Long Term Learning for Content Extraction in Image Retrieval

Pradhee Tandon and C. V. Jawahar
Center for Visual Information Technology
International Institute of Information Technology
Hyderabad-500 032, INDIA
{pradhee@research.,jawahar@}iiit.ac.in

Abstract

Learning in the form relevance feedback is popular for bridging the semantic gap in content based image retrieval (CBIR). However, learning across users and sessions did not get its due attention in CBIR. In this paper, we propose a computationally simple yet effective scheme for learning relevant features for a specific image. Learned concept is related to the spatial distribution of pixels to identify the pixels/regions which contribute to the semantic content of the image. This learning also makes the retrieval more accurate. Experimental studies validate the applicability, both qualitatively and quantitatively.

1 Introduction

Content based image retrieval (CBIR) has received considerable attention in the last decade [1]. In CBIR, typically, an image is represented as a feature vector, and retrieval is done by finding the most similar images in the database. However, human perception of similarity could be considerably different from that a machine could directly compute, resulting in a semantic gap. This necessitated learning schemes for finding the appropriate distance metrics [3] as well as features [4, 9]. Relevance feedback schemes [4, 9] learned the useful features with the help of user interactions. User refines the retrieved results, and indirectly contributes to the feature selection in these approaches. However, this class of learning schemes lack in memory. They often start from the same state (of all features being equally relevant) and learn to improve the precision within a user-session. This class of learning schemes are referred to as short term learning (STL), since the learning is limited to a user session.

A complementary paradigm, which has received some attention in the recent years is Long Term Learning (LTL) in CBIR. The focus has been primarily on extending the knowledge learned from user interactions to unseen images with the help of techniques like LSI [2, 7, 8]. This class of algorithms have been motivated by the use of the factorization schemes in text retrieval literature. However, most of these schemes require huge memory and computation to extend the knowledge across images. Another class of LTL approaches, which got motivated by the web mining approaches, archive the user-logs and thereby achieve learning across sessions. This class of algorithms use the co-occurrence patterns of images to learn. They typically need considerable user-logs to result in useful learning.

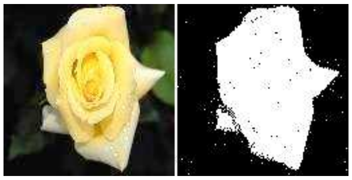


Figure 1: (a) A flower in the hedge and (b) the learned content

In this work, we explore the possibility of a long term learning scheme which can seamlessly merge with the traditional CBIR with relevance feedback. We would like to retain (or use) the valuable user inputs obtained through user feedbacks, to learn across sessions. Such a learning scheme should be: (i) Incremental in nature. It should improve with every session, but with minimal computations. (ii) It should be content-specific. Learning should result in enhancing the image representations and thereby reducing the semantic gap. (iii) Transparent to the retrieval. Such a learning should be transparent to the retrieval process. Retrieval should be possible even when the content is not learned completely.

Our solution is motivated by some of the simple, but powerful techniques in text retrieval. While indexing textual documents, not all words are found to be equally relevant or useful for the retrieval tasks. There exist effective techniques for capturing the key words, given a collection of documents [6]. Borrowing this idea, we find a set of fea-

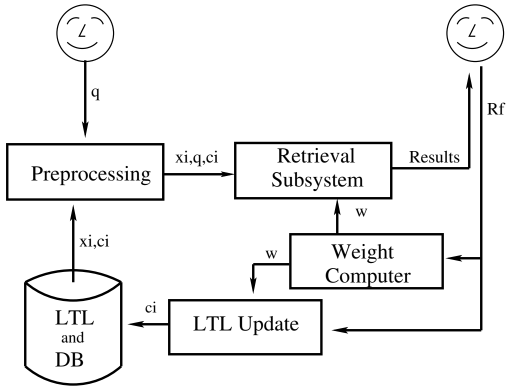


Figure 2: Long Term Learning in Retrieval

tures which could be informative for the retrieval task. From the pattern classification point of view, these are more of discriminative features. However they are calculated with computationally efficient techniques which avoid factorizations and eigenvector computations. In addition, when the retrieval takes place (with or without relevance feedback), the relevance of the features to a specific image is learned, across sessions, to further refine the keywords (or discriminative features).

