International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

Feature Based Image Reranking Using Fusion Weights

Minakshi Somanath Bagad[1], Sonal P. Patil[2]

[1]M.E (C.S.E) Student, Department of Computer Science, G.H.R.I.E.M, Jalgaon, Maharashtra, India
[2]Assistant Professor, Department of Computer Science, G.H.R.I.E.M, Jalgaon, Maharashtra, India

Abstract: ***Search reranking is considered as a best and common way to improves retrieval precision. 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. Image re-ranking, as an effective way to improve the results of web-based image search however the problem is not trivial especially when we are considering multiple features or modalities for search in image and video retrieval. This paper suggests a new kind of reranking algorithm that supports the mutual exchange of information across multiple modalities for improving search performance and follows the philosophy of strong performing modality could learn from weaker ones.***

Keywords: Image Reranking, image retrieval, Modality, SVM

1. Introduction

Searching for relevant images from large scale community databases given a query term is an important task. The image ranking approach represents an image collection as a graph that is built using multimodal similarity measures based on visual features and user tags. To improve the performance of this image search image re-ranking technology is used. Search re-ranking is regarded as a common way to boost retrieval precision. The problem nevertheless is not trivial especially when there are multiple features or modalities to be considered for search, which often happens in image and video retrieval. Different re-ranking algorithms are available in computer world which gives different precisions. Formally; the definition of the re-ranking problem with a query image is as follows. 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 features such as color, Shape, Texture from each image.

2. Working Principle of System

Image retrieval systems have certain drawbacks like images obtained are many a time duplicated, of low precision, and irrelevant. This scenario may occur due to sparse and noisy query. Due to this aspect user cannot be always sure of perfect images being obtained in available time. Many a times user has to surf many pages of images available to land at the perfect one. This possesses a great threat to the fast technology.

2.1 Working methodology

The proposed work in this respect is as follows;

1. Rerank the images obtained on user side
2. Use highly efficient reranking algorithm to facilitate grouping of similar images considering multiple features score one at a time and select perfect among them
3. Re rank the images again by considering the user feedback
4. Use various concepts in combination to get an excellent image retrieval system

If the entered query is "sunset", color should be the considered feature as color is the primary identifier. For "building" shape as a feature rather than color is appropriate. Whereas, for "snow" if color and shape is considered then differentiation between "snow" and "cotton" would become difficult for the system. Thus, texture will become the primary identifier for "snow" and not colour or shape. The large image collection is subjected to feature extraction process where the attributes of the image both visual such as colour, texture and shape and semantic such as intentional, clicks, labels etc. are extracted from the feature database using appropriate methods.[8]

The query image can be any of the popular formats. The query image is subjected to feature extraction process and query features are obtained.

In similarity measurement process, the query's feature is compared with the features stored in feature database. The distance between the two features is calculated and weights are determined. The output images are then sorted and ranked, so that most similar images can be displayed to the user.[11]

Re ranking at the first stage is based on the features extraction Re ranking at the next level will be based on the user feedback

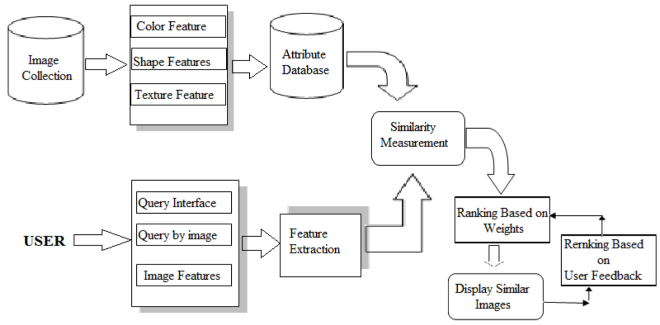

Figure 1: Feature based image Re-ranking Framework

2.2 Dataset Description

In this system we are going to use WANG database or initial image collection and experiments. The WANG database is a subset of 1,000 images of the Corel stock photo database which have been manually selected and which form 10 classes of 100 images each. The WANG database can be considered similar to common stock photo retrieval tasks with several images from each category and a potential user having an image from a particular category and looking for similar images which have e.g. cheaper royalties or which have not been used by other media. The 10 classes are used for relevance estimation: given a query image, it is assumed that the user is searching for images from the same class, and therefore the remaining 99 images from the same class are considered relevant and the images from all other classes are considered irrelevant.

