International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 05 | Aug-2015 www.irjet.net p-ISSN: 2395-0072

SEARCHING IMAGE WITH HASH CODE GENERATIONS

[1]R.Lawanya,*[2]Mrs.G.Sangeetha Lakshmi, [3]Ms.A.Sivasankari

*[1,*2,3]Department of Computer Science,DKM College for Women, Vellore, Tamil Nadu, India.*

Abstract: ***The scalable search image depends on the visual similarity has been research on hash code generation with the image. The State- of -the –art solution often use hashing techniques to embed high-dimensional image features into hamming space, the searching process can be performed into real-time depends on the hamming distance of the hash codes. The hamming distance is a large no. of images to providing the equal hamming space to a query, the largely hurts where fine grained ranking is important to the searching process.***

The query-adaptive image with an equal distance to the queries. The weight learning process as a quadratic programming problem that minimizing the intra –class distance while inter-class relationship to capture by the original raw image features. The query-adaptive bitwise weights returned images can be easy to arrange by weighted hamming distance at a finer grained hash code level rather than the original hamming space level. These techniques should be used to clear the picture quality to display a collection of dataset in the proposed approach.

KEY WORDS:*Weight hamming distance, Fine-grained ranking, Hash code generation, scalability, Query-adaptive image search.*

1. INTRODUCTION

Image processing is a method to convert an image into digital form and perform some operation to get an enhanced image or to extract some useful information's from it.

It's a type of signal dispensation in which input is image like video frame or photograph and characteristics associated with the images. It's includes treating image as two dimensional signals while applying already set signal processing in the concepts.

The application in different aspect a management, image processing forms core research area within engineering and computer science disciplines too. The local invariant pictures description are extracted and quantized depend on collections of data's.

Searching images basically includes in the steps:

- The important of the digital photography.
- It's manipulating the pictures which include a data compression and image enhancement that are not to humanface like satellite photographs.

- The result can be altered image or report that is depends on picture analysis.

1.1 IMAGE SEARCHING RESULT

The query-adaptive bitwise weights need to be computed in real-time process in the final stage.A collection of semantic concepts classes that cover the entire different semantic aspects of image content.e.g., scenes and objects.

Bitwise weights for each of the semantic classes are learned offline using a novel formulation that not only maximizes intra-class sample similarities but also preserves inter-class relationship with optimal weights can be computed by iteratively to solving the quadratic programming problem in the searching process.

These pre-computed by the class-specific bitwise weights are then utilizing for computations of the query-adaptive weights, to directly evaluating the proximity of a query images to the picture samples of the semantic classes.

The weighted Hamming space to apply that evaluates similarities between the queries and images in a target database. We name this weighted distance as query-adaptive Hamming distance, as opposed to the query-

independent Hamming spaces widely used in existing works.

The online searching it's unnecessary to compute the weighted Hamming distance based on real-valued vectors; it's one of the most important advantages of hashing. Instead the weights can be utilized as indicators to efficiently order the returned images (found by logical XOR operations) at hash code lev

