affinity relationships between $\mathcal{Q}$ and the database images (such that relevant images are highly ranked while irrelevant images are lowly ranked). When $M = 1$, our group query method degenerates to a single query method similar to the DQE method in [10], but with an additional boosting step as described in Section II.B.2. More details of our object retrieval method can be found in Fig. 1 and Algorithm 1.

Algorithm 1 Ranking function learning using Adaboost-LinearSVM

```
1:  Input: Training samples $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^K$, maximum number of boosting iteration $T$.
2:  Output: Ranking function $F(\mathbf{x})$.
3:  Initialize: Data distribution: $D_1(i) = (1)/(K), \forall i$ and $t \leftarrow 1$.
4:  while $t < T$ do
5:      Weighted random sampling positive and negative data from $\mathcal{T}$ with distribution $D_t$.
6:      Train a linear SVM $f_t(\mathbf{x})$ on the training set.
7:      Calculate the training error: $\epsilon_t = \sum_{i=1}^K D_t(i) I(y_i \neq f_t(\mathbf{x}_i))$, where $I$ is the indicator function.
8:      Calculate the weight: $\alpha_t = (1)/(2)\ln((1-\epsilon_t))/(\epsilon_t)$.
9:      Update the distribution $D_{t+1}(i) = (D_t(i)\exp(-\alpha y_i f_t(\mathbf{x}_i)))/(Z_t)$, where $Z_t$ is a normalization factor.
10:     $t \leftarrow t+1$.
11: end while
12: Return: Ranking function $F(x) = \sum_{t=1}^T \alpha_t \cdot f_t(\mathbf{x})$.
```

B. Discriminative Ranking Function for Group Query

1) Training Sample Selection: Initially, we obtain training data samples $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^K$ where $\mathbf{x}_i$ is the tf-idf vector and $y_i \in \{1, -1\}$ inidcates whether the corresponding image contains the object. There are many ways to collect these samples from query instances; we use spatial verification [4], as in [10]. This proceeds as follows: images (tf-idf vectors) with more than a minimum number (10) of spatially consistent matches to any query image are provided as positive examples and images with the lowest ranked non-zero similarity score are taken as negative examples.

2) Discriminative Ranking Function With Boosting: Using the training data samples $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^K$, we learn a ranking function that aims to capture non-linear discriminative information from the training data samples. Linear SVM classifiers are adopted as weak learners in a boosting framework due to their simplicity and efficiency. Using ensemble learning, these linear SVM classifiers can be adaptively combined to generate a strong classifier. In each boosting iteration, a linear SVM classifier is learned over a subset of $\mathcal{T}$, which is obtained by weighted random sampling over $\mathcal{T}$. Instead of simple linear SVM ranking functions used in [10], our boosting-like ranking function effectively captures nonlinear discriminative information by constructing a nonlinear decision boundary using an additive linear approximation. To cope with the nonlinear classification problem, non-linear kernel SVM is an alternative, but prediction is usually computationally expensive, making it impractical for scalable object retrieval. Therefore, our ranking function is more suitable because it can not only improve the retrieval effectiveness but does so with low computational complexity as [10]. Algorithm 1 provides the details of constructing our ranking function. Moreover, as most computation of Algorithm 1 is spent on training linear SVMs, its running time can be reduced by parallelising SVM training before boosting iteration. We do this by collecting a weak classifiers pool (linear SVM) before boosting from random positive and negative samples from $\mathcal{T}$. After that, an approximate ranking function $F(x)$ is formed to fit the data by selecting from these weak classifiers during each boosting iteration. In effect, steps 5 and 6 in Algorithm 1 are executed in parallel before the boosting iteration begins.

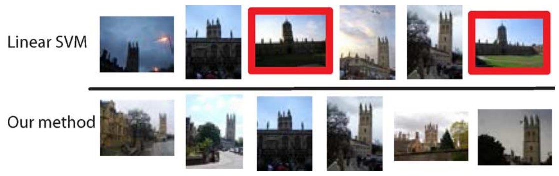

Fig. 2. Top-6 retrieved results of using the linear SVM ranking function and our boosting-like ranking function with respect to the object landmark *Magdalen*. The highlighted images correspond to false positive samples.

