

Image Reranking by Example: A Semi-supervised Learning Formulation

Zhang Wei, Xue Tianfan

The Chinese University of Hong Kong

1 Introduction

With the popularity of digital cameras, the amount of digital images available on the internet have grown dramatically in recent years. This has brought a new challenge to image retrieval. Currently, text-based search has gain a great success in image retrieval. Most of commercial search engines, such as Google, Live, and Baidu, are based on this technique. The images are retrieved using the associated textual information, such as surrounding text from the web page. The performance of such systems mainly relies on the relevance between the text and the images. However, they may not always match well enough, which causes noisy ranking results. For instance, visually similar images may have very different ranks. So reranking has been proposed to solve the problem.

1.1 Reranking: Problem Definition

Formally, the definition of the re-ranking problem with a query image is as follows. Assuming we have n images, retrieved from initial text-based search results, as illustrated in Fig. 1. The re-ranking process is used to improve the search accuracy by reordering the images based on the multimodal information extracted from the initial text-based search results, the auxiliary knowledge and the example image. The auxiliary knowledge can be the extracted visual features from each images or the multimodal similarities between them.

1.2 Previous Work

There have been two different classes of methods aiming at solving the problem of the text-based search.

One is to refine the ranking result by visual information. Recently, many re-ranking methods by visual information have been proposed both for image retrieval and video retrieval. There are generally three categories of visual reranking methods: classification-based [9, 10, 14, 17], clustering-based [5], and graph-based [6–8, 13, 15, 18]. Most of the previous work on visual reranking assumes that there is a dominant class cluster of images inside image set found by text-based retrieval results, and the irrelevant images are visually different from this cluster. However, this method may face some problems. Given a keyword query,

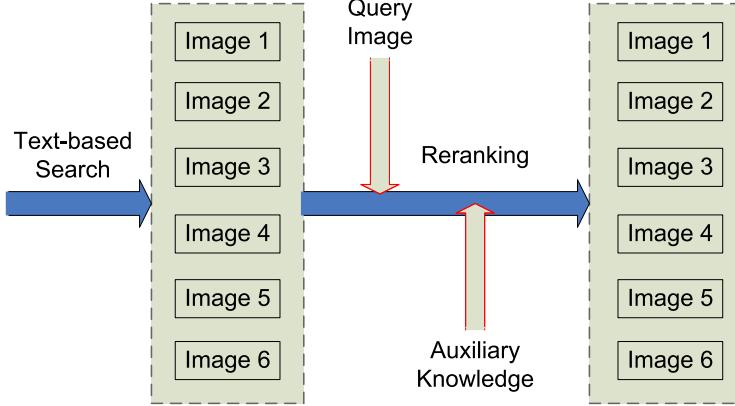


Fig. 1. Illustration of the video re-ranking problem.



Fig. 2. (a) The typical retrieval results of query "Paris". (b) The typical retrieval results of query "Apple".

the images found by text-based method may contain several subclasses of images, of which the intra-group visually difference is very large. For example, in Figure 2, the search result for query "apple" contains two main subclasses: fruit apple and Apple digital product, and the search result for query "Paris" also contains two main subclasses: a person Paris Hilton and a city Paris. The user query for "apple" may either desire for fruit apple or for Apple digital product.

The other approach is example image based reranking. Cui *et al.* proposed a new re-ranking scheme [1]: after query by keyword, user can click on one image, which is the image desired by the user. Then the image search engine re-ranks the images according to this query image: those that are visually similar to query images are top ranked. However, their method is based on direct comparison of the example image and each image in the ranking list. Therefore, noisy results usually appear. Figure 3 shows such a failed example.

1.3 Our Work

In this project, we propose a novel method for example image based reranking. Our method combines the advantages of both the example image based reranking and visual reranking.



Fig. 3. Example image based reranking of keyword “paris”. The orange box and the light blue boxes indicate the example image and noisy images, respectively.

Our algorithm falls into the class of semi-supervised learning methods, which is superior to both unsupervised learning solutions for reranking problem, i.e. conventional visual reranking methods, and supervised learning solutions for reranking problem, i.e. directly comparison of the example image and each image retrieved by the text-based search in existing example image based reranking. To our best knowledge, our algorithm is the first semi-supervised learning algorithm for the reranking problem.

We adopts a graph-based formulation to solve this problem. We further deduct a closed form solution and an iterative solution for this problem. Our algorithm is justified from both the theoretic aspect and the experiments.

