Multiple Kernel Learning for Image Indexing

Ehtesham Hassan*
Department of Electrical Engg.
IIT Delhi, India
hassan.ehtesham@gmail.com

Santanu Chaudhury
Department of Electrical Engg.
IIT Delhi, India
santanuc@ee.iitd.ac.in

M Gopal
Department of Electrical Engg.
IIT Delhi, India
mgopal@ee.iitd.ac.in

ABSTRACT

The paper presents a novel framework for learning the hash functions for indexing through Multiple Kernel Learning. The Distance Based Hashing function is applied which does the object projection to hash space by preserving inter object distances. In recent works, the kernel matrix has been proved to be more accurate representation of similarity in various recognition problems. Our framework learns the optimal kernel for hashing by parametrized linear combination of base kernels. A novel application of Genetic Algorithm for the optimization of kernel combination parameters is presented. We also define new texture based feature representation for images. Our proposed framework can also be applied for optimal combination of multiple sources for indexing. The evaluation of the proposed framework is presented for CIFAR-10 dataset[1] by applying individual and combination of different features. Additionally, the primary experimental results with MNIST dataset[2] is also presented.

Keywords

Indexing, Multiple kernel learning, Genetic algorithm, Distance based hashing

1. INTRODUCTION

The efficient and accurate indexing of the large amount of digital documents e.g. images, documents, audio and video files existing across the web is a challenging task. In such scenario, the nearest neighbor search for retrieving similar objects does not give a practical solution. The approximate nearest neighbor search give efficient solution for such problems. The LSH is widely accepted as state of the art method for solving the approximate nearest neighbor search problems [1]. The LSH based methods index similar objects to same location where the similarities of objects is computed with respect to predefined distance metric. The effective representation of semantic information inherent in the documents is still an open research problem. However, the category or label information attached with the documents to a great extent represent the semantics in the document which can be utilized to improve the performance of hashing scheme.

In this paper, we define a novel hashing function learning scheme through performing Multiple Kernel Learning. The Distance based hashing (DBH) which is fundamentally based on line projection formula is applied [2]. The DBH projects the object to low dimensional hash space by preserving the inter object distances. Therefore similar objects are hashed to the same or nearby buckets, the similarity is represented by the distance between objects. The kernel methods have given excellent results for many classification problems. The recent development of Multiple Kernel Learning in this direction have the capability to learn the optimal kernel for classification from the data itself, as well as by combining the multiple data sources. The kernel space representation of DBH has inherent capability to perform hashing by applying the given kernel matrix as similarity matrix. However, the availability of optimal kernel can never be guaranteed. We define a hashing function learning framework on the line of existing MKL algorithms. The framework learns the optimal kernel for hashing by combining set of base kernels. We have applied Genetic Algorithm (GA) for MKL which have the advantage of applicability to symbolic objectives and it does not suffer from the problem of local maximas and minimas. In addition, we propose novel texture based image feature which represents the image using *bag of words* model. For various recognition problems combination of multiple data sources have given improved results [3][11]. Our proposed MKL framework learns the optimal kernel for hashing through the set of supplied kernels. The capability can be efficiently utilized to optimally combine different features for indexing.

The DBH is fundamentally an unsupervised hashing method, which performs the object indexing based on object space distance computation. However, the object space distance does not completely represent the semantic distance between objects. Additionally for many computer vision problems kernel distance is preferred as the notion of similarity. With the application of kernel trick, the DBH can be extended for hashing in kernel space. However, in practice the selection of optimal kernel parameters for performing hashing in kernel space is time consuming and difficult process because

*Corresponding author
[1]http://www.cs.toronto.edu/ kriz/cifar.html
[2]http://yann.lecun.com/exdb/mnist/

ICVGIP '10, December 12-15, 2010, Chennai, India

of the complex geometric structure of object space. In such case, defining a MKL framework to learn the optimal kernel for hashing can provide efficient solution to the problem. Our MKL framework for hashing function learning combines a set of base kernels for hashing. The optimization problem is formulated in the evolutionary algorithm framework, which optimizes the kernel combination weights at convergence. The category information available in training set (i.e. labeled data) concludes the similarity and dissimilarity between the objects. We can utilize this labeled data to evaluate the fitness function for GA population evaluation. The limited amount of labeled data may lead to over fitting solution. Thus to increase the generalizing ability of optimization objective a regularizer is required. We use information theory based regularizer proposed in [12] in our objective function. The primary experimental results of the proposed framework using the GIST feature and the proposed features is presented for CIFAR-10 dataset. We also present indexing performance for the dataset by optimal combination of GIST and texture features. The basic experimental results with MNIST dataset is also presented.