3. Feature Extraction

Feature extraction is a means of extracting compact but semantically valuable information from images. This information is used as a signature for the image. Similar images should have similar signatures. Furthermore, we can take the size of the objects in the image into account.

We have considered three features: texture, shape, and color, which are used to compare the images.

3.1 Colour Features

Colour is linked to the chromatic part of an image. A colour histogram provides allotment of colours which is achieved by damaging image colour and plus how many numbers of pixels fit into every colour. For the whole collection every image's colour histogram is examined and saved in the database. Retrieval of those images has been done in the matching process whose colour allotment matches to the example query very much.

We are going to use color moments for color feature extraction.

Moment 1: Mean

The first color moment can be interpreted as the average color in the image, and it can be calculated by using the following formula:

$$Ci = \frac{1}{N}\sum_{j=1}^{N} \text{Pij} \ ()$$

P_{ij} is the color value of the i-th color component of the j-th image pixel and N is the total number of pixels in the image.

Moment 2: Standard Deviation

The second color moment is the standard deviation, which is obtained by taking the square root of the variance of the color distribution. It can be calculated using following formula:

$$\sigma i = \sqrt[2]{\frac{1}{N}\sum_{j=1}^{N}(\text{Pij} - \text{Ci})^2} \ (1)$$

where E_i is the mean value, or first color moment, for the i-th color channel of the image.

Algorithm: Extract Color Feature

1. **INPUT:** Take the query image as well as image in the database which is RGB color space
2. Calculate the color moments that is Mean and Standard Deviation for each image using equations
3. **OUTPUT**: The color feature vector

3.2 Texture Features

By dissimilarity in brightness with high frequencies in the image spectrum, textures are characterized. While making a distinction between areas of the images with same colour, these features are very useful. Measures of image texture such as the degree of contrast, coarseness, directionality, regularity and randomness can be calculated using second-order statistics.

We are going to use Entropy function for calculating values for texture feature.

E = entropy (I) (2)

returns E, a scalar value representing the entropy of grayscale image I. Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image.

Algorithm: Extract Texture Feature

1. **INPUT:** The query image and image in the database.
2. Capture texture feature i.e. Entropy for all images.

3.3 Shape Feature

By either the global form of the shape or local elements of its boundary, shape features can be differentiated. Global form of the shape: like the area, the extension and the major axis orientation. Local elements of its boundary: like corners, characteristic points or curvature elements. The degree of similarity between two shapes is evaluated through standard mathematical distances measures, like Euclidian distance between two points. The capability of shape features to tolerate semantic significance can be used for semi-automatic extraction of high-level features of multimedia data by providing characteristic shapes for special real-world-objects.

Skewness, kurtosis, moment are found to be best measures for retrieving and reranking images. Based on scores/weights given by these methods weights images can be ranked efficiently.

a) Skewness

The skewness of a distribution is calculated by formula

$$s = \frac{E(x-\mu)^3}{\sigma^3} \quad (3)$$

where μ is the mean of x, σ is the standard deviation of x, and E(t) represents the expected value of the quantity t. skewness computes a sample version of this population value.

b) Kurtosis

The kurtosis of a distribution is defined

$$k = \frac{E(x-\mu)^4}{\sigma^4} \quad (4)$$

where μ is the mean of x, σ is the standard deviation of x, and E(t) represents the expected value of the quantity t. kurtosis computes a sample version of this population value

.

c) Moment:

The central moment of order k of a distribution is defined as

$$m_k = E(x-\mu)^k \quad (5)$$

where E(x) is the expected value of x.

Algorithm: Extract Shape Feature

1. **INPUT:** Take the query image as well as image in the database which is RGB color space
2. Calculate skewness for each image using equations.
3. Calculate kurtosis for each image using equations.
4. Calculate moment for each image using equations.
5. **OUTPUT:** The Shape feature vector

3.4 Relevance Feedback

Relevance feedback is a technique that takes advantage human-computer interaction to refine high level queries represented by low level features. The weights for the low-level feature, i.e. color, shape and texture etc. are dynamically updated based on the user's feedback.