One of the key advantages of our work is that, it allows to discover the content in the images over time. Given an image of a flower in a garden, our approach allows to learn the pixels corresponding to the flower without any explicit segmentation or user interaction at the pixel-level, provided that multiple users have retrieved this image, while searching for the flower. Figure 1 gives an example of the content learned over time. The brightness of the pixel is proportional to the utility of the content.

2 Discriminative Long Term Learning

In CBIR, an image is represented using a feature vector of automatically extracted visual characteristics. Let $\mathbf{x_i} = [x_{i1}, \ldots, x_{ij}, \ldots x_{iN}]^T$ be the feature vector corresponding to image i in the database. When an example image (query) is given, its corresponding feature representation $\mathbf{q}$ is computed, and a set $\mathcal{R}$ of images with minimal distance ($d(\mathbf{x_i}, \mathbf{q})$) to $\mathbf{q}$ is treated as optimal for retrieval. It has been argued that not all feature dimensions are equally useful for the distance/similarity computation. Relevance feedback based approaches estimate the importance of features to the query concept in terms of weights for each dimension. This relevance ($\mathbf{w}$) is obtained through continued user interactions. This is then used in the distance computation $d(\mathbf{x_i}, \mathbf{q}, \mathbf{w})$.

The success of relevance feedback comes from the fact that not all features are relevant for a given query. However, a relatively unnoticed fact has been that not all features are helpful in characterizing the semantic content of a given image. For example in Figure 1 (a), the yellow flower is the useful (or popular) content rather than the green leaves around it. In this paper, we argue that such content can be automatically characterized from the history of interactions, and there after used in image retrieval.

Let $\mathbf{c_i}$ be the relative importance of features of image i. Then a better semantic similarity can be computed as

$$d_i = f(\mathbf{x_i},\, \mathbf{q},\, \mathbf{w},\, \mathbf{c_i}) \tag{1}$$

There are two possible clues which could allow us to build an estimate of $\mathbf{c_i}$: (a) the similarities and dissimilarities of images within a database (b) Past user preferences in terms of acceptable and non-acceptable images to a given query.

Given a collection of images, there is some amount of inherent information in it describing the content in the constituent images. The features which are prominent in one image and those that are not prominent in other images would be more relevant and should be weighted higher while computing the semantic similarity. Such an estimate of the relevance of features to images can be *apriori* computed and used for image retrieval. In the case of relevance feedback, user feedback results in two sets of images: relevant and irrelevant images. The goal is then to emphasize features which selectively prefer relevant images and then remember and reinforce them for the future sessions. This relative importance is captured well by the consistency of features across the relevant samples, and discriminability of irrelevant examples from relevant examples. The consistency in features is characterized by their relative low variance. Therefore the idea is to emphasize those features which show high consistency over the relevant set and high variance over other images.

There are two possible possible modes in which image retrieval typically takes place. In the first category, a query (text or image) is given and a set of relevant images are retrieved. User accepts (selects) some of the images and thus gives an indirect feedback. The second popular approach is to use relevance feedback and along with query-by-example. In this case, user gives explicit feedback to identify positive and negative images. Both these methods can be understood as a process of splitting the retrieved images $\mathcal{R}$ into two subsets $\mathcal{P}$ and $\mathcal{N}$.

We argue that a ratio of the inverse of the variance (or any other similar measure of dispersion) of a feature over the relevant set to the inverse of its variance over the irrelevant set captures the utility of the feature for the retrieval task. Given a set of relevant and irrelevant images, such a measure could be computed for each individual feature as

$$s_j = \left[\frac{\sigma_{j_{\mathcal{N}}}}{\sigma_{j_{\mathcal{P}}}}\right] \tag{2}$$

where σ captures the measure of dispersion. Here s_j is only an instantaneous estimate of the importance of the feature. It changes with user feedback. Note that this is computed for individual features.

$$\sigma^2_{j\mathcal{N}} = \frac{1}{|\mathcal{N}|}\sum_k (x_{kj} - \mu_j)^2$$

where $\mathbf{x_k} \in \mathcal{N}$. One could also think of using other similar measures. A similar approach has been shown to be promising in text retrieval research. There, the idea has been to select key words [6], i.e., the terms which selectively or discriminatively describe the current document with respect to others.