The organization of paper is as follows. The section 2 presents brief review of DBH and its extension to kernel space (Kernel DBH). The section 3 presents the Distance based hashing function learning framework through MKL. The discussion includes optimization problem definition and GA based optimization framework for MKL. Section 4 presents the details for Local texture feature computation. The section 5 presents the experimental results and discussion of the proposed framework on two standard recognition datasets. Finally, we conclude and present the perspective of our work.

2. DISTANCE BASED HASHING IN KERNEL SPACE

2.1 Review:The DBH

The DBH performs projection of objects on a carefully selected line such that inter object distances are preserved. The line projection is applied for performing object mapping. For two objects (x_1, x_2) in the space (X, D), the line projection $F^{x_1,x_2} : X \rightarrow \mathcal{L}$ for object x is defined as

$$F^{x_1,x_2}(x) = \frac{D(x_1,x)^2 - D(x_2,x)^2 + D(x_1,x_2)^2}{2D(x_1,x_2)} \quad (1)$$

Here $\mathcal{L}$ represents the line connecting point representation of pivot objects (x_1, x_2). The equation (1) can be used to define a rich family of functions having $N(N-1)/2$ unique functions for each pairs of N objects from the database. In practice, the hash values are discrete in nature, whereas the equation (1) gives real values. The discretization of real hash functions defined by (1) is performed by using threshold parameters $t_l, t_2 \in R$ as

$$F^{x_1,x_2}_{t_1,t_2}(x) = \begin{cases} 1 & \text{if} \quad F^{x_1,x_2}(x) \in [t_1, t_2] \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The selection of (t_1, t_2) should be such that, $F^{x_1,x_2}_{t_1,t_2}(x)$ maps approximately half the data points in X to 0 and half to 1, i.e. F generates balanced hash tables. Therefore the set $V(x_1, x_2)$ of intervals $[t_1, t_2]$ for all pairs $(x_1, x_2) \in X$ is defined as

$$V(x_1,x_2) = [t_1,t_2] | Pr_{x \in X}(F^{x_1,x_2}_{t_1,t_2}(x) = 0) = 0.5 \quad (3)$$

Now, the hash function family H_{DBH} is defined as

$$H_{DBH} = F^{x_1,x_2}_{t_1,t_2}(x) | x_1, x_2 \in X, [t_1,t_2] \in V(x_1,x_2) \quad (4)$$

The H_{DBH} can be generated by selecting N sample objects. The equation (4) can be used to define an indexing scheme by generating L hash tables where each hash table corresponds to a k-bit hash function formed by concatenation of k functions selected randomly from H_{DBH}. The retrieval process includes query hashing on hash tables (mapping of query on each hash table) and performing similarity search over the pool of objects collected from all the query buckets. The hash table parameters (L, k) are adjustable parameters and are defined by performance requirements.

2.2 Kernel based DBH

In the following discussion, the Kernel based DBH is presented. Considering X as Euclidean vector space and D Euclidean distance, the squared distance $D^2(x_1, x_2)$ can be expanded as $x_1^T x_1 + x_2^T x_2 - 2x_1^T x_2$. Equation (1) is redefined as

$$F^{x_1,x_2}(x) = \frac{x_1^T x_1 - x_1^T x + x_2^T x - x_1^T x_2}{\sqrt{x_1^T x_1 - 2x_1^T x_2 + x_2^T x_2}} \quad (5)$$