Algorithm: Get Final ranked list based on Relevance Feedback

1. **INPUT**: A query image.
2. Extract color and texture feature from the image.
3. Find the matching images from image database by fusing different modalities.
4. Get the positive examples.
5. Rerank query result. Display the relevant images from database.
6. **OUTPUT:** Find the top relevant images.
7. Collect feedback from user.
8. If user is not satisfied then repeat the steps i.e. try to Re-rank again.
9. If user is satisfied then Stop and return the Re-ranking result to user.

4. SVM Classifier

SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Steps to be followed by SVM classifier are as follow:

1. Set up the training data
2. Set up SVM's parameters
3. Train the SVM

We call the method SVM::train to build the SVM model.

4. Regions classified by the SVM
5. Get Support vectors

Algorithm: Classification of images according to various features using SVM

1. **INPUT:** Matrix of training data
2. Train support vector machine classifier
3. **Initialize** i=1, k=1
4. For i= 1:size(Trainingg,2) Training(i,:) = Trainingg{1,i};
5. for k=1
   computesvmstruct
   svmstruct= svmtrain(Training,Group)
6. To classify new data, use the result of training, SVMStruct, with the svmclassify function
   Svmclassify (SVMStruct, Sample)
7. **End for**
8. **OUTPUT** : SVM classified groups i.e. vectors

5. Comparative Result of Ranking with Different Iterations

In this section results of different iterations of getting ranked images are compared. As we allow a user to refine results till the user get satisfied for ranked images. As user clicks repeatedly on Refine result button various iterations are observed for the same so it becomes the Reranking process. From that it is observed that different numbers of iterations are required for various categories of images.

The comparative results of different images and different iterations are as follow.

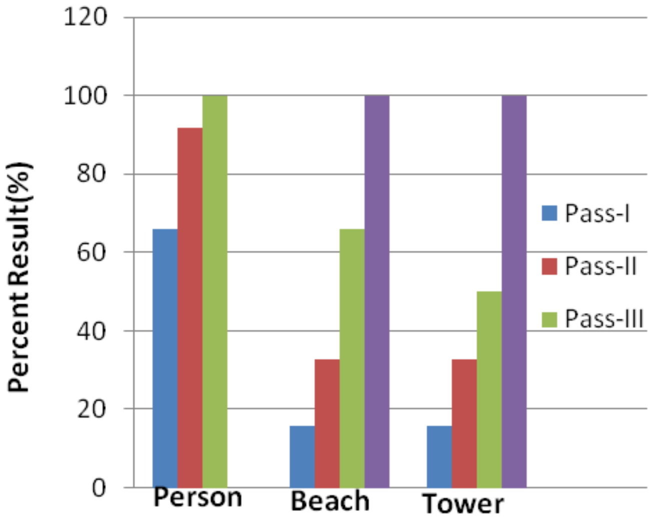

Figure 2: Comparison of various iterations while Re-ranking images

Here when user select query image from category-I from dataset user get the final reranked list with three iterations with user feedback, when user select query Image of beach scene user get the results with four iterations.

6. Conclusion

The purpose of Reranking is to present an image conceptually and re-rank the images according to specified features, with a set of low-level visual features such as color, texture, and shape. We have used the same three modalities as Shape, color and texture. Any image retrieval and reranking technique gives more accurate result with multiple features than with a single feature Relevance feedback techniques were incorporated into CBIR such that more precise results can be obtained by taking user's feedbacks into account. Existing relevance feedback-based CBIR methods usually request a number of iterative feedbacks to produce refined search results. So we use Relevance Feedback to achieve the high Re-ranking results images using three modalities.

