An Interactive Content Based Image Retrieval Technique and Evaluation of its Performance in High Dimensional and Low Dimensional Space

Tanusree Bhattacharjee tanusree_bhattacharjee85@yahoo.co.in
Computer Science and Engg. Dept
Jadavpur University Kolkata – 700032, India

Biplab Banerjee gettbiplab@gmail.com
Computer Science and Engg. Dept
Jadavpur University Kolkata – 700032, India

Nirmalya Chowdhury nirmalya_chowdhury@yahoo.com
Faculty,Computer Science and Engg. Dept
Jadavpur University Kolkata – 700032, India

Abstract

In this paper we have developed an Interactive Content Based Image Retrieval System which aims at selecting the most informative images with respect to the query image by ranking the retrieved images. The system uses relevance feedback to iteratively train the Histogram Intersection Kernel Based Support Vector Machine Classifier. At the end of the training phase of the classifier, the relevant set of images given by the final iteration of the relevance feedback is collected. In the retrieval phase, a ranking of the images in this relevant set is done on the basis of their Histogram Intersection based similarity measure with query image. We improved the method further by reducing dimensions of the feature vector of the images using Principle Component Analysis along with rejecting the zero components which are caused by sparseness of the pixels in the color bins of the histograms. The experiments have been done on a 6 category database created whose sample images are given in this paper. The dimensionality of the feature vectors of the images was initially 72. After feature reduction process, it becomes 59. The dimensionality reduction makes the system more robust and computationally efficient. The experimental results also agree with this fact.

Keywords: Relevance feedback, Similarity measures, Content Based Image Retrieval.

1. INTRODUCTION

With increase in size of digital image collections, there is a need of developing an efficient search engine for the images for browsing, searching and retrieval.

There are two types of image retrieval technique. The first one is the text based image retrieval [15],[16] where each image in the database is labeled by keywords describing the image. For searching for an image, user formulates the query using keywords which best describes his query image. In [17] a comprehensive survey is provided in this regard.

Since image databases are very large, so annotating such a huge collection is a very tedious task. Also subjective description of a query image may differ from person to person due to differences in human perception.

The second type of image retrieval technique is the content based image retrieval which has become an attractive research area in the last few decades [12],[13],[14] since it do not possess the difficulties of text based approach. The word content here means low level feature descriptors like color ([18],[19]), texture([20],[21]), shape([22],[23]) etc. The CBIR is the technique to map each of the images in the database to a feature space and then retrieve based on the feature of the query image. The main challenge of CBIR is to bridge the semantic gap between low level content descriptors with high level concepts (like faces, flowers, architectures etc).

For this reason relevance feedback has become a critical component of CBIR systems. Through its use the CBIR system interacts with user and tries to bridge the semantic gap. Relevance feedback has shown considerable improvement in the performance of the CBIR system ([24],[25]).

CBIR can be thought of as a classification task. It classifies the images in the image database into two sets: relevant and irrelevant. The relevant set contains all those images which are relevant to the user's query image and the irrelevant set contains the rest of the images in the database. This relevance with user's query image can be found out using relevance feedback given by user.

SVM_{Active} implemented by Tong et.al.[1] uses the relevance feedback component with the traditional SVM classifier for CBIR. It has given better results than existent Query Point Movement (QPM) and Query Expansion (QEX) scheme. But SVM_{Active} has been implemented with traditional kernels like RBF or polynomial. Recently S. Maji et.al. [5] has shown that classification using Intersection kernel based SVM (IKSVM) not only shows improvement in performance but can also be made computationally efficient by using an approximation technique.

In this work we have designed an interactive Content Based Image Retrieval Technique which uses Relevance Feedback. The Classifier used for this purpose is the Intersection Kernel based SVM which is integrated with a ranking based retrieval using Histogram Intersection based similarity measure.

The color feature subspace computed by dividing into bins resulted into non uniform distribution of the pixels. The pixel concentration is fully dependent on the particular image being considered. Hence due to such sparseness of pixel values there is a necessity to reduce the dimensionality of the feature space.