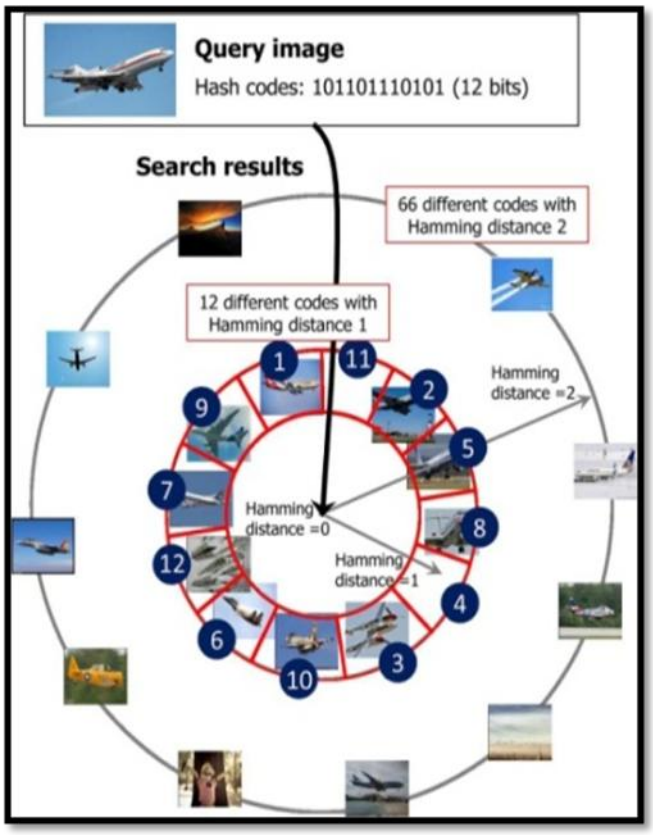

Fig1. Image searching result

1.2PURPOSE OF THE IMAGE PROCESSING

Visualization- Observe the objects that are not visible.

- Image sharpening and restoration- To create a better image.
- Image retrieval- Seek for the image of interest.
- Measurement of pattern- Measures various objects in an image.
- Image Recognition- Distinguish the objects in an image.

1.3TYPES OF IMAGE PROCESSING

- Analog Image processing.
- Digital Image processing.

1.3.1 ANALOG IMAGE PROCESSING

Analog or visual techniques of image processing can be used for the hard copies like printouts of the images.The Image analysis use in different fundamental of interpretation.

The image processing is related to area that has to be studied but on knowledge of analyzing. Association is another important tool in image processing through visual techniques. It's applying a combination of personal knowledge and collateral data to image processing.

1.3.2 DIGITAL IMAGE PROCESSING

Digital image processing techniques help in manipulation of the digital images by using this purpose. The raw data from image sensors from satellite platform contains deficiency. To get the originalities of information's, in difference phases are using in this process.

Then three general phases that all types of data have to undergo while enhancement and provided images finally information's are extract in the processing.

1.4 WORKING DIAGRAM OF IMAGE PROCESSING

Before going to processing an image, it's converting to a digital format. Digitization it includes the sampling of image and quantization of the sample values. After converting the image into bit information, processing is performed.

The processing techniques may be Image enhancement, image restoration, and Image compression.

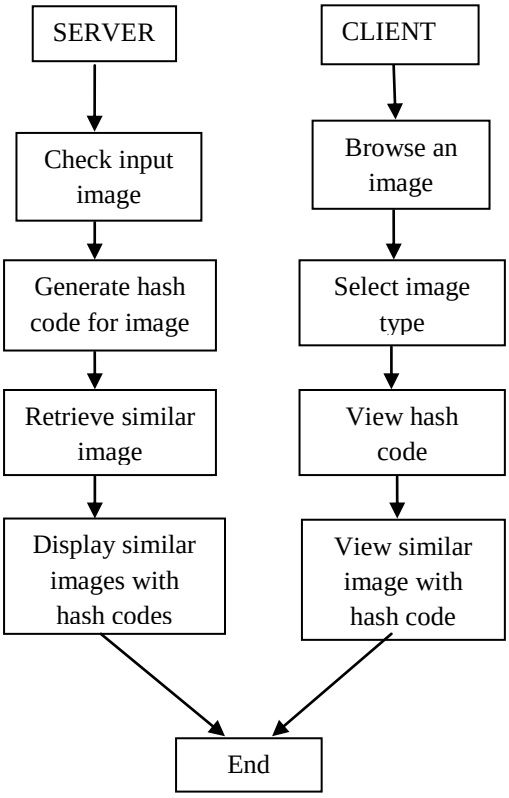

Fig2. Flow chart with image processing

1.5 PROCESSING FOR SEARCHING IMAGE WITH HASH CODE GENERATION

Given a query image, we first extract bag-of-visual-words feature and embed it into a short hash code. The hash code is then used to predict query-adaptive bitwise weight by a natural source to collect the semantic concept classes with pre-computing the quantity of class-specific bitwise weights. Finally, the query-adaptive weights are applied to rank search results using weighted (query-adaptive) Hamming distance.