2 Graph-based Semi-supervised Learning Formulation

The re-ranking process is as follows. First, the images returned by a text-based search engine are re-ranked according to their distances to the query image, and the distances are used as the initial ranking scores. Second, a graph-based semi-supervised learning algorithm is applied to propagate the scores between images.

We expect that the final scores have the following properties: (1) they are consistent across visually similar images; (2) they are close to the initial scores; (3) the example query images have high scores. This problem can be formulated as graph-based semi-supervised learning.

2.1 A Toy Example

We start by a toy example to show why semi-supervised learning [20] can improve example-based image reranking. As shown in Fig. 4, the separator learnt from both labelled and unlabelled data reflect the structure of the data space better than that learnt only from labelled data. The presence of unlabelled points is used to put the boundary in low density region of the input space. This is specially useful when only few points are labelled.

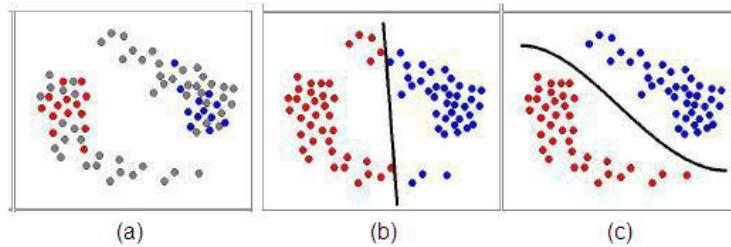


Fig. 4. A toy problem to show the advantage of semi-supervised learning. (a) The labelled data in red and blue, and the unlabelled points in gray. (b) the separator found by supervised learning. (c) the separator found by semi-supervised learning.

2.2 Our Formulation

Given the image set, we first construct a graph, whose nodes are the images and edges are pairwise similarity between images. In our problem, the set of all samples is partitioned into two subsets: a labelled set $L = \{\mathbf{x}_1, \dots, \mathbf{x}_l\}$ and an unlabelled set $U = \{\mathbf{x}_{l+1}, \dots, \mathbf{x}_n\}$. L includes only the example image provided by the user interaction, and U includes all the other images. We give a high score Y to the example image. We aim at predicting the scores for all images.

Denote the initial scores as $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_n]$. Our algorithm minimizes the following function

$$\frac{1}{4} \sum_{i,j} w_{ij} (y_i - y_j)^2 + \frac{1}{2} \lambda_u \sum_{i \in U} (y_i - \hat{y}_i)^2 + \frac{1}{2} \lambda_l \sum_{i \in L} (y_i - Y)^2, \quad (1)$$

where $\mathbf{y} = [y_1, \dots, y_n]^T$ is the final score, w_{ij} is the visual similarity between image i and j , and Y , λ_u and λ_l are constants ($Y \gg \max(\hat{\mathbf{y}}_0)$ and $\lambda_l \gg \lambda_u$). The first term in Eqn. (1) implies that visually similar images have close scores, the second term cooperates the initial scores into the final result, and the last term enforces a constraint on the score of the example image.

Eqn. (1) can be simplified as

$$\frac{1}{2} \mathbf{y}^T L \mathbf{y} + \frac{1}{2} (\mathbf{y} - \mathbf{y}_0)^T \Lambda (\mathbf{y} - \mathbf{y}_0), \quad (2)$$

where L is the graph Laplacian of the weight matrix $W = [w_{ij}]$, Λ is a diagonal matrix, whose diagonal entry $\Lambda(i, i) = \lambda_l$ when $\mathbf{x}_i \in L$ and $\Lambda(i, i) = \lambda_u$ when $\mathbf{x}_i \in U$.

It is easy to derive a closed-form solution for Eqn. (2)

$$\mathbf{y} = (L + \Lambda)^{-1} \Lambda \mathbf{y}_0. \quad (3)$$

2.3 Alternative Graph Laplacian Matrices

The Laplacian matrix L used above is called unnormalized graph Laplacian. It is a symmetric matrix.

Other types of graph Laplacian matrices can also be used, including normalized Laplacian matrix [19] and random walk Laplacian matrix [11].

If we redefine Eqn. (1) as

$$\frac{1}{4} \sum_{i,j} w_{ij} \left(\frac{y_i}{\sqrt{d_i}} - \frac{y_j}{\sqrt{d_j}} \right)^2 + \frac{1}{2} \lambda_u \sum_{i \in U} (y_i - \hat{y}_i)^2 + \frac{1}{2} \lambda_l \sum_{i \in L} (y_i - Y)^2, \quad (4)$$

where d_i is the i -th diagonal element of D , the unnormalized Laplacian matrix L is replace by the normalized Laplacian matrix $L_n = I - D^{-1/2}WD^{-1/2}$.