In this context of dimensionality reduction Sirovich et. al.[28] directly used eigen images. Faloutsos and Lin[29], Chandrasekaren *et al.* [30] and Brunelli and Mich [26] used principal component analysis (PCA) to perform the dimension reduction in feature spaces.

One problem with PCA is the unnecessary information loss which may prove to be disastrous. Unnecessary information loss due to feature reduction do not occur for our case as it is found that over 10 dimensions in the feature space became zero after reduction. So we are only eliminating the zero dimensions and considering all the rest of the dimensions.

The paper is organized in the following way. Section 2 contains an overview of Principle Component Analysis. Support Vector Machine and Histogram intersection kernel are discussed in chapter 3 and 4 respectively. Section 4 has the description of histogram intersection based similarity measure. Section 5 contains the feature extraction module, and finally our proposed algorithm is discussed in Section 6. Experimental results are given in section 7. The paper finishes with conclusion and discussions in Section 8.

2. PRINCIPAL COMPONENT ANALYSIS

Our task of content based image retrieval typically focuses on representing the images in the training database and the test database in a relatively high dimensional feature vector. We can apply some dimensionality reduction scheme to eliminate those dimensions that have low impact on the classification process, we are discussing here a method of dimensionality reduction using PCA[11].

PCA aims at highlighting the variability of data. Let us consider we have s_n number of data points in the two dimension space. If they are represented as (x_i, y_i) $\{i = 1,2,\ldots\ldots s_n\}$. The mean of the data points are calculated by the equation,

$$x' = \sum_{i=1}^{n} x_i / s_n \tag{1}$$

$$y' = \sum_{i=1}^{n} y_i / s_n \tag{2}$$

These means are subtracted from the data points for the purpose of normalization,

$$\textit{Data Adjusted} = \begin{bmatrix} (x_1 - x') & -- & (y_1 - y') \\ | & & | \\ (x_{s_n} - x') & -- & (y_{s_n} - y') \end{bmatrix} \tag{3}$$

We can compute the relative variability of the dimensions in terms of a co variance matrix,

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix} \tag{4}$$

where the principle diagonal indicates the variances of the dimensions *x* and *y* respectively. The Covariance of *x, y* can be computed as follows,

$$\operatorname{cov}(x,y) = \sum_{i=1}^{n} \{(x_i - x'),(y_i - y')\} / n-1 \tag{5}$$

From the covariance matrix the eigen values are calculated. There will be two eigen values for 2 dimensional data points stated above. From these eigen values eigen vectors are calculated. The two eigen vectors corresponding to the two eigen values will be orthogonal to each other. From the point of view of PCA, the eigen vectors should be of unit length. The eigen vector with the highest eigen value is called the principal component of the data set as it contributes maximum information about the patterns in data. The pattern information given by the other eigen vectors decreases with the eigen values. Hence a vector of eigen vector is formed where the eigen vectors are arranged in decreasing order of corresponding eigen value.

$$Feature\ Vector\ =\ (eig\_vec1\ eig\_vec2) \tag{6}$$

The lower order eigen vectors can be ignored as they contribute very little to the pattern information of the data set. The final data is got by the following equation,

$$Final\ Data\ =\ Data\ Adjusted^{T}\ x\ Feature\ Vector^{T} \tag{7}$$

The original data set can be got back from this final data using reverse mathematical operations. PCA has been used successfully in many application areas like face recognition, and dimensionality reduction in many data mining applications

3. SUPPORT VECTOR MACHINE

Support Vector Machine, an important machine learning technique has been used efficiently for variety of classification purposes like object based image analysis[2], hand written digit recognition[3], image segmentation[cite reference] among others. SVM can be used efficiently as a binary classifier as well as multi class classifier [27].

3.1 Theoretical background of SVM

We have a set of data points (images)$\{x_i\}$ where i =1…..l (where l is the no. of training points) and each x_i is drawn from a d dimensional feature space (in our case d = 72). Our aim is to draw a hyperplane using SVM classifier [4] that will separate those training points into two separate classes i.e. +1 for positive classes and -1 for negative classes. The classes are denoted as y_i $\left[y_i \in \{-1,+1\} \right]$.