In addition to selecting such key words, there is another aspect which is very relevant to the text processing community. It is the removal of *stop words* from text. Stop words are words which are common in a majority of the documents and lack any descriptive capacity. They could come from the apriori information coming out of languages or the statistics/distribution of words in the database. In images, these correspond to features which show similar variation over all images and thus should be de-emphasized by the formulation with there weights ideally set close to *zero*. This can be efficiently accomplished by using a modified formulation where we take the *logarithm* of the ratio of inverse of variances discussed earlier. The *logarithm* ensures that the features which show similar variation over the relevant set and the other images are weighted to zero. The modified formulation can be presented as:-

$$s_j \;=\; \log\left[\frac{\sigma_{j\mathcal{R}}}{\sigma_{j\mathcal{P}}}\right] \tag{3}$$

First we explain, how the relative of importance of features $\mathbf{s}$, which gets accumulated over iterations, can be used for computing the weight $\mathbf{w}$ in a relevance feedback framework, and then how to incrementally use them for computing $\mathbf{c_i} = [c_{i1}, c_{i2}, \ldots c_{iN}]^T$. The estimate of the importance of the feature s_j can be used for incrementally updating the weight, w_j, for the corresponding feature as in the equation below, which can then be used for tuning the comparison metric,

$$w^t_j \;=\; w^{t-1}_j \;+\; s_j \tag{4}$$

where w_j represent the weight for feature j after the iterations t and $(t\,-\,1)$. One could also add a learning rate to control the rate across iterations. Here we had shown how the relative importance can be captured within a user session. In a non-iterative mode, this relative importance can be directly computed as we explain below. However, our objective is to use these weights for computing the content vector $\mathbf{c_i}$ corresponding to the image i.

At the end of the session the weight vector is used for updating the content weights for the relevant images of this session as long term learning from this query and is then memorized for future use as in

$$c_{ij} \;=\; c_{ij} \;+\; \rho\, w_j \tag{5}$$

Here w_j is the weight after the final user iteration, c_{ij} represents the relative importance of feature j in image i and reflects the relevant content learned by the system over all previous queries. ρ slows the learning based on the number of past sessions. When many users access and accept an image for a specific feature or features, we indirectly conclude that these features are important for that image.

However, in scenarios where iterative feedback from the user is missing, there is no incremental learning of the features relevant to the query. Such is the case even with popular web-based image search engines. Here, when the images are indexed by surrounding textual content, this method can be employed. This is, in a way, similar to the standard text processing scenario where given a database or a collection of documents the selective terms for all documents are to be estimated. Here the idea is to emphasize those features which better discriminate the relevant samples with respect others. On the lines of the method discussed above for the iterative feedback based approaches, we expect the consistency of the features over the relevant set and their variation over the irrelevant images makes them relevant for the images and vice versa.

With little adaptation our earlier formulation for iterative feedback in Equation 3 fits the requirements as in

$$w_j \;=\; \log\left[\frac{\sigma'_{j\mathcal{R}}}{\sigma'_{j\mathcal{P}}}\right] \tag{6}$$

where $\sigma'_{j\mathcal{R}}$ and $\sigma'_{j\mathcal{P}}$ denote the dispersions of feature j over the $\mathcal{R}$ and $\mathcal{P}$ sets. The weights thus learned are then used for incremental long term learning as in Equation 5.

Our incrementally improving approach to long term learning allows flexibility in the learning, controlled in rate by factors like ρ. This ensures convergence of long term learning to the generally acceptable content in the image. Our incremental learning approach has a distinct advantage in terms of its computational expense, allowing efficient long term learning. The incremental nature makes it independent of available archived logs of feedback etc. It is also independent of the query so it performs irrespective of the availability of iterative relevance feedback. This allows it to perform independent of the retrieval approach employed by the system thus making it a highly portable approach for long term learning. Such unique characteristics of our approach make it an inexpensive and effective approach for long term learning of content.