The above expression represents the line projection computation using dot products. The kernel methods increase the computational power of linear learning algorithms by mapping the data to high dimensional feature space [2]. The mapping $\phi : X \rightarrow S$ i.e. $x \rightarrow \phi(x)$ from input space X to kernel space S, defines dot product $x^T x'$ in the kernel space as $\phi^T(x)\phi(x')$. It is clear that direct mapping to space S can be implicitly performed by selecting a feature space which supports the direct computation of dot product using a non-linear function in input space. The kernel function k which performs such mapping is defined as

$$k(x, x') = \langle \phi(x), \phi(x') \rangle = \phi^T(x)\phi(x')$$

The expression shows the mapping to space S by function k happens implicitly without considering the actual form of ϕ. In this case, kernel space equivalent of the squared distance $D^2(x_1, x_2)$ is defined as $k(x_1, x_1) + k(x_2, x_2) - 2k(x_1, x_2)$. Therefore kernel space representation of (5) is defined as

$$F^{\phi(x_1),\phi(x_2)}(\phi(x)) = \frac{k(x_1,x_1) - k(x_1,x) + k(x_2,x) - k(x_1,x_2)}{\sqrt{k(x_1,x_1) - 2k(x_1,x_2) + k(x_2,x_2)}} \quad (6)$$

The above expression gives the formulation of line projection in kernel space defined by pivot objects $(\phi(x_1), \phi(x_2))$. Equation (6) can be discretized by defining the thresholds as discussed in the section 2.1. Following the procedure discussed in section 2.1, we can generate family of hash functions H_{KDBH} by applying the discrete hash functions defined for mapping function (6). The indexing framework and retrieval procedure remains same as traditional DBH (section 2.1).

3. DISTANCE BASED HASHING FUNCTION LEARNING THROUGH MKL

3.1 Optimization problem formulation

Equation (6) represents line projection formula in kernel space which defines the Kernel based DBH. The selection of optimal kernel can be performed by defining a learning based framework which is similar to MKL applied for various recognition problems. Therefore we can learn the kernel K for hashing by parameterized linear combination of the set of base kernels, i.e. $k(x_1, x_2) = \sum_{i=1}^{q} w_i k_i(x_1, x_2)$ is considered. The resultant kernel should satisfy Mercer's condition therefore all the weights should be positive real number i.e. $\forall i,\ w_i \geq 0$. The maximization of precision based retrieval defines the optimization objective. We use the labeled data information for evaluation of the objective. The limited amount of labeled data may lead to the condition of over fitting, therefore a regularizer term is required in the objective. In practice, the DBH partitions object space uniformly without considering the object data distribution. However most of the real datasets are uniformly distributed. Therefore, for a hashing scheme to be efficient, each hash function should have 50% probability of getting 1 or 0, and the hash functions should be correlated [13]. We apply the maximum entropy principle based regularizer which ensures partition balancing constraint. Following the result in [12], the maximization of hash values $h(X)$ satisfies maximum entropy condition for a hash function. The complete optimization objective of MKL problem is defined as.

$$w^* = \texttt{argmax}_w \quad \mathbf{F}(X, X_v, w) \tag{7}$$

$\mathbf{F}$ is defined as $\texttt{mean}\{\mathbf{J}(X_v,\ w)\} + \lambda \mathbf{V}(X,\ w)$

X is the complete training set, X_v is part of training data assumed to be available with label information for which function $\mathbf{F}$ is evaluated for different weight parameter w. λ is the regularization parameter. The function $\mathbf{J}(X_v,\ w)$ represents the retrieval performance of KDBH computed over X_v and w, and is defined as

$$\mathbf{J}(X_v, w) = \sum_{i=1}^{|X_v|} \delta(y_i, \hat{y}(x_i, w))$$

y_i represents the correct label set and $\hat{y}_i$ represents the predicted label for each object $x_i \in X_v$. Function $\delta(\cdot\,,\cdot)$ represents the Kronecker delta defined as

$$\delta(a, b) = \begin{cases} 1 & a = b \\ 0 & \text{otherwise} \end{cases}$$

The function $\mathbf{V}(X_v,\ w)$ represents regularizer term defined as the sum of variance of the hash values for all hash tables which is computed as

$$\mathbf{V}(X, w) = \sum_{i=1}^{L} \texttt{mean} \sum_{j=1}^{k} \texttt{Variance}\{h_{ij}(X, w)\}$$