Linear SVM is the simplest type of SVM classifier which separates the points into two classes using a linear hyperplane that will maximize the margin between the positive and negative set. In our case as we are using SVM with relevance feedback, initially we are not labeling all the training points. We are labeling them iteratively by showing the user some images from the database, which the user marks as relevant (positive) or irrelevant (negative) classes.

The concept of non linearity comes when the data points can not be classified into two different classes using a simple hyperplane, rather a nonlinear curve is required. In this case, the data points are mapped non-linearly to a higher dimensional space so that they become linearly separable. For this purpose the kernel functions are used. Different types of kernel functions are already invented. A choice of kernel function for classification using non linear SVM depends upon the problem domain. Equations for some of them are stated below,

Polynomial :

$$K(x,z) = (w.x + B)^{p},\ p = \deg ree \tag{8}$$

Radial Basis:

$$K(x,z) = e^{-\Upsilon \| x-z \|^{2}} \tag{9}$$

In both the equations, x and z are the data points. *K(x,z)* gives the kernel function for computation of non linear SVM. The main drawback of SVM is its time complexity which can be reduced reasonably as discusses in the next section.

4. INTERSECTION KERNEL BASED SVM

The histogram Intersection Kernel $K(H,H^{'})=\sum_{i=1}^{L}\min(H_i,H_i^{'})$ [6] is used as measure of similarity between histograms *H* and *H'* with *L* number of bins. The Histogram Intersection Kernel based Support Vector Machine(IKSVM) has a complexity of *O(mn)* where n is the dimension of the data points in the data set and m is the number of support vectors of the learned classifier. IKSVM algorithm proposed by S. Maji et. al [5] shows that time complexity is reduced to *O(nlogm)* and space complexity is *O(nm).*Finally they are successful in formulating an approximate method which has time and space complexity devoid of the number of support vectors that is *O(n).*Hence they proved that classification using IKSVM is efficient in terms of resources required.

This result is particularly important because intersection kernel SVMs have been shown to be successful in detection and recognition (pyramid match kernel [7] and spatial pyramid matching [8]), but this is computationally inefficient compared to linear SVMs because non linear kernels require computation and space linearly proportional with number of support vectors (for classification) which also increases with increase in training data [9].

5. HISTOGRAM INTERSECTION BASED SIMILARITY MEASURE

It is shown in [10] Euclidean based similarity measure may not be able to capture user's query concept or user's perception. Inspired by the author [10] we have not used Euclidean based similarity measure.

If two histograms *H* and *H'* has *L* number of bins, then the histogram intersection distance between them is given by,

$$d(H,H^{'})=\frac{\sum_{i=1}^{L}\min(H_i,H_i^{'})}{\min\left(\sum_{i=1}^{L}H_i,\sum_{i=1}^{L}H_i^{'}\right)} \tag{10}$$

6. FEATURE EXTRACTION

We calculate both Color and Texture features for each image in the database. The shape feature is calculated as a property of color and texture features. Number of components calculated for feature vector for each image is 72. Each image represents a point in 72 dimensional feature spaces.

Characteristic Values	R	G	B
1	0	0	0
2	0	0	255
3	0	255	0
4	0	255	255
5	255	0	0
6	255	0	255
7	255	255	0
8	255	255	255

TABLE 1: Characteristic RGB Values.

6.1 Color Feature

We have used histogram based feature extraction by binning the pixel values based on RGB color space. We have defined 7 bins for which the ranges are defined in Table 1. Each of the bin is corresponds to the pixel values between the Range *x* and Range *x+1* where *x*=1,2,…7.

The number of pixels in each of the bin varies from image to image. Many of the bins contain very little number of pixels in comparison to other bins.