If we redefine Eqn. (1) as

$$\frac{1}{4} \sum_{i,j} w_{ij} \left(\frac{y_i}{d_i} - \frac{y_j}{d_j} \right)^2 + \frac{1}{2} \lambda_u \sum_{i \in U} \left(\frac{y_i - \hat{y}_i}{d_i} \right)^2 + \frac{1}{2} \lambda_l \sum_{i \in L} \left(\frac{y_i - Y}{d_i} \right)^2, \quad (5)$$

the unnormalized Laplacian matrix L is replace by the random walk Laplacian matrix $L_n = I - D^{-1}W$.

2.4 Theoretical Justifications for Our Algorithm

Our algorithm can be understood from three aspects.

Manifold Interpretation The energy minimization in Eqn. (1) follows manifold assumption, i.e. data lie on separate manifolds and points which can be connected via a path through high density regions on the data manifold are likely to have close scores. We use a regularizer which prefers functions which vary smoothly along the manifold and do not vary in high density regions.

To intuitively understand the manifold assumption, if we choose $w_{ij} = \left(\frac{1}{dist(\mathbf{x}_i, \mathbf{x}_j)} \right)^2$, where $dist(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between \mathbf{x}_i and \mathbf{x}_j , we have

$$\begin{aligned} \sum_{i,j} w_{ij} (y_i - y_j)^2 &= \sum_{i,j} \frac{1}{d_{ij}^2} (y_i - y_j)^2 \\ &= \sum_{i,j} \left(\frac{y_i - y_j}{d_{ij}} \right)^2 \propto \int_{\mathcal{M}} \|\nabla y(\mathbf{x})\|^2 p(\mathbf{x}) d\mathbf{x}. \end{aligned} \quad (6)$$

So the Laplacian matrix is the discrete counterpart of Laplacian-Beltrami operator on the graph. Strict proof can be found in [4].

Bayesian Interpretation The reranking problem aims at the following Maximum A Posteriori (MAP) estimation

$$\mathbf{y}_{opt} = \operatorname{argmax}_{\mathbf{y}} p(\mathbf{y} | \{\mathbf{x}_i\}_{i=1}^n, \mathbf{y}_0). \quad (7)$$

Using Bayes Theorem (conditioned on $\{\mathbf{x}_i\}_{i=1}^n$), the posteriori distribution can be written as

$$\begin{aligned} p(\mathbf{y} | \{\mathbf{x}_i\}_{i=1}^n, \mathbf{y}_0) &= p(\mathbf{y} | \{\mathbf{x}_i\}_{i=1}^n) p(\mathbf{y}_0 | \{\mathbf{x}_i\}_{i=1}^n, \mathbf{y}) \\ &= p(\mathbf{y} | \{\mathbf{x}_i\}_{i=1}^n) p(\mathbf{y}_0 | \mathbf{y}). \end{aligned} \quad (8)$$

By defining the conditional prior

$$p(\mathbf{y} | \{\mathbf{x}_i\}_{i=1}^n) \propto \exp \left[\frac{1}{4\sigma^2} \sum_{i,j} w_{ij} (y_i - y_j)^2 \right] \quad (9)$$

and the conditional likelihood

$$p(\mathbf{y}_0 | \mathbf{y}) \propto p(\mathbf{y} | \{\mathbf{x}_i\}_{i=1}^n) \propto \exp \left[\frac{1}{2\sigma^2} \lambda_u \sum_{i \in U} (y_i - \hat{y}_i)^2 + \frac{1}{2\sigma^2} \lambda_l \sum_{i \in L} (y_i - Y)^2 \right], \quad (10)$$

we have the equivalence between MAP and Eqn. (1).

Score Propagation Interpretation Our algorithm can also be understood as the score propagation between samples. The closed form solution in Eqn. (3) can be written as an iterative solution

$$\mathbf{y}_{t+1} = [(I + \Lambda)^{-1} \tilde{W}] \mathbf{y}_t + (I + \Lambda)^{-1} \Lambda \mathbf{y}_0. \quad (11)$$

It is an efficient method for computing in practice. Note that here we use the random walk Laplacian matrix to show the connection between our algorithm and score propagation. $\tilde{W} = D^{-1}W$ is the transition probability matrix of the random walk.