3.2 GA based optimization framework for MKL

The weight parameters $\{w_i \text{ for } i = 1, ..., q\}$ in the equation (7) are optimization parameters. The existing MKL formulations developed for various recognition problems have objective functions which are continuous in nature and apply conventional gradient based methods for optimization [4, 3, 9, 10]. The current optimization problem (Equation (7)) is discrete in nature while the parameter space is continuous. The discrete nature of optimization objective restricts the application of gradient based algorithms. For such optimization tasks, the Evolutionary Algorithms can provide efficient solution. The Genetic Algorithm (GA) is a type of Evolutionary Algorithm which is well suited for global optimal parameter search in complex spaces. In addition, GA has the advantage of working with raw objectives when compared with conventional techniques. Therefore we formulate MKL for indexing in GA like paradigm. The function $\mathbf{F}$ defines fitness function for GA population string. The iterative process of selection and regeneration of individuals in the population is based on the evaluation of F as retrieval performance for a validation query set.

The labeled data X_v is used as validation query set, and the precision oriented retrieval i.e. average precision in K nearest neighbors (mean of correct matches in K nearest neighbors) in retrieved result represents the fitness value for population individual. The tournament selection is applied for the selection of individuals for successive population generation. The process selects p individuals randomly from the current population, and individual with highest fitness among the selected p is placed in Mating Pool. The process is repeated for M times, here M is population size and p is tournament size. The reproduction operators for offspring generation from the individuals selected in Mating Pool consists of single point crossover and uniform mutation. The construction of new population for successive genetic algorithm iteration is performed by applying elitist selection strategy. The elitist selection combines offspring with current population and selects M best individuals with based on their fitness value. The distance computation for nearest neighbor search is performed in kernel space using equation (5).

	Algorithm: GA for MKL
1	Population generation $\Rightarrow$ *Pop*=`Generate`()
2	Population initialization $\Rightarrow$ `Initialization`(*Pop*): `Evaluate`($\mathbf{F}$) for each individual in *Pop*
	For each **i** < **noIterations**
3	Selection of individuals for successive population generation $\Rightarrow$ *Pop_sub* = `Selection`(*Pop*) using *Tournament Selection*
4	Offspring generation step 1 $\Rightarrow$ *Pop_1* = `Crossover`(*Pop*) using *Single Point Crossover*
5	Offspring generation step 2 $\Rightarrow$ *Pop_2* = `Mutation`(*Pop_1*) using *Uniform Mutation*
6	Evaluate Offsprings $\Rightarrow$ `Evaluate`($\mathbf{F}$) for each individual in *Pop_2*
7	New population generation $\Rightarrow$ *Pop_new* = `Generate`(*Pop*, *Pop_2*) using *Elitist Selection*
	end

4. LOCAL TEXTURE FEATURES

Texture feature is an important cue for image analysis. It has been extensively used for various content based image retrieval applications. The image texture is defined as the set of local neighborhood properties of the gray levels of an image region. The texture features represent the image at multiple resolutions which contain spatial as well as frequency information of the image.

Fundamentally, an image can be considered as a mosaic of different texture regions. In our texture based feature representation, we consider the image as the mosaic of different overlapping texture regions. The regions are identified as the key point neighborhoods. For key point identification, we follow the procedure used for SIFT key point identification [5]. The initial step of key point identification is done by analysis of image scale space at multi scale. The key points are obtained as the local maxima/minima points obtained after difference of Gaussian smoothened image $g(x, y)$ applied in scale space. The selection of maxima and minima locations as key points helps in achieving rotational invariance in key point selection.