Since RGB color space does not provide any information of image brightness and how each pixel is saturated with white color, RGB space is not an effective way of computing feature space. Hence we convert the pixels in each bin from RGB to HSV space where H corresponds to hue or the brightness, S corresponds to the amount by which each pixel is saturated with white color and V corresponds to the dominant color. In each bin, for each of H, S and V channel statistical moments like mean and variance are calculated. Hence they provide 7(number of bins) x3(channels per bin) x2(number of features per channel per bin) = 42 features.

Using Gaussian distribution of average mean (average over three channels) and average variance, spreadness of bin and elongation of bin are calculated. These provides an additional 7(number of bins)x2(number of features per bin) = 14 features.

6.2 Texture Feature

Texture is the tactile characteristic of a surface. For texture feature extraction purpose, we first obtain the discrete wavelet transformed version of a given image using haar wavelet mask. Wavelets are functions generated from a single function by its dilations and translations. The haar wavelet transform forms the simplest and oldest compression of this kind. The haar wavelet transform sub divides the image into 4 parts which denotes the image orientations in 4 different angular values, 0,45,90,180; the process corresponds to the following filtering operations,

- *Top left:* passing the image through 2-D lowpass filter (Lo-Lo).
- *Top right:* passing the image through horizontal highpass and vertical lowpass filter (Hi-Lo).
- *Lower left:* passing the image through horizontal lowpass and vertical highpass filter (Lo-Hi).
- *Lower right:* passing the image through 2-D highpass filter (Hi-Hi).

From an image of resolution *M* x *N*, the haar wavelet transform obtains sub images in four different orientation each of resolution *M*/2 x *N*/2. From these 4 sub images, we calculate the grey level co-occurrence matrix (GLCM). GLCM represents the occurrences of a given pair of intensity values of an image in a certain direction and hence helps in detecting the repeating patterns in the image. We calculate the energy co-efficient from each of the GLCM and sum them up. The inherent idea is that, the energy measure of a complex signal can be obtained by summing up the energy measures of its constituent signals. The energy measure obtained can be used as an useful image feature because it highlights on the local (pixel wise) as well as global (images in different orientations) descriptions of the image.

The energy measure for the 4 GLCMs are calculated as follows

$$Energy = \sum_i \sum_j P_d^2(i,j)$$

$$Energy = \sum_i \sum_j P_d^2(i,j)$$

(11)

where (*i,j*) represents the (row, col) pair of each GLCM represented by P_d where *d* represents displacement of the GLCM.

From each GLCM we obtain energy mean, energy variance, energy spreadness and energy elongation. The energy spreadness and energy elongation are calculated from Gaussian distribution of energy mean and energy variance. These provides another additional 4(number of GLCM) x4(number of features per GLCM) = 16 features

7. PROPOSED METHOD

The proposed method shows some relevant images to the user based on the query image the user selects. This is done by interactively asking the user to label a few images from training set into relevant and irrelevant category. Then using those labeling the histogram intersection kernel based SVM is trained. These feedback rounds go on iteratively and each time a refinement of the images shown to the user is done. Finally those images which are judged by the SVM as positive are taken. The query image is plotted in the positive set. Now those images which are nearest to the query image based on a histogram intersection based similarity ranking are taken and shown to the user.

The performance of the proposed method is checked in two stages. The first and second stage is without and with dimensionality reduction respectively. The above paragraph described the first stage briefly. The second stage blindly follows the first stage with only one exception. After each feature extraction phase, there is a following dimensionality reduction phase with the PCA in the second stage. We are showing the second phase in the form of algorithm as the first phase could be easily derived from it by eliminating the dimensionality reduction phase.

Notations	Descriptions
C	Number of categories of images in the database
p	Number of images in each category
Tr	Training set
r	Number of images of each category in *Tr*
Ts	Test set
k	Number of images displayed to user for feedback
F	*k* images forms set *F*
Rr	Set of images marked as relevant by user
Ri	Set of images marked as irrelevant by user
Fr	Feature vector set of *Rr*
Fi	Feature vector set of *Ri*
F_{Tr}	Set of images classified as positive by SVM
F_{Ti}	Set of images classified as negative by SVM
g	Number of images to be displayed according to user requirement

TABLE 2: Notations for Algorithm.