References

[1] Vidit Jain and Manik Varma, "Learning to re-rank query – Dependent Image Re-ranking using Click data" 2011
[2] C. A. Bhatt and M. S. Kankanhalli, "Multimedia data mining: State of the art and challenges," Multimedia Tools Appl., vol. 51, no. 1, pp. 35–76, 2011.
[3] X.-Y. Wei, Y.-G. Jiang, and C.-W. Ngo, "Concept-driven multi-modality fusion for video search," IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 1, pp.62–73, Jan. 2011.
[4] T. Yao, T. Mei and C.-W. Ngo "Co-reranking by mutual reinforcement for image search," in Proc. ACM Int. Conf. Image Video Retr., 2010, pp.34–41.
[5] P. Wilkins and A. F. Smeaton "Weighted data fusion for content-based multimedia information retrieval," in Proc. Internet Multimedia Search Min. Jan. 2010, pp. 1–45.
[6] P. K. Atrey, M. A. Hossain, A. E. Saddik and M.S. Kankanhalli. "Multimodal fusion for multimedia analysis: A survey," Springer Multimedia Syst. J., vol. 16, no. 6, pp. 345–379, 2010.
[7] F. Richter, S. Romberg, E. Hörster, and R. Lienhart, "Multi-modal ranking for image search on community databases," in Proc. ACM SIGMM Int. Workshop Multimedia Inf. Retr., 2010, pp. 63–72.
[8] H.-K.Tan and C.-W. Ngo, "Fusing heterogeneous modalities for video and image re-ranking," in Proc. ACM Int. Conf. Multimedia Retr., 2011, p. 15.
[9] Y. Liu, T. Mei, and X.S. Hua, "Crowd Reranking: Exploring multiple search engines for visual search reranking," in Proc. ACM Special Interest Group Inf. Retr. 2009, pp. 500–507.
[10] M. Wang, L. Yang, and X.S. Hua "MSRA-MM: Bridging research and Tech industrial societies for multimedia information retrieval," Microsoft, Washington, DC, Rep. MSR-TR-2009-30, 2009.
[11] Y. S. Jing and S. Baluja, "Visualrank: Applying page rank to large- scale image search," Trans. Pattern Anal. Mach. Intel., vol. 30, no. 11, pp. 1877–1890,Nov. 2008
[12] L. Kennedy, S. F. Chang, and A. Natsev, "Query-adaptive fusion for multimodal search," Proc. IEEE, vol. 96, no. 4, pp. 567–588, Apr. 2008.
[13] Y. Liu, T. Mei, X. Wu, and X.-S. Hua, "Optimizing video search reranking via minimum incremental information loss," in Proc. ACM Int. Workshop Multimedia Inf. Retr., 2008, pp. 253–259.
[14] D. Zhou, S. A. Orshanskiy, H. Zha, and C. L. Giles, "Co-ranking authors and documents in a heterogeneous network," in Proc. IEEE Int. Conf. Data Mining, Dec. 2007, pp. 739–744.
[15] W. Hsu, L. Kennedy, and S.-F. Chang, "Reranking methods for visual search" IEEE Multimedia, vol. 14, no. 3, pp. 14–22, Jul.–Sep. 2007.
[16] W.Hsu, L. Kennedy, and S.-F. Chang, "Video search reranking through random walk over document-level context graph," in Proc. ACM Int. Conf. Multimedia, 2007, pp. 971–980.
[17] P. Wilkins, T. Adamek, G. J. F. Jones, N. E. O'Connor, and A. F. Smeaton "Trecvid 2007 experiments at dublincity university," in Proc.
[18] W. Hsu,L. Kennedy, and S.-F. Chang, "Video search reranking via information ottleneck principle," in Proc. ACM Int. Conf. Multimedia, 2006, pp. 35–44.
[19] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in Proc. ACM SIGMM Int. Workshop Multimedia Inf. Retr., 2006, pp. 321–330.
[20] P. Wilkins, P. Ferguson, and A. F. Smeaton, "Using score distributions for Query time fusion in multimedia retrieval," in Proc. ACM SIGMM Int. Workshop Multimedia Inf. Retr., 2006, pp. 51–60.
[21] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2005, pp. 1816–1823.
[22] K. M. Donald and A. F. Smeaton, "A comparison of score, rank and probability based fusion methods for video shot retrieval," in Proc. ACM Int. Conf. Image Video Retr., Jul. 2005, pp. 61–70.
[23] L. Kennedy, A. Natsev, and S. Chang, "Automatic discovery of query class dependent models for multimodal search," in Proc. ACM Int. Conf. Multimedia, 2005, pp. 882–891.
[24] R.Yan, J. Yang, and A. Hauptmann, "Learing query-class dependent weights in automatic video retrieval," in Proc. ACM Int. Conf. Multime- dia, 2004.
[25] R. Yan, A. Hauptmann, and R. Jin, "Multimedia search with pseudo- relevance feedback," in Proc. ACM Int. Conf. Image Video Retr., 2003, pp. 238–247