The proof is as follows.

From the closed form solution, we have

$$\begin{aligned} \mathbf{y} &= (L + \Lambda)^{-1} \Lambda \mathbf{y}_0 \\ &= (I - \tilde{W} + \Lambda)^{-1} \Lambda \mathbf{y}_0 \\ &= [I - (I + \Lambda)^{-1} \tilde{W}]^{-1} (I + \Lambda)^{-1} \Lambda \mathbf{y}_0 \\ &= \lim_{T \rightarrow \infty} \sum_{t=0}^T [(I + \Lambda)^{-1} \tilde{W}]^t (I + \Lambda)^{-1} \Lambda \mathbf{y}_0. \end{aligned} \quad (12)$$

At iteration T the scores are

$$\begin{aligned} \mathbf{y}_T &= [(I + \Lambda)^{-1} \tilde{W}]^T \mathbf{y}_0 + \sum_{t=0}^{T-1} [(I + \Lambda)^{-1} \tilde{W}]^t (I + \Lambda)^{-1} \Lambda \mathbf{y}_0 \\ &\rightarrow 0 + [I - (I + \Lambda)^{-1} \tilde{W}]^{-1} (I + \Lambda)^{-1} \Lambda \mathbf{y}_0 \text{ (when } T \rightarrow \infty) \\ &= (L + \Lambda)^{-1} \Lambda \mathbf{y}_0. \end{aligned} \quad (13)$$

So the iterative solution converges to the closed form solution.

The score propagation equation is just a weighted combination of the scores propagated from neighbors and the initial scores in each iteration, i.e.

$$\mathbf{y}_{t+1}(i) = \frac{1}{1+\lambda_i} \widetilde{W} \mathbf{y}_t(i) + \frac{\lambda_i}{1+\lambda_i} \mathbf{y}_0(i) \quad (14)$$

From this point of view, our algorithm is closely related to previous random walk reranking [7].

3 Implementation Details

We implement a system to test our framework, as shown in Fig. 5. First, we use the text-based search to retrieve a large set of images relevant to the query word given by the user from the web. We simply utilize online commercial search engines at this step. Second, visual features are extracted for all images and a visual similarity graph is built. Then the images are reranked using their distances to the example image selected by the user. Finally we perform graph-based reranking in Section 2. We also implement an user interface for viewing the reranking results and quantitatively evaluating the results.

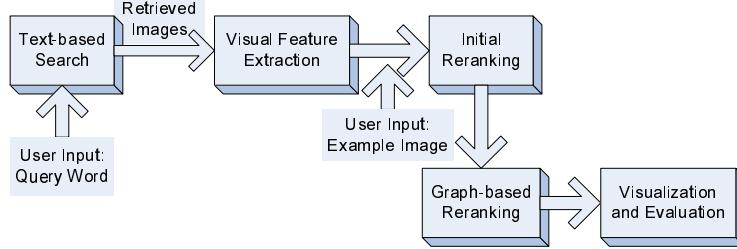


Fig. 5. The diagram of our system.

In this section, we give the details on three remaining problems: visual feature extraction, multiple feature fusion and initial reranking by example.

Visual Feature Extraction In order to quantitatively analyze the visual similarity between two images, we first select a set of features that can effectively characterize an image in different perspectives, such as color, texture and shape. The features we used in our project are listed as follows:

Attention Guided Color Signature. Color signature is first proposed in [12] to describe the overall color distribution in an image. For each image, all the pixels are clustered into several groups according their LAB colors, and the center and density of each color clusters are used as color signature to characterize the color distribution of this image. In this project, we used an attention guided color

signature proposed in [1], which takes the importance of pixels into consideration when calculation the color distribution. It first use an attention detector to compute a saliency map for the image, then perform k -means clustering weighed by this map.

Color Spatialet. Color Spatialet (**CSpa**) is proposed in [1] to describe the spatial distribution of colors in an image. For each image, it is equally divided into $n \times n$ patches, and the main color of each block is calculated as the largest cluster after k-means clustering. The main colors of each blocks, which constitute an n^2 -dim color vector, is the CSpa of this image. n is set to 9 in our experiment. In order to accommodate spatial shifting and resizing, when comparing the similarity of two images, we use following formula to calculate the distance their CSpas A and B :

$$d(A, B) = \sum_{i=1}^n \sum_{j=1}^n \min[d(A_{i,j}, B_{i\pm 1, j\pm 1})],$$

where $A_{i,j}$ denotes the main color of the (i, j) th block in the image.