We have collected a database with *C* categories of images, each category having *p* number of images. The training set *Tr* is formed from this parent database by selecting randomly *r* ($r<p$) images for each of the category. Hence *Tr* has total *rxC* images. The rest of the images in the database form the test set *Ts*. Hence *Ts* has total *(p-r)xC* images.

7.1 Training by $IKSVM_{Active}$

Phase 1: Querying:
User selects an image from *Ts*.

Phase 2: Relevance Feedback:
Step 1: A fixed number (*k*) of images are selected randomly without replacements from *Tr* and kept in set *F*.

Step 2: User is shown the set *F* containing those images.
Step 3: User marks them as relevant or irrelevant.
Step 4: Relevant images form set *Rr* and irrelevant images form set *Ri* $\left(Rr \in Tr \quad \& \quad Ri \in Tr \right)$.

Phase 3: Training IKSVM:
Step 1a: The set *Rr* is modified by including the query image selected by user in it.
Step 1b: Calculate feature vector of *Rr* and *Ri*. Feature vectors of Fr forms the positive set and feature vectors of *Fi* forms the negative set of data points for training the classifier.
Step 1c: The dimensionality of these feature vectors are reduced by calculating the principle components and rejecting all those dimensions which reduces to zero due to sparseness of pixel concentration in the bins of histogram calculated.
Step 2: These sets are then given as input to IKSVM. It draws the hyperplane on the basis of them.
Step 3: The feature set of Tr is calculated and then fed to IKSVM for classification so that the hyperplane formed can separate data points in training set as positive or negative. Those data points which falls on the positive side of the hyperplane forms a set F_{Tr} and similarly the negative side forms another set F_{Ti}. Since an image is a data point in a feature space we use the term data point and feature vector interchangeably.

Phase 4: Selecting sample images for next feedback round:
Step 1: *k/2* data points are selected from the set F_{Tr} which are the most nearest to the hyperplane and similarly *k/2* data points are selected from the set F_{Ti}. The images corresponding to these *k* data points are kept in set *F*.
Step 2: If this is not the final iteration then, Iterate from Step 2 of Phase 2 to Step 2 of Phase 4. Else iterate from Step 2 of Phase 2 to Step 3 of Phase 3.

7.2 Retrieval using histogram intersection based similarity measure
Step 1: Collect the set F_{Tr} from the last iteration of training by IKSVM_{Active}.
Step 2: Find from F_{Tr} , the reduced dimensional feature vector of the query image Now plot it in reduced dimensional space feature space.
Step 3: Plot all the data points in F_{Tr} in the same space (reduced dimensional).
Step 4: Now, say user wants *g* number of images from the database which are most relevant to the query image, then on the basis of histogram intersection based similarity measure calculate the distance of query data point with data points in the set $F_{Tr.}$
Step 5: Sort in descending order the distances as two of the most relevant images will have histogram distance close to 1 and the dissimilar ones will have distance close to 0 and then retrieve first g images from that sorted list.
The experimental results are discussed in the next section.

8. EXPERIMENTAL RESULTS
We have done our experiments on a 6 category image database, each category having 120 images. The categories include field, sea, sunset, white rose, red rose and sunflower.

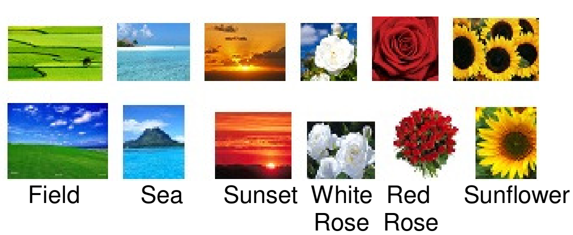


FIGURE 1: Some Sample Images from the Database Used for Experimentation.

The categories have ambiguous RGB values. In sunset category the images have mainly red, yellow and orange colors. In sunflower and red rose category, the images have mainly yellow and red color values. Similarly in white rose category the images have mainly white and green colors. In sea and field category white and green are predominant. The database (http://www.flickr.com/photos/48753989@N03/) is created in this way by collecting images from Google search engine, mainly to confuse the classifier and check how it behaves in this environment. Two images from each category of the database created is shown in Fig 1.