As shown in Fig. 1, a group query with respect to some clear object landmarks can help to overcome the limitation of individual query. For example, *All souls* and *Radcliffe camera* from the Oxford dataset have average precision (AP) scores 0.96 and 0.97, respectively. The boosting-like ranking function attains slightly higher results, with AP scores being 0.98 and 0.97 on the same clear landmarks. However, the retrieval performance with the linear SVM ranking function degrades greatly in the cases of lower quality object landmarks, e.g., *Magdalen* and *Keble* with the AP score being 0.288 and 0.692, respectively. In contrast, using boosting enables the retrieval accuracy (mAP) to reach 0.407 and 0.870 on *Magdalen* and *Keble*. For an intuitive understanding, Fig. 2 shows the top ranked results of using two different ranking functions with respect to *Magdalen*. As is seen in Fig. 2, the top ranked results of our boosting-like ranking function is better than those of the linear SVM ranking function (containing a few false positive samples).

III. Experiments

A. Experimental Setup

Dataset Description: The retrieval experiments are conducted on three public object retrieval datasets: two small-scale datasets (Oxford 5 K and Paris 6 K [12]) and a large-scale dataset Oxford 105 K (consisting of Oxford 5 K images and 100 K images from MIRFLICKR-1M [13]. Both of the Oxford and Paris datasets [12] contain 11 building landmarks for

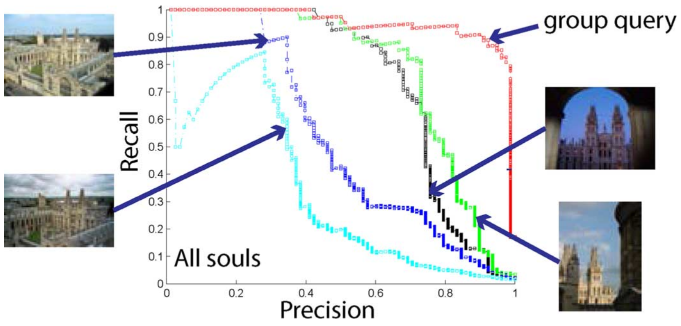

Fig. 3. Precision-recall curves of individual query *v.s.* group query. We obtain higher retrieval accuracy of group query compared to the individual queries. Best viewed in color. More examples can be found in the supplemental file.

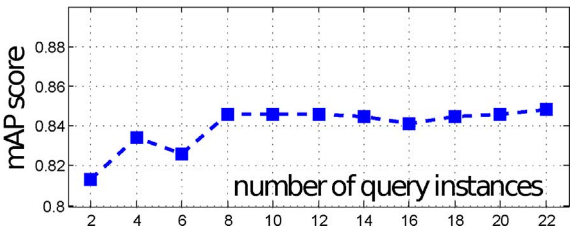

Fig. 4. Retrieval performance with different numbers of query instances. In this experiments, we use high quality query instances.

evaluation, with four kinds of visual condition: "good" (clear image of the object), "ok" (more than 25% of the object is clear visible), "junk" (less than 25% of the object is clear visible) or "bad" (the object is not present). Each image within these datasets is represented as a histogram of SIFT words after tf-idf weighting. The SIFT words are obtained by quantizing the SIFT feature descriptors using approximate k-means [4], [14].

Implementation Details: In each boosting iteration, we randomly select (up to) 50 and 200 tf-idf vectors as positive and negative examples, respectively. The linear SVM classifier is trained using the LIBSVM tool [15] with $C = 1$ as in [10]. The maximum iteration number T in boosting learning is set to 20. We implement our method with Matlab on a 2.20 GHz double core machine with 2 GB memory.

Group query instances are provided online. These can be any images of a particular object. In our experiments, we utilize annotated image collections to organize query instances $\mathcal{Q}$ which are varied but contain the same object landmark. The query instances are organized as follows: i) We randomly select the same number of images from "good" and "ok" collection as **high quality query** to evaluate the effect of group query (Tables I–IV and Fig. 4). ii) Similarly, images from "good", "ok" and "junk" are treated as **low quality query** to evaluate the discriminative ranking function (Table II). iii) To compare with state-of-the-art, we use the 55 queries with bounding boxes defined in [12] (Table V). We adopt the widely used *mean average precision* (mAP) [4] to evaluate the retrieval performance.

B. Experimental Results

Effect of Using Group Query: Table I compares the mAP score obtained for each landmark in the Oxford data set, using individual query images $(M = 1)$ and groups of 4 query images $(M = 4)$. The individual query results are obtained by running each of the 4 queries in the group separately, and storing the maximum result. The query groups are sampled using the "High quality" strategy. Results are averaged over 11 groups. It is clear from Table I that the group query improves retrieval performance for 9 out of 11 queries, and by an average of 29.9%. Note that our method fails in the cases of (*Magdalen, Pitt river*) due to a lack of quality query images. Fig. 3 illustrates that the group query can result in significantly higher precision-recall performance than any individual query, using the object landmark (*all souls*).