Wavelet. Wavelet is a widely used features in image processing and computer vision. In this project, we use wavelet the encode the texture information of an image in multiple scales [16]. The 2nd order moments of wavelet coefficients in various frequency bands are used as the wavelet features of an image. We use the L2-distance calculate the distance of wavelet features of two images.

Multi-Layer Rotation Invariant EOH. Edge Orientation Histogram (EOH) proposed in [3] to describe the histogram of edge orientation, which is also a widely used features in computer vision. In order to accommodate the spatial rotation, when calculate the distance of EOH feature vectors, we first rotate one of them to find the best match with the others and then consider this distance as the distance between two EOH feature vectors.

Histograms of Oriented Gradients (HOG). HOG is first proposed in [2] to describe the shape information of an image. An image is divided into several small cells, and histograms of edge orientation is calculated for each cells. With some normalization process, all these histograms constitute the HOG features of an image. HOG features are especially effective if the images contains strong long edges.

Multiple Feature Fusion As we extract multiple features for a single image, we need to fuse them when constructing a single similarity graph to encode similarities between all images. Our method is as follows.

For a single feature k , we first compute pairwise distances $\text{dist}_{ij}^{(k)}$ between images, and then compute the similarity as $w_{ij}^{(k)} = \exp \left[- \left(\text{dist}_{ij}^{(k)} / \sigma_k \right)^2 \right]$.

If we construct multiple graphs for this multiple view semi-supervised learning problem, we have

$$\frac{1}{4} \sum_k \sum_{i,j} w_{ij}^{(k)} [y_i - y_j]^2 + \frac{1}{2} \lambda_u \sum_{i \in U} [y_i - \hat{y}_i]^2 + \frac{1}{2} \lambda_l \sum_{i \in L} [y_i - Y_i]^2, \quad (15)$$

which is equivalent to

$$\frac{1}{2} \sum_k \mathbf{y}^T L^{(k)} \mathbf{y} + \frac{1}{2} (\mathbf{y} - \mathbf{y}_0)^T \Lambda (\mathbf{y} - \mathbf{y}_0). \quad (16)$$

So we simply use the sum of all weights from the multiple graphs as the new weight, i.e. $w_{ij} = \sum_k w_{ij}^{(k)}$, for multiple feature fusion.

Initial Scores We use distance-based reranking in the initial reranking step. First, compute the sum of normalized distances of all features. Then rerank the images using the distances. The initial scores are set as

$$\hat{\mathbf{y}} = \frac{n - \text{rank}(i)}{n}. \quad (17)$$

We find that the above scores are empirically better than distance-based scores.

4 Experiment

In this section, we present the experimental results. We compare our algorithm with the re-ranking results by directly computing distances. We can see that our algorithm provides much more visually smooth and less noisy results.

Our test data are a collection of images associated with several keywords crawled from Microsoft Live Image Search. We manually label the ground-truth of images as either “noise” or “relevant”. Relevant images are further classified into Subclasses (images that are visually similar). We further manually annotate whether each pair of subclasses is semantically relevant.

Given an example image and all ranked images, we use Normalized Discounted Cumulative Gain (NDCG) as our evaluation criteria. NDCG is a normalized DCG by a constant so that the NDCG of optimal ranking list is 1. The definition of DCG is

$$\text{DCG}@k = \sum_{i=1}^k \frac{1}{\log_2(1+i)} (2^{L_i} - 1),$$

where L_i is 2 if image i in the ranked list is in the same subclass of the example image, 1 if it is in a relevant (but not the same) subclass of the example image, 0 if it is in the other subclasses. We choose $k = 100$ as people usually care about the returned images in the first several pages.

The quantitative evaluation results are shown in Table 1. The results are averaged using each of the top 10 text-searched images as the example image separately.

We show some visual comparisons in Fig.s 6– 11.

In Fig. 6, we use the keyword “apple”, and the apple logo as the example image. We can see that our method produces a visually smoother result, and the fruit “apple” is ranked lower than the distance-based method.

Table 1. The quantitative evaluation results.

Query Word	Apple	Tiger	Bear	Seattle	Paris	Lotus	Cougar
Distance-based	0.9175	0.8103	0.8713	0.9608	0.8663	0.8980	0.9174
Ours	0.9306	0.8230	0.8795	0.9680	0.8732	0.9050	0.9217

In Fig. 7, we use the keyword “apple”, and the apple product “ipod” as the example image. Our method’s result has no noisy image, while the distance-based method has one.