The training set is computed in such a way that it contains 80 images from each category. The test set contains remaining 40 images from each category. One of the performance measures for image retrieval is precision which is defined below,

Precision = (Number of relevant images retrieved)/(No of images retrieved)

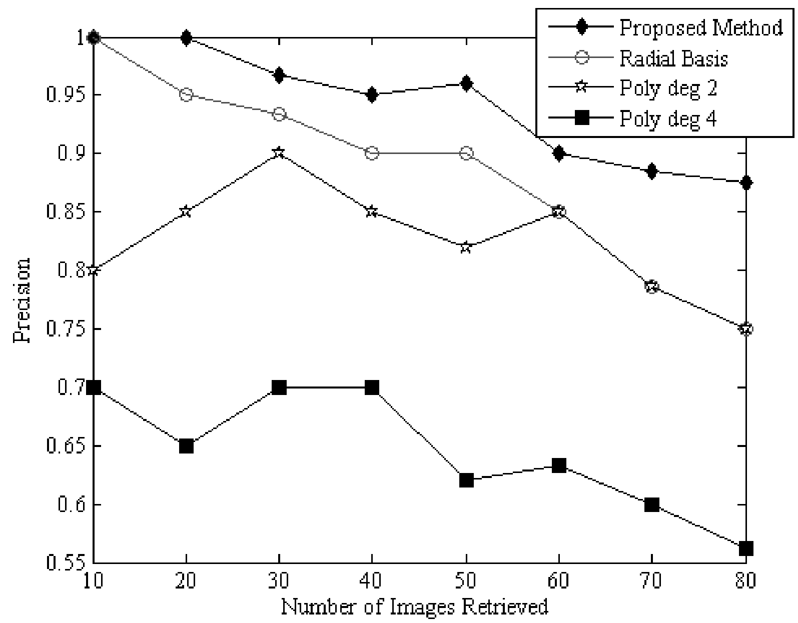


FIGURE 2: Comparative study of Existing Kernel Functions Used With SVM_{Active} and the Proposed Method After Three Feedback Rounds Before Dimensionality Reduction.

The graph of number of images retrieved vs. precision of the system in Stage 1 is shown in fig 2. It shows that the proposed method performs better than other 3 methods for instance when 20 images are retrieved the proposed method has 100% precision whereas SVM_{Active} used with Radial Basis kernel gives 95% precision, with polynomial degree 2 kernel gives 85% precision and with polynomial degree 4 kernel gives 65% precision.

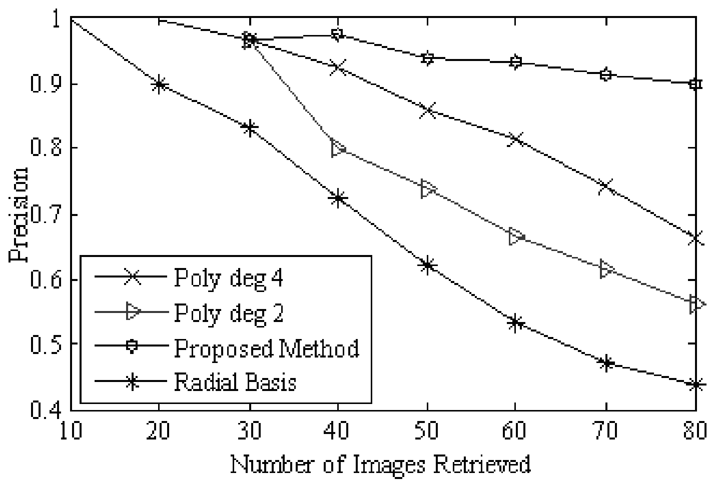


FIGURE 3: Comparative Study of Existing Kernel Functions Used With SVM_{Active} and the Proposed Method After Three Feedback Rounds After Dimensionality Reduction With PCA.