3 Experiments and Discussion

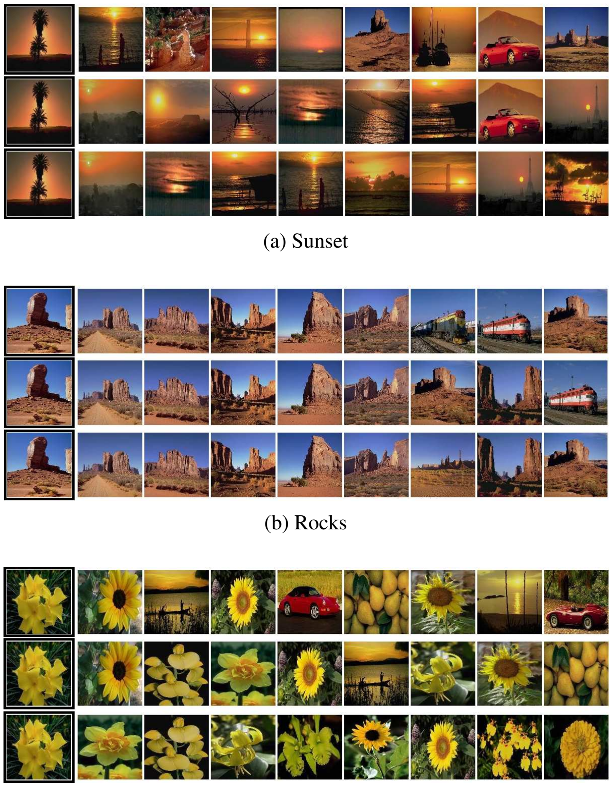
We have conducted extensive experiments to validate the applicability of the proposed long term learning approach. In this section, we demonstrate the improvement in *precision* with our learning. As as baseline, we compare our performance with a system which has no long term learning. Our CBIR implementation supports query by example, and content represented using a subset of MPEG-7 visual descriptors [5]. This implementation also supports relevance feedback for intra-session learning.

The MPEG-7 features we used, primarily capture the color and texture of the image. We have incorporated Color Moments, Color Layout Descriptor(CLD) and Color Structure Descriptor(CSD) from MPEG-7 into our representation. We have used the top three color moments namely, mean, variance and skew. CLD first breaks up the image into 8×8 blocks and extracts the dominant color in YCbCr space for each of these and forms a 8×8 pseudo-image. It then computes a DCT on this small image for each channel. After scaling the coefficients with a standard matrix we choose the top few coefficients for each channel in zigzag scan order. CSD slides a 8×8 window over the image and computes a occurrence based frequency histogram of the image. We use a combination of the above descriptors for our feature vector.

We have used a diverse set of images for all our experiments. The set consists of around 1000 real images with about 100 from each of the categories including trains, surfers, hills, cars, sunset images, flowers etc. and thus vary in their visual content.

We have used the improvement in precision percentage across sessions to demonstrate the effectiveness of our proposed long term learning method. We have conducted experiments to show the performance in both presence and absence of iterative relevance feedback. For the iterative feedback based experiment we have randomly picked set of 20 queries while ensuring equal representation from all categories. We experimented for 20 sessions with 5 iteration for each. In each iteration, we gave feedback on the top 48 retrieved images. The system estimates the feature weights and performs retrieval using them. After 5 iterations, these weights update the long term learning for the relevant images. This should result in a characteristic gain in precision in the first iteration of next session. We averaged the percentage precision over the queries and have included some randomly sampled instances in Table 1. As expected, our approach shows improvement in performance as sessions progress, in comparison to the approach without LTL.

In the next experiment, we show how our approach improves retrieval even in absence of incremental feature learning. We use the same random set of 20 queries and run the system for 20 sessions each with only one round of feedback, the first one. As a result relevant and irrelevant subsets of the retrieved set are formed. Using these, our approach computes the update to the content vector in long term learning. The average percentage precision for some randomly sampled sessions is compared to the LTL free approach in Table 2. The improved performance again validates our approach.


(a) Sunset

(b) Rocks

(c) Flower

Figure 3: Top 7 results for 3 queries for 3 different sessions.

Session	**1**	**2**	**5**	**10**	**20**
noLTL	58.7	58.7	58.7	58.7	58.7
Ours	58.7	63.7	69.5	71.6	73.8

Table 1: Percentage precision with iterative relevance feedback.

We have also included some visual results for a few queries from the database as shown in Figure 3. We show the top 7 results for 3 sessions for each query. The marked image on the left is also the query. The improving results in the subsequent rows show the gain with our approach.

These results validate the ability of our proposed long term learning method to improve retrieval accuracy.

Session	**1**	**2**	**5**	**10**	**20**
noLTL	58.7	58.7	58.7	58.7	58.7
Ours	58.7	64.5	70.8	74.1	76.6

Table 2: Percentage precision in absence of iterative relevance feedback.