In Fig. 8, we use the keyword “bear”, and the white bear as the example image. Our method’s result has only one noisy image, while the distance-based method has two, and one noisy image is even ranked as the top one.

In Fig. 9, we use the keyword “seattle”, and the night scene of the city seattle as the example image. Our method’s result has no noisy image, while the distance-based method has two.

In Fig. 10, we use the keyword “tiger”, and the animal “tiger” as the example image. Our method’s result has only one noisy image, while the distance-based method has two. In addition, the image ranked second by the distance-based method is not visually similar to the example image, although relevant.

In Fig. 11, we use the keyword “paris”, and an actress named Paris as the example image. Our method’s result has no noisy image, while the distance-based method has one.



Top 20 results by computing distances



Top 20 results by our method

Fig. 6. Comparison on results of keyword “apple”. The orange box and the light blue boxes indicate the example image and noisy images, respectively.



Top 20 results by computing distances

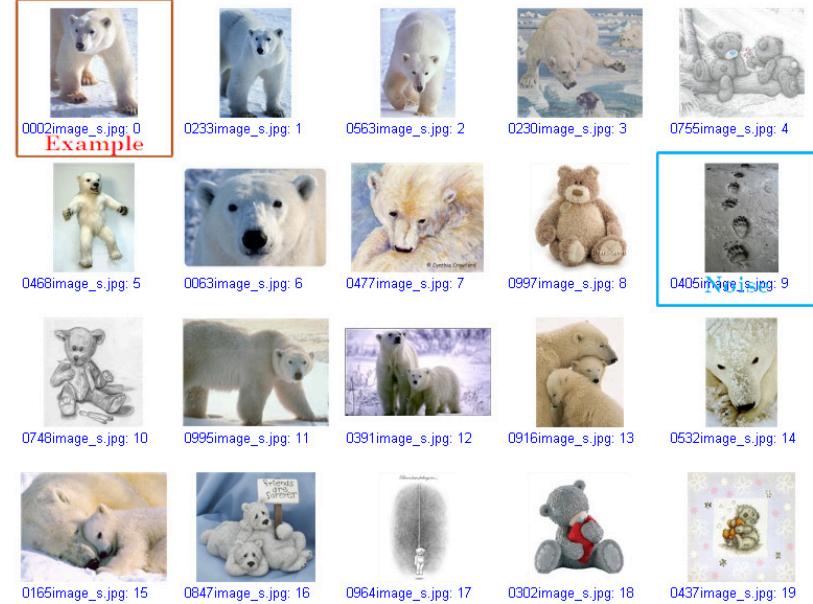


Top 20 results by our method

Fig. 7. Comparison on results of keyword “apple”. The orange box and the light blue boxes indicate the example image and noisy images, respectively.