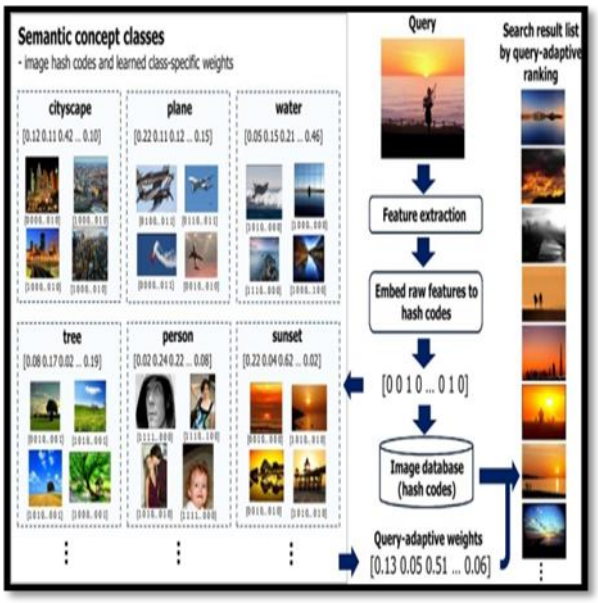

Fig3. System architecture

2. AN OVERVIEW TO SEARCHING IMAGE

We now introduce the hash code generations to overview of the searching image architecture and divide the entire protocol descriptions.

- Query image search.
- Feature extraction.
- Embed raw features to hash code.
- Search result with query adaptive ranking.

2.1QUERY BASED IMAGE SEARCH

The person to search the specific mage query search and they are provided by the original or relatedimages,

2.2EMBED RAW FEATURES TO HASH CODE

We generate and embed image with hash codes. if two object are equal according to the equals (object) method, then calling the hash code method on each of the two objects must produce the same integer result.

These features are generates in the two different of hash codes.

- General hash code (the general hash codes for a particular related image groups)
- Class specific hash code (it's specific images are search to get the collection of relevant image group)

A query represented by a 10-bit hash code. Traditionally hashing-based search result are ordered by integer value for hamming spaces, which is not ideal since many different hash codes share the same distance to the query.

For instance there are 10 hash codes having hamming distance 1 (each differs from the query in one bit).

2.3SEARCHING RESULT WITH QUERY ADAPTIVE RANKING

We extend the framework for query-adaptive hash code selection. The semantic concept classes are used to infer query semantics, the selection of a good set of hash codes for the query, and the computation of corresponding query-adaptive weights on the chosen hash code.

The selected class-specific codes are used together with the general codes for image search. Query adaptive weights will be based on both general and class specific codes.

The hash code is then used to predict query-adaptive bitwise weights (we generate query-adaptive bitwise weights using algorithm) by harnessing a set of semantic concept classes with pre-computed class-specific bitwise weights. Finally, the query –adaptive weights are applied to rank search results using weighted (query-adaptive) hamming distances.

3. HASHING

This work two state-of-the-art hashing techniques are adopted, semi-supervised hashing and semantic hashingwith deep belief networks.

3.1 SEMI-SUPERVISED HASHING

The Semi-Supervised Hashing (SSH) is a newly proposed algorithm that leverages semantic similarities among labeled datawhile remains robust to over fitting.

The objective function of SSH consists of two major components, supervised empiricalfitness and unsupervised information theoretic regularization. More specifically, on one hand, the supervised part triesto minimize an empirical error on a small amount of labeleddata.

The unsupervised term, on the other hand, provides effectiveregularization by maximizing desirable properties likevariance and independence of individual bits.