The graph of number of images retrieved vs. precision of the system in Stage 2 is shown in fig 4. The graph shows that the proposed method performs better in reduced dimensionality condition, than other 3 methods for instance when 40 images are retrieved the proposed method has 98% precision whereas SVM_{Active} used with polynomial (degree 4 kernel) gives 92% precision, with polynomial degree 2 kernel gives 80% precision and with polynomial degree 4 kernel gives 72% precision.

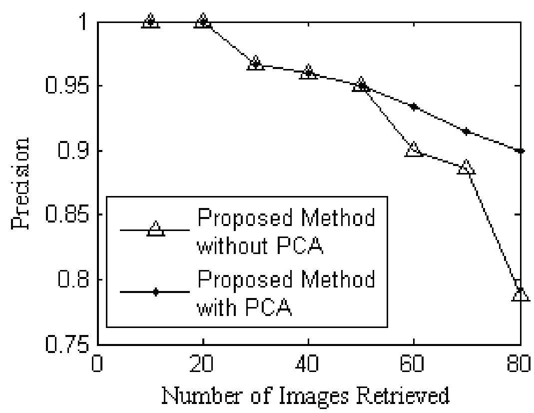
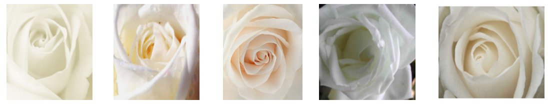


FIGURE 4: Proposed Method With and Without Dimensionality Reduction by PCA.

Query Image Q1

First 5 Results with Proposed Method Before Dimensionality Reduction



First 5 Results With Proposed Method After Dimensionality Reduction

FIGURE 5: Comparative Study of Proposed Method Before and After Dimensionality Reduction by PCA.

The number of images retrieved vs. precision graph in fig 4 shows a comparison of the performance of the proposed method in Stage 1 and Stage 2.

Query Image Q2

First 5 Results With Proposed Method Before Dimensionality Reduction

First 5 Results With Proposed Method After Dimensionality Reduction

FIGURE 6: Another Comparative Study of Proposed Method Before and After Dimensionality Reduction by PCA.

The method without using PCA shows a sharp fall in performance whereas the one with using PCA gradually degrades its performance. Fig 5 and 6 also shows the comparison in the form of simulation results. In fig 5 the query image is a close up view of the white rose. Here dimensionality reduction phase returns those images which have more such type of images as the query image. Similarly in fig 6 the breadth of the field and sky in the query image is preserved more in the returned images after dimensionality reduction.

9. CONCLUSION AND DISCUSSIONS

The proposed method is efficient because unnecessary zero dimensions are deleted. Hence number of computations lessens for drawing the hyperplane using SVM as well as for nearest neighbor computation.

For large databases, the complexity can be reduced by sampling the database as we have done. Further sub sampling may further reduce the complexity. A future direction of research may be to use large databases and efficiently sub sampling them for using the proposed method and the results could be verified.