4 Content Extraction

In most of the learning solutions in CBIR, validation of the idea is done using the performance improvement typically measured as precision. There is no explicit method for validating whether the semantic gap is really getting bridged. In case of user feedback based learning, right content is in agreement with the feedback of a majority of users. Long term learning provides us an estimate of the importance of specific features to the content in the image using $\mathbf{c_i}$. Every pixel in the image has contributed to the feature vector description of the image. Now, in a class of situations, the relative importance of a feature c_{ij} could be mapped back to the image. i.e., the relevance of a pixel (or region) to the semantic content of the image can be computed. To simplify lets assume that the feature is primarily a color histogram. Then the estimated content $C_{mn} = c_{ij}$ iff I_{mn} is j, where I_{mn} is the color of the pixel (m, n). This allows distribution of the estimate of the content to the constituent pixels which have contributed to the specific feature. All pixels are similarly assigned a relevance based gray value. This image shows the most relevant as the brightest regions, performing a naive *region of interest* extraction. Figure 4 shows how the learning based content identification improves with sessions for a few sample images (left-most), from no information (in second image) towards *region of interest* extraction (in the right-most).

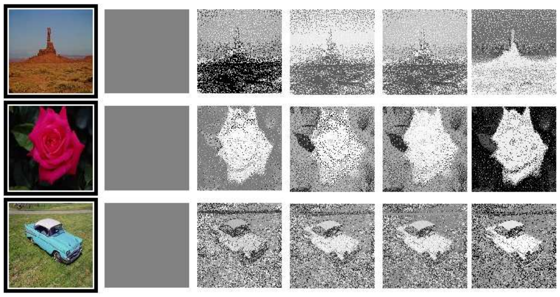


Figure 4: Content regions emerge over sessions(from *l* to *r*)

Next we present some images with their corresponding content images to validate our approach over some varied concepts in Figure 5. Our approach for content presentation also allows visual evaluation of the performance of long term learning.

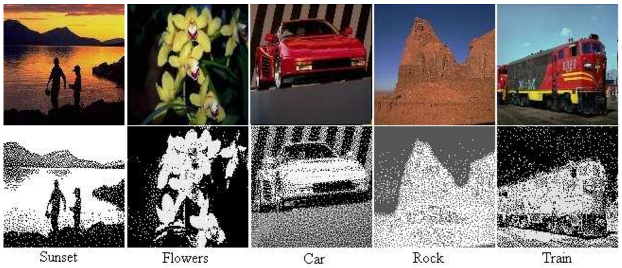


Figure 5: Content extracted from sample images using our proposed long term content learning algorithms.

5 Conclusion

In this paper we have derived inexpensive incrementally learning solutions for long term learning motivated by pioneering ideas in text processing. We have proposed methods which work independent of the retrieval approach, making them highly portable. Our experiments show the effectiveness of our proposed approach on real datasets. We also visually prove the correctness of our learning based content extraction. In future we would like to extend our approach to also handle multiple concepts in images.

References

[1] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Survey*, 40(2):1–60, 2008.

[2] D. Heisterkamp. Building a latent semantic index of an image database from patterns of relevance feedback. *Proc. of Intl Conf on Patt. Recog.*, 4:134–137, 2002.

[3] S. C. Hoi, W. Liu, and S.-F. Chang. Semi-supervised distance metric learning for collaborative image retrieval. *Conf. on Comp. Vision and Patt. Recog.*, pages 1–7, June 2008.

[4] T. Huang and X. S. Zhou. Image retrieval with relevance feedback: from heuristic weight adjustment to optimal learning methods. *Proc. of the Intl Conf on Image Proc.*, 3:2–5, 2001.

[5] J. M. Martínez. Mpeg-7: Overview of mpeg-7 description tools, part 2. *IEEE MultiMedia*, 9(3):83–93, 2002.

[6] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, pages 513–523, 1988.

[7] A. Shah-Hosseini and G. M. Knapp. Learning image semantics from users relevance feedback. In *Proc. of ACM Multimedia*, pages 452–455, New York, USA, 2004.

[8] T. Yoshizawa and H. Schweitzer. Long-term learning of semantic grouping from relevance-feedback. In *Proc of the Intl Workshop on MIR*, pages 165–172, NY, USA, 2004. ACM.

[9] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, April 2003.