Mathematically, one is given a set ofn points, r={V_i }, i=1…..n, $V_i \in R^D$, in which a small fraction of pairs are associated with two categories of label information, M and C. Specifically, a pair (V_i ,V_j) ∈ M is denoted as a neighbor-pair when (V_i ,V_j) share common class labels. Similarly, (V_i ,V_j) ∈ C is called a non neighbor- pair if the two samples have no common class label. The goal of SSH is to learn hash functions (H) that maximize the following objective function.

$$J(H) = \sum_{k=1}^{d} \{ \sum_{(vi,vj)\in M} h_k(v_i) h_k(v_j) - \sum_{(vi,vj)\in C} h_k(v_i) h_k(v_j) \} + \sum_{k=1}^{d} var[h_k(v)]$$

The first term measures the empirical accuracy over the labeled sample pair sets and the second part, i.e., the summation of the variance of hash bits, realizes the maximum entropy principle.

This optimization problem is nontrivial.However, after relaxation, the optimal solution can beapproximated using Eigen-decomposition. Furthermore, an algorithm called semi-supervised sequential projection learningbased hashing (S3PLH) was designed to implicitly learn bit-dependenthash codes with the capability of progressively correctingerrors made by previous hash bits [10], where in each
Iteration.

The weighted pairwise label information is updated byimposing higher weights on point pairs violated by the previoushash function.

In this work, the (S3PLH) algorithm is applied to generate hash codes .The labeled samples for learning hash functions. Both training (hash function learning) and testing of SSH are very efficient.

3.2 SEMANTIC HASHING WITH DEEP BELIEF NETWORKS

Learning with deep belief networks (DBN) was initially proposed for dimensionality reduction. It was recently adoptedfor semantic hashing in large-scale search applications. Like SSH, to produce good hash codes DBN also requiresimage labels during training phase, such that images with thesame label are more likely to be hashed into the same bucket.

Since the DBN structure gradually reduces the number of unitsin each layer[3], the high-dimensional input of original image featurescan be projected into a compact Hamming space.

Broadly speaking, a general DBN is a directed acyclic graph,where each node represents a stochastic variable. There aretwo critical steps in using DBN for hash code generation, thelearning of interactions between variables and the inference ofIn practice, the number of units may increase or remain stable for a fewlayers, and then decrease.Observations from inputs.

The learning of a DBN with multiplelayers is very hard since it usually requires estimating millionsof parameters.

The training process can be much more efficientif a DBN is specifically structured based on the RBMs. Eachsingle RBM has two layers containing respectively outputvisible units and hidden units, and multiple RBMs can bestackedto form a deepbelief net.

Starting from the input layerwith dimension, the network can be specifically designed toreduce the number of units, and finally output compact –dimensionalhash codes.

To obtain optimal weights in the entirenetwork, the training process of a DBN has two critical stages:unsupervised pre-training and supervised fine-tuning. Thegreedy pre-training phase is progressively executed layer bylayer from input to output, aiming to place the network weights(and the biases) to suitable neighborhoods in parameter space.

After achieving convergence of the parameters of one layer viaContrastive Divergence, the outputs of this layer are fixed andtreated as inputs to drive the training of the next layer.

Duringthe fine-tuning stage, labeled data is adopted to help refine thenetwork parameters through back-propagation. Specifically,a cost function is defined to ensure that points (hash codes)within a certain neighborhood share the same label.

Thenetwork parameters are then refined to maximize this objectivefunction using conjugate gradient descent.