Evaluation of Query Instance Number M: Fig. 4 investigates the effect of varying the numbers of query instances. As is seen in Fig. 4, the average retrieval accuracy rises when M increases. As M increases, the computational cost increases linearly, due to the repetition of the dot-product ranking and spatial verification for each query instance. In order to balance effectiveness and efficiency, M is set to 4 in the experiments below.

Investigation of Using Linear SVM v.s. Adaboost-LinearSVM: Table II compares the retrieval performance using two types of classifiers used in the discriminative ranking function: the linear SVM (referred to as R1) and the *AdaBoost-LinearSVM* (referred to as R2). As is seen in Table II, the discriminative ranking function with *AdaBoost-LinearSVM* always performs better than linear SVM. The superiority of the boosting-like ranking function is more evident in the low quality query. For example, in Group E the retrieval performance using boosting-like ranking function (R2) is 3.7% (5.9%) higher on the Oxford (Paris) datasets than the results using linear SVM ranking function (R1). Moreover, Table III illustrates the re-ranking CPU time of different discriminative ranking functions. From Table III, we see that our method achieves the best retrieval accuracy with high searching efficiency in the post-processing. By parallelising SVM training

TABLE I
COMPARISON OF (MAXIMUM) INDIVIDUAL AND GROUP QUERY RETRIEVAL PERFORMANCE ON THE OXFORD 5 K DATASET

Landmark	$M = 4$	$M = 1$	Landmark	$M = 4$	$M = 1$
all souls	0.980	0.746	hertford	0.895	0.801
ashmolean	0.805	0.418	keble	0.870	0.844
balliol	0.867	0.200	magdalen	0.407	0.439
bodleian	0.799	0.774	pitt river	0.852	0.974
chr. church	0.864	0.762	rad. camera	0.965	0.326
cornmarket	0.711	0.657			

TABLE II
RETRIEVAL PERFORMANCE WITH DIFFERENT DISCRIMINATIVE RANKING FUNCTIONS: R1: GROUP QUERY WITH LINEAR SVM [10], R2: GROUP QUERY WITH BOOSTING. S DENOTES USE OF SPATIAL VERIFICATION. M DENOTES THE MAXIMUM NUMBER OF QUERIES IN $\mathcal{Q}$. BOTH **HIGH** AND (**LOW**) QUALITY QUERIES ARE TESTED, AS INDICATED IN THE TABLE

	Method	S	M	Oxford 5K	Paris 6K
A	R1	$\surd$	2	0.789	0.772
high	**R2**	$\surd$	2	0.813	0.780
B	R1	$\surd$	4	0.809	0.857
high	**R2**	$\surd$	4	0.834	0.871
C	R1	$\surd$	10	0.833	0.864
high	**R2**	$\surd$	10	0.846	0.875
D	R1	$\surd$	2	0.668	0.701
low	**R2**	$\surd$	2	0.676	0.714
E	R1	$\surd$	4	0.755	0.682
low	**R2**	$\surd$	4	0.783	0.722
F	R1	$\surd$	10	0.756	0.791
low	**R2**	$\surd$	10	0.791	0.824

TABLE III
AVERAGE RE-RANKING CPU TIME FOR DIFFERENT CLASSIFIERS USED IN THE DISCRIMINATIVE RANKING FUNCTION. THE GROUP QUERY ($M = 4$) IS CONDUCTED ON THE OXFORD 5 K DATASET

Method	mAP	re-ranking CPU time (s)
linear SVM	0.809	0.37
Adaboost-linearSVM	0.834	1.12
RBF-kernel SVM	0.815	12.26
Adaboost-linearSVM (parallelised)	0.830	0.49

TABLE IV
COMPARISON WITH QUERY EXPANSION METHOD, APPLIED TO BOTH INDIVIDUAL AND GROUP QUERIES

Method	S	M	Oxford 5K	Paris 6K
tf-idf + dot product (maximum)		4	0.631	0.605
AQE (maximum)	$\surd$	4	0.742	0.657
AQE (positive examples)	$\surd$	4	0.743	0.809
Our method	$\surd$	4	0.834	0.871