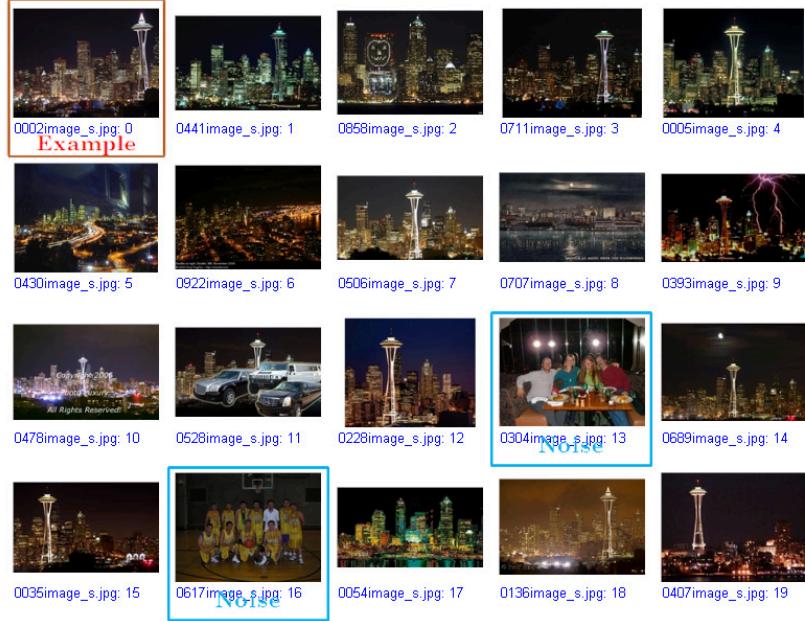


Top 20 results by computing distances

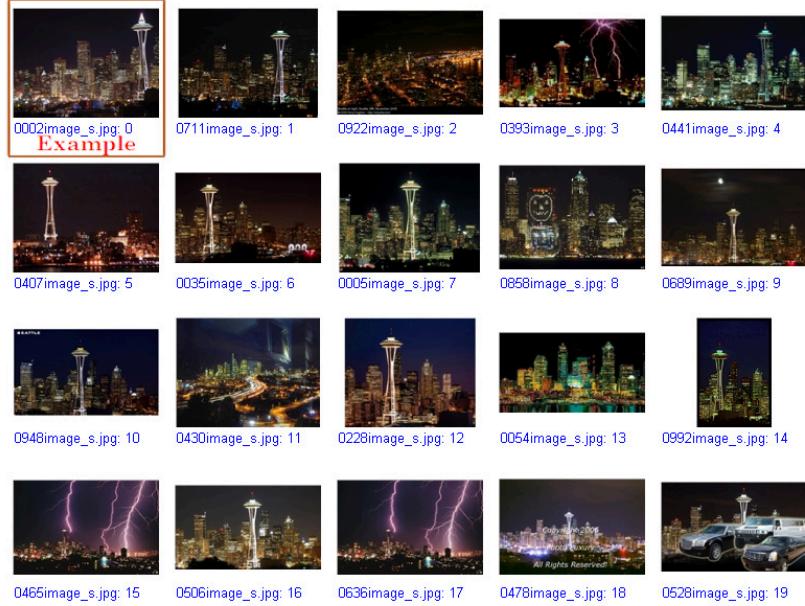


Top 20 results by our method

Fig. 8. Comparison on results of keyword “bear”. The orange box and the light blue boxes indicate the example image and noisy images, respectively.



Top 20 results by computing distances

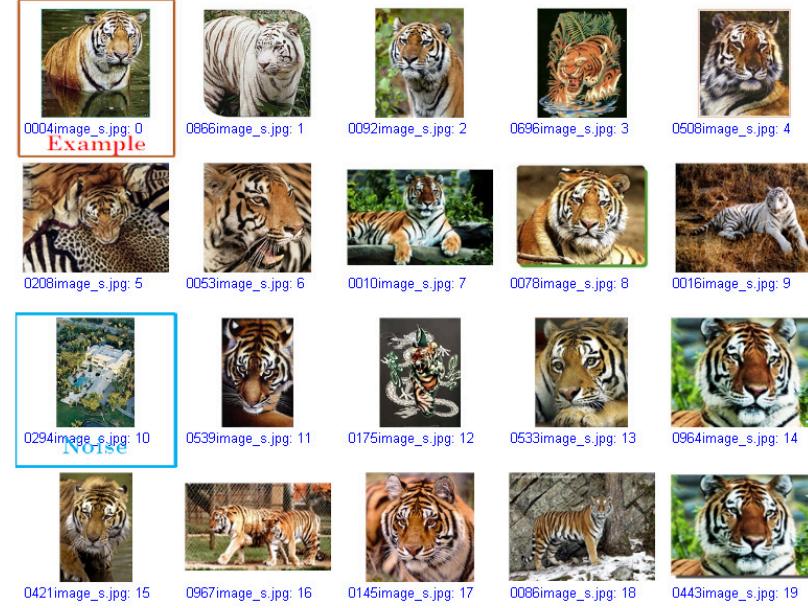


Top 20 results by our method

Fig. 9. Comparison on results of keyword “seattle”. The orange box and the light blue boxes indicate the example image and noisy images, respectively.