ALGORITHM: LEARNING CLASS-SPECIFIC BITWISE WEIGHTS

1. INPUT:
Hash code $\boldsymbol{X}$;
Class similarity $\mathbf{S}_{ij}$in original image
Feature space,
i, j=1,……..,k.
2. Compute$\boldsymbol{C}^{(i)}$;
3. Initialize a_j =1/d, j=1,……., k;
4.Repeat
5. For i = 1,……, k.
6.Compute $\boldsymbol{Q}_{ij}$,$\boldsymbol{P}_i$, and$\boldsymbol{t}_i$;
7.Solve the problem in $\boldsymbol{QP}$:

$\mathbf{a}_i$ *= arg min1/2 $\mathbf{a}_i^T\boldsymbol{Q}_i\mathbf{a}_i + \boldsymbol{P}_i^T\mathbf{a}_i + t_i$
S.t $a_i^T\mathbf{1}$=1 and a_i>= $\mathbf{0}$;

8.Set $\mathbf{a}_i = \mathbf{a}_i$*;
9.End for
10.Until convergence
11. OUTPUT:
Class-specific bitwise
weights$\mathbf{a}_j$,j= 1,……., k.

4. RESULTS AND DISCUSSIONS

4.1 CHARACTERISTICS OF HASH CODES BASED SEARCH

Let us first check the number of test images with each Hamming distance value to a query. The 48-bit hash codes fromDBN
are used in this experiment.

We will not specificallyinvestigate the effect of code-length in this paper, since severalprevious works on hashing have already shown that codes of32–50 bits work well in practice. In general, usingmorebits may lead to higher precision, but at the price of low recalland longer search time.

The images at each Hamming distance rapidly grow with the distance values until. This verifies one nature ofHammingdistance, as mentioned in the introduction, there can be differenthash codes sharing the same integer distanceto a query in a -dimensional Hamming space. Consequently,the number of hash codes (and correspondingly the number ofimages) at each specific distance increases dramatically asgrows until.For some queries, there can be as manyas - images sharing equal distances.

This motivates theneed of our proposed approach that provides ranking at a finergranularity. Although our approach does not permute/re-rankimages with Hamming distance 0 to the queries, this analysisreveals that this is not a critical issue since most queries havenone or just a few such images (2.4 on average in this evaluation).

4.2 QUERY-ADAPTIVE RANKING

Next we move on to evaluate how much performance gaincan be achieved from the proposed query-adaptive Hammingdistance, using 32-bit and 48-bit hash codes from both SSH andDBN (the general sets trained with labels from every class).

The approach significantly out per forms traditional Hamming distance.For the DBN codes, it improves the 32-bit baseline by6.2% and the 48-bit baseline by 10.1% over the entire ranklists. A little lower but very consistent improvements (about5%) are obtained with the SSH codes.

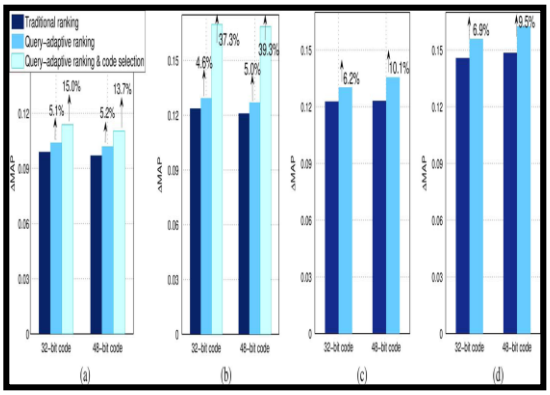

Fig4.Search performance comparison Graph.

The steady improvements clearly validate the usefulness of learning query-adaptive bitwise weights for hash codes based image search. To performance over the upper half part of search results, using the same set of queries. The aim of this evaluation is to verify whether our approach is able to improve the ranking of top images, i.e., those with relatively smaller Hamming distances to the queries.

As expected, we observesimilar performance gain to that over the entire list.Looking at the two hashing methods, DBN codes are betterbecause more labeled training samples are used (50 k for DBNvs. 5 k for SSH). Note that the comparison of DBN and SSH isbeyond the focus of this work, as the latter is a semi-supervisedmethod, which prefers and is more suitable for cases with limitedtraining samples. Direct comparison of the two with equal
training set size can be found in.