10. REFERENCES

1. S. Tong and E. Chang. "*Support vector machine active learning for image retrieval*". In Proceedings of ACM Multimedia. Ottawa, Canada, 2001
2. A. Tzotsos and D. Argialas. "*A support vector machine approach for object based image analysis*". Support Vector Machine Classification for Object-Based Image Analysis, Springer-Verlag, Chapter 7.2, 2008
3. T. Joachims. "*Text categorization with support vector machines*". In Proceedings of the European Conference on Machine Learning. Springer - Verlag, 1998
4. C. Burges. "*A tutorial on support vector machines for pattern recognition*". Data Mining and knowledge Discovery, 2:121-167, 1998
5. S. Maji, A.C. Berg and J. Malik "*Classification using intersection kernel support vector machine is efficient*". Appeared in IEEE Computer Vision and Pattern Recognition Anchorage, 2008
6. M. J. Swain and D. H. Ballard. "*Color indexing*". IJCV, 7(1):11–32, 1991
7. K. Grauman and T. Darrell. "*The pyramid match kernel: discriminative classification with sets of image features*". ICCV, 2: 2005
8. L. Lazebnik, C. Schmid, and J. Ponce. "*Beyond bags of features: spatial pyramid matching for recognizing natural scene categorie*s". In Proceedings of CVPR, 2006
9. I. Steinwart. "*Sparseness of support vector machines-some asymptotically sharp bounds*". In Proceedings of NIPS, 2003
10. Y. Rui, T.S. Huang, S.F. Chang. "*Image Retrieval: Past, Present, and Future*". Journal of Visual Communication and Image Representation, 10:1-23, 1999
11. L. I. Smith. "*A tutorial on Principal Component Analysis*".
12. J. Bi, Y. Chen and J.Z. Wang. "*A sparse support vector machine approach to region-based image categorization*". In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '04), 2004
13. I.J. Cox, T.P. Minka, T.V. Papathomas, P.N. Yianilos. "*The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments*". IEEE Transaction on Image Processing, 9:20-37, 2000
14. K.-S. Goh, E.Y. Chang, and W.-C. Lai. "*Multimodal concept-dependent active learning for image retrieval*". In Proceedings of 12th Annual ACM Internationall Conference of Multimedia, 2004
15. N.S. Chang and K.S. Fu. "*A relational database system for images*". Technical Report. TR-EE. 79-28, 1979
16. N.S. Chang and K.S. Fu "*Query-by pictorial- example*". IEEE Transaction On Software Engineering, 6(6):1980
17. S. K. Chang and A. Hsu. "*Image information systems: where do we go from here?*" IEEE Transaction on Knowledge and Data Engineering, 4(5):1992
18. J. Huang, R. Kumar, M. Mitra, W. J. Zhu and R. Zabih. "*Image Indexing Using Color Correlograms*". In Proceedings of CVPR, 1997
19. N. Sebe and M. S. Lew. "*Robust color indexing*". In Proceedings of the 7th ACM International conference on Multimedia, 1999
20. B.S. Manjunath, W.Y. Ma. "*Texture features for browsing and retrieval of image data*". IEEE Transaction on Pattern Anaysis and Machine Intelligence, 18: 837-842,1996
21. J.Z. Wang, G. Wiederhold, O. Firschein, X.W. Sha. "*Content-based image indexing and searching using daubechies' wavelets*". International Journal of Digital Libraries, 1(4):311-328, 1998
22. C. Schmi, R. Mohr. "*Local grayvalue invariants for image retrieval*". IEEE Transaction on Pattern Analysis and Machine Intelligence. 19: 530-535, 1997
23. X. S. Zhou, Y. Rui and T. Huang. "*Water-Filling: a novel way for image structural feature extraction*". In Proceedings of IEEE International Conference on Image Processing, 2: 570-574, 1999
24. P. Hong, Q. Tian and T.S. Huang. "*Incorporate support vector machines to content-based image retrieval with relevance feedback*". In Proceedings of IEEE International Conference on Image Processing, Vancouver, Canada, 2000

25. S. D. MacArthur, C. E. Brodley and C. Shyu. "*Relevance feedback decision trees in content-based image retrieval*". In Proceedings of IEEE Workshop CBAIVL, South Carolina, 2000
26. B. Roberto and M. Ornella. "*Image retrieval by examples*". IEEE Transaction on Multimedia, 2: 2000
27. K. B. Duan and S. S. Keerthi. "*Which is the best multiclass svm method? an empirical study*". Springer-Verlag, 2005
28. M. Kirby and L. Sirovich. "*Application of the Karhunen-Loeve procedurefor the characterization of human faces*". IEEE Transaction on Pattern Analysis and Machine Intelligence. 12:103–108, 1990
29. C. Faloutsos and K. Lin. "*Fastmap: A fast algorithm for indexing,data-mining and visualization of traditional and multimedia*". In Proceedings of SIGMOD, 1995
30. S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler and H. Zhang. "An eigenspace update algorithm for imageanalysis". CVGIP: Graph. Models Image Process. J., 1997