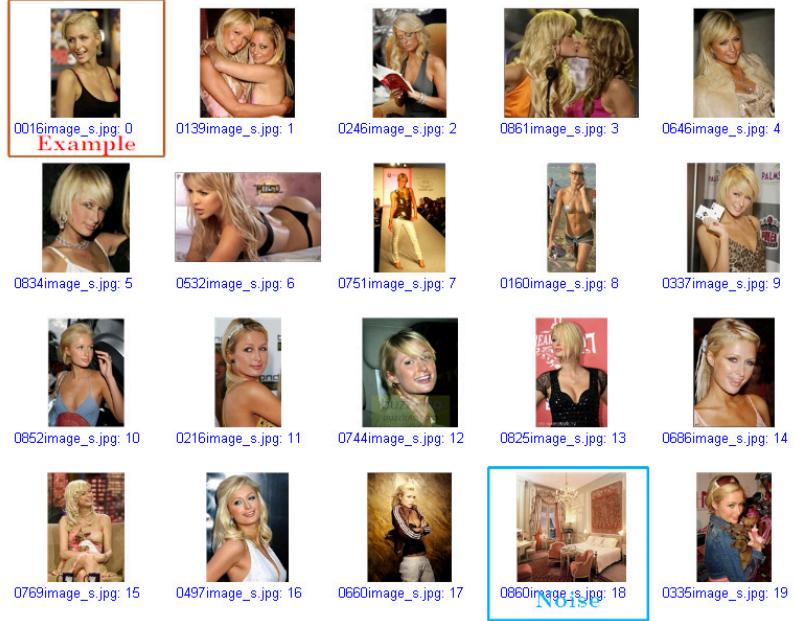


Top 20 results by computing distances



Top 20 results by our method

Fig. 10. Comparison on results of keyword "tiger". The orange box and the light blue boxes indicate the example image and noisy images, respectively.



Top 20 results by computing distances



Top 20 results by our method

Fig. 11. Comparison on results of keyword “paris”. The orange box and the light blue boxes indicate the example image and noisy images, respectively.

5 Conclusion and Future Work

In this project, we proposed to utilize both the visual information and user interaction to rerank the images returned by text-based search. Previous methods either only use visual information or only use an example input by the user. We formulate the problem as graph-based semi-supervised learning. We further extends it to multiview learning for multiple feature fusion. In this problem, a bottleneck may be that we only have one example image for semi-supervised learning. Our future work may be to explore better methods for multiple feature fusion, such that the challenging problem of semi-supervised learning using very few labeled samples can be well solved.

References

1. J. Cui, F. Wen, and X. Tang. Real time google and live image search re-ranking. In *Proceeding of the 16th ACM international conference on Multimedia*, pages 729–732, 2008.
2. N. Dalai, B. Triggs, I. Rhone-Alps, and F. Montbonnot. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, volume 1, 2005.
3. W. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, volume 12, 1995.
4. M. Hein, J. Audibert, and U. Von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8:1325–1368, 2007.
5. W. Hsu, L. Kennedy, and S. Chang. Video search reranking via information bottleneck principle. In *Proc. ACM International Conf. on Multimedia*, 2006.
6. W. Hsu, L. Kennedy, and S. Chang. Video search reranking through random walk over document-level context graph. In *Proc. ACM International Conf. on Multimedia*, 2007.
7. Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1877–1890, 2008.
8. J. Liu, W. Lai, X. Hua, Y. Huang, and S. Li. Video search re-ranking via multi-graph propagation. In *Proc. ACM International Conf. on Multimedia*, 2007.
9. Y. Liu, T. Mei, X. Hua, J. Tang, X. Wu, and S. Li. Learning to video search rerank via pseudo preference feedback. In *IEEE International Conf. on Multimedia and Expo*, 2008.
10. A. P. Natsev, A. Haubold, J. Tešić, L. Xie, and R. Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *Proc. ACM International Conf. on Multimedia*, pages 991–1000, 2007.
11. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1998.
12. Y. Rubner, L. Guibas, and C. Tomasi. The earth movers distance, multi-dimensional scaling, and color-based image retrieval. In *Proceedings of the ARPA Image Understanding Workshop*, pages 661–668, 1997.

13. J. Tang, X.-S. Hua, G.-J. Qi, M. Wang, T. Mei, and X. Wu. Structure-sensitive manifold ranking for video concept detection. In *Proc. ACM International Conf. on Multimedia*, pages 852–861, 2007.
14. J. Tang, X.-S. Hua, G.-J. Qi, and X. Wu. Typicality ranking via semi-supervised multiple-instance learning. In *Proc. ACM International Conf. on Multimedia*, pages 297–300, 2007.
15. X. Tian, L. Yang, J. Wang, Y. Yang, X. Wu, and X. Hua. Bayesian video search reranking. In *Proc. ACM International Conf. on Multimedia*, 2008.
16. M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on image processing*, 4(11):1549–1560, 1995.
17. R. Yan, A. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. *CIVR*, pages 238–247, 2003.
18. X. Yuan, X.-S. Hua, M. Wang, and X.-Q. Wu. Manifold-ranking based video concept detection on large database and feature pool. In *Proc. ACM International Conf. on Multimedia*, pages 623–626, 2006.
19. D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, 2004.
20. X. Zhu. Semi-supervised learning literature survey. Technical report, Computer Science, University of Wisconsin-Madison, 2006.