To see whether the improvement is consistent over the evaluatedqueries, we group the queries into 81 categories based ontheir associated labels. Results from the 48-bit DBN codes are displayed in the significant performance gain is observedfor almost all the categories, and none of them suffers from performance degradation.

This shows another advantage ofOur approachit offers consistently improved results for mostqueries. Where the query-adaptive ranking approach produces betterresults by replacing less relevant images to the queries withmore suitable ones.

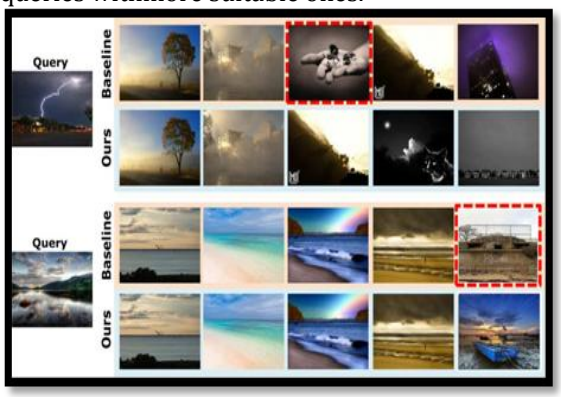

Fig5.Top 5 returned images of two queries, using the 48-bit DBN codes.

We also evaluate the effect of parameter in class-specificweight learning and in query-adaptive weight computation using the DBN codes. Results are visualized in the images.

To performance gain of query-adaptive ranking versus parameters (left)and (right), using DBN codes, that the performance gain increases with first and then decreases,indicating that the inter-class relationship is helpful. For, we observe a fairly stable performance gain when it is set toa value larger than 50.

4.3 QUERY-ADAPTIVE HASH CODE SELECTION

Finally we evaluate query-adaptive hash code selection,following the extended framework described in Section V.C. To shows the overall search performance on thesame set of queries using SSH codes (the light blue bars onthe right).

We see obvious performance improvement fromthis extension, with overall gains 9.4% (32-bit) and 8.1%(48-bit) over the results obtained by query-adaptive rankingwith general hash codes (15.0% and 13.7% respectively overthe baseline traditional ranking).

The upper half of searchresult lists, the improvement over query-adaptive ranking isvery significant: 31.2% and 32.8% for 32-bit and 48-bit codesrespectively. These results confirm the effectiveness of theclass-specific hash codes and our query-adaptive code selection
framework.

In addition, we also find that the query-adaptiveweights imposed on the selected class-specific hash codesdo not contribute as much as that on the general hash codes(around 2%, versus 5–10% on the general codes).

This isprobably because the hash code selection process already takesquery semantics into account by choosing a more suitable setof hash codes for each query. Although the bitwise weights canstill improve result ranking, from query-adaptive perspectivevery limited additional information can be further attained.

While promising, it is worth noting that the query-adaptivehash code selection framework incurs additional computationand memory cost.

First, query images need to be hashed twiceand search needs to be performed with more bits, whichasmentioned in Section V.Care generally acceptable sincehashing algorithms and bitwise operations are efficient. Secondand more importantly, in order not to affect real-time search,we also need to pre-load the class-specific codes of all databasesamples, which would require 81 times of the memory neededby the general codes.

Although this is still much less than thatneeded by original raw features, the requirement is nontrivialwhen very large database is in use. Therefore we conclude thatclass-specific hash codes are useful for improved performance,but this introduces a trade-off between search performanceand memory usage that needs to be carefully considered inreal-world applications, e.g., per application needs and hardwareconfiguration.

5. CONCLUSION

We have presented a novel framework for searching image with hash code to generations with the large no. of collections of predefined semantic concept classes in these approaches to provide the predict

Query-adaptive bitwise weights of hash codes in real-time with which search result can be rapidly ranked by weighted hamming distance at finer grained ranking provided for hash codes.

The capability of a largely unpleasant condition to the effect of a coarse ranking problem that is common in hashing-based image search.