TABLE V
RETRIEVAL PERFORMANCE COMPARISON WITH RECENT METHODS, BASED ON 55 QUERIES DEFINED IN [12]. BASELINE [4], SPATIAL VERIFICATION [4], TOTAL RECALL [5] (AQE), AND DISCRIMINATIVE TOTAL RECALL (DQE) [10] ARE OUR IMPLEMENTATION, OTHERS ARE CITED RESULTS

Method	Oxford 5K	Paris 6K	Oxford 105K
Baseline [4]	0.612	0.639	0.515
Spatial verification [4]	0.645	0.655	0.571
Local geometry [6]	0.788	0.634	0.725
AQE [5]	0.800	0.769	0.767
Total recall II [7]	0.827	0.805	0.767
Hello neighbors [16]	0.814	0.803	0.767
DQE [10]	0.798	0.783	0.809
DQE+Boosting	0.823	0.782	0.818
DQE+Boosting (group)	**0.896**	**0.856**	**0.890**

(Section II.B.2), we reduce the re-ranking time at a slight cost to mAP.

Comparison With Query Expansion: Similar to our method, average query expansion (AQE) [5] also applies spatial verification to top ranked retrieval results and collect a number of true positives. However, it uses query averaging of the visual words collected in the positive images, while our method trains a discriminative ranking function with the same positive images and additional negative images. Table IV compares our group query method with average query expansion (AQE) [5] in the following aspects: i) the best (maximum mAP socre) of AQE for each individual query. ii) AQE on all positive examples collected by the group query. As is seen in Table IV, our group query method can outperform AQE in different experimental configurations.

Comparison With State-of-the-Art Methods: We compare our method with several state-of-the-art methods (listed in Table V) that focus on query model refinement with post-processing[1]. In Table V, spatial verification [4] and local geometry [6] use pairwise spatial consistency to promote true positives; AQE [5], [7] and DQE [10] aim to improve the query model; and [16] uses a pre-defined graph to find similar images to query. The instances $\mathcal{Q}$ used for group query in Table V are the 5 ground truth files defined in [12] for each landmark. From Table V, we see that our method can outperform state-of-the-art methods. In particular, when only a single query instance is available, our method $(\mathrm{DQE+Boosting})$ can outperform DQE [10] (using linear SVM ranking function instead). In cases where multiple query instances are available, boosting group retrieval with a small number of query instances can significantly improve the retrieval results.

[1]Note that DQE is our implementation, which is slightly but consistently lower than the results reported in [10] for both DQE and baseline results.

IV. CONCLUSION

In this letter, we have introduced the notion of a group query, and shown its effectiveness for object retrieval. We proposed a boosted discriminative ranking function to refine the group query model. The proposed ranking function captures non-linear discriminative information on the retrieved data samples effectively and efficiently. Experimental results show that our method can achieve higher performance than competing methods.

REFERENCES

[1] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2008.
[2] Y. Jiang, J. Meng, and J. Yuan, "Randomized visual phrases for object search," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 2911–2918.
[3] F. Li, Q. Dai, W. Xu, and G. Er, "Weighted subspace distance and its applications to object recognition and retrieval with image sets," *IEEE Signal Process. Lett.*, vol. 16, no. 3, pp. 227–230, 2009.
[4] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2007, pp. 1–8.
[5] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2007, pp. 1–8.
[6] M. Perd'och, O. Chum, and J. Matas, "Efficient representation of local geometry for large scale object retrieval," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2009.
[7] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2011.
[8] M. Rastegari, C. Fang, and L. Torresani, "Scalable object-class retrieval with approximate and top-k ranking," in *Proc. Int. Conf. Comp. Vis.*, 2011, pp. 2659–2666.
[9] J. Meng, J. Yuan, Y. Jiang, N. Narasimhan, V. Vasudevan, and Y. Wu, "Interactive visual object search through mutual information maximization," in *Proc. ACM Int. Conf. on Multimedia*, 2010, pp. 1147–1150.
[10] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012.
[11] R. Arandjelović and A. Zisserman, "Multiple queries for large scale specific object retrieval," in *Brit. Machine Vision Conf.*, 2012.
[12] [Online]. Available: http://www.robots.ox.ac.uk/~vgg/data/
[13] [Online]. Available: http://press.liacs.nl/mirflickr/dlform.php
[14] [Online]. Available: http://www.robots.ox.ac.uk/~vgg/software/fast-cluster/
[15] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 1–27, 2011.
[16] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2011.