Experimental result on a widely adopted Flickr image data set confirmed the effectiveness of our proposal. It's indicating that the class-specific codes can further improve search performance significantly.

6. REFERENCES

1. A. Torralba, R. Fergus, and W. Freeman, **"80 million tiny images: A large data set for nonparametric object and scene recognition,"**IEEE Trans.Pattern Anal. Mach. Intell.vol. 30 No. 11, pp.1958–1970, Nov.2008.

2.J. Sivic and A. Zisserman, **"Video Google: A text retrieval approach to object matching inVideos,"** in Proc. IEEE Int. Conf.Computer Vision, 2003.

3.D. Lowe, **"Distinctive image features From scale-invariant keypoints,"** Int. J. Compute. Vision, vol. 60, no. 2, pp.91–110, 2004.

4. J. R. Smith and S.-F. Chang, **"Visual Seek:A fully automated contentbased imagequery System,"** in Proc. ACM Int.Conf.Multimedia,1996.

5. A. Olivia and A. Torralba,**"Modeling the shapeOf the scene: A holistic representation of the spatialenvelope"** Int. J.Comput. Vision, vol. 42, pp. 145–175, 2001.

6.D. Nister and H.Stewenius, **"Scalable recognition With a vocabulary tree,"** in Proc. IEEE Conf. Computer Vision and PatternRecognition, 2006.

7. G. Hinton and R. Salakhutdinov, **"Reducing theDimensionality of data with neural networks,**Science,vol.313, no. 5786,pp. 504–507, 2006.

8. R. Datta, D. Joshi, J. Li, and J. Z.Wang, **"ImageRetrieval: Ideas,influences,and trends of the new Age,"** ACM Comput. Surveys, vol. 40, no.2, 2008.

9. J. L. Bentley,**"Multidimensionalbinary searchTrees used forassociative searching,"**Commun. ACM, vol. 18, no. 9, pp. 509–517, 1975.

10. A. Torralba, R. Fergus, and Y.Weiss, **"SmallCodes and largeimagedata base form Recognition,"** inProc. IEEEConf.Computer Vision and Pattern Recognition, 2008.

11. R. A. Fisher,**"The use of multiple measurements in taxonomic with the problems,"**Ann. Eugenics, vol. 7, pp. 179–188 1936.

12. E. Xing,andM. Jordan and S.Russell, **"Distance metric learning with in the application to clustering with side-in formation,"** Adv. Neural Inf Process. Syst., pp. 521–528, 2003.

13.K.Weinberger and L. Saul, **"Fastsol-vers and efficient implementationsfor distance metric learning,"** inProc. Int. Conf. Machine Learning ,2008.

14.T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng**"NUSWIDE: A real-world web image database from National University of Singapore,"** in Proc. ACM Int. Conf. Image and Video Retrieval, 2009.

15.Y. G. Jiang, C. W. Ngo, and J. Yang, **"Towards optimal bag-of-features For object categorization andsem-Antic video retrieval,"** in Proc.ACM Int. Conf. Image and VideoRetrieval, 2007.

16 J. Yang and A. G. Hauptmann, **"Unreliability of video concept detection,"**in Proc. ACM Int. Conf. Image and Video Retrieval, 2008.

7. ABOUT THE AUTHORS

Lawanya. R is a Research Scholar in the Department of Computer Science at DKM College for Women, Vellore pursuing M.Phil in Thiruvalluvar University, Vellore. Her special research interests are in Computer Networks and Database Management Systems. She completed her Masters in Computer Science from DKM College for Women, Vellore.

Prof. SangeethaLakshmi.G is an Assistant Professor in Department of Computer Science at DKM College for Women.Her areas of interest are Microprocessor and Computer Architecture.

Prof. Sivasankari.A is the Head of the Department at DKM College of Women. She has a deep knowledge in the field of computers. Her interests are in DBMS, Network Security, Multimedia, Java, VC++ and Microprocessor.