$$g(x, y) = \frac{1}{\sqrt{2\pi}\sigma} exp- \frac{x^2 + y^2}{2\sigma^2}$$

The efficient identification of key points can be performed by generating the image pyramid by resampling between each level. The detection of maxima and minima is performed by comparing each pixel in the pyramid with its 8-neighbors. First the comparison is performed at the same level of the image pyramid. If the pixel is maxima or minima at this level, closest pixel location at next lower level is identified by performing 1.5 times resampling. If the pixel remains lower or higher than the closest pixel location and its 8-neighbors, we compare the pixel with closest pixel location and its 8-neighbors at the image a level above. If the pixel corresponds to local maxima or minima location point, we consider this pixel as key point (Figure 1). We have considered Haar wavelet response to define the texture property of the key point neighborhood. We have considered, $W \times W$ pixel neighborhood around centered around each point for wavelet response computation. The noisy key points i.e. the points for which the neighborhood region of $W \times W$ pixel crosses the image boundary have been neglected. Each key point is represented by approximation coefficient obtained after 2-scale decomposition of neighborhood image. We consider only the approximation coefficient as fine scale wavelets capturing high frequency details are inefficient in characterization of different object details. To enhance the effectiveness of wavelet response against intensity value transitions, we normalize coefficients value with mean of approximation coefficients in the neighborhood. Following the above procedure, set of local feature vectors for each image is obtained. Here each local feature vector is associated with a key point. For a unique feature extraction method many key point neighborhoods from different images as well as the same image are similar in terms of feature value. Therefore we can associate each key point neighborhood to a *word* and, we can define the image representation based on *bag of words* model. We can generate a *visual vocabulary* using all the local feature vectors from the training images by performing *k-means* clustering. The clustering generates a *visual vocabulary* having p *words*. The *words* are defined by the learned cluster centers. The *visual vocabulary* based image representation improves robustness to minor variations local feature vectors corresponding to similar image regions.

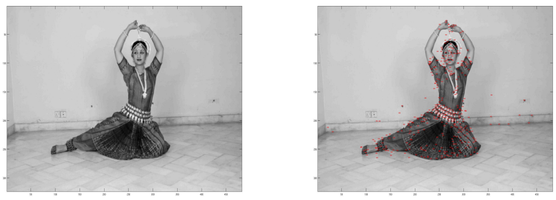

Figure 1: Distribution of key points on an image

Using the *visual vocabulary*, each image can be represented by a vector which is basically the histogram of *word* frequency appearing in the image. The number of histogram bins as equal to p, i.e. number of unique words. Formally the Image I is represented as vector $\overrightarrow{I} = \{sw_1 : sw_2 : \cdots : sw_p\}$. Here, the value sw_i represents the occurrences of i^{th} *word* in the image. The initial evaluation of the proposed feature is performed by applying it for annotation task over a collection of Indian classical dance posture images discussed in [6]. The experimental images belong to collection of Odissi and Bharatnatyam dance postures. All the images belong to medium sizes group therefore the neighbor window parameter W is fixed as 33. For the selection of number of bins for indicator vector computation corresponding to texture feature, we performed initial experiments with 50,60,100 and 150 bins. For both the collection, the indicators vectors computed with 100 bins gave the best result. The comparison of the proposed feature (LWV) is performed with SIFT feature. For the selection of number of bins for indicator vector computation corresponding to SIFT feature, we performed initial experiments with 50,100 and 150 bins. For Odissi collection indicators vectors computed with 100 bins, and for Bharatnatyam collection indicators vectors computed with 50 bins gave the best result. The comparison results are presented in the table 1 and 2.

Table 1: With Odissi dance image collection

	KNN	SVM
SIFT	87.63	92.04
LWV	**88.26**	**91.96**

Table 2: With Bharatnatyam dance image collection

	KNN	SVM
SIFT	79.46	81.22
LWV	**83.20**	**84.18**

5. EXPERIMENTAL RESULTS AND DISCUSSION

The experiments have been performed on two standard datasets. First, the CIFAR-10 dataset is selected which has primarily been used for object recognition problems. The dataset is the labeled subset of **80 million image dataset**[3]. The image set consists of 60000 color images of size 32x32 and belong to 10 different categories. There are 6000 example images belonging to each category and dataset is partitioned as 50000 training images and 10000 test images. Next, we test the proposed MKL framework over the MNIST dataset. The dataset has been primarily used for testing handwritten digit recognition and classification algorithms. The dataset contains 60000 training and 10000 test images of handwritten digits. Each image is a 28x28 image displaying an isolated digit between 0 and 9.

For all the MKL experiments, common coding and stopping criterion is applied for Genetic Algorithm. Each kernel weight parameter is encoded with 6-bit binary string. The

[3]http://groups.csail.mit.edu/vision/TinyImages/

initial population set for experiments consisted 40 individuals. The iterative GA optimization cycle is run for 100 iterations for parameter optimization. The details of the algorithm have been presented in section 3.2.

Corresponding to both the datasets, we have selected 5% of the complete training images as the set of pivot objects for generation of family of hash functions. For MNIST dataset X_v consists of 600 images i.e. 60 images selected randomly from the subset of images belonging to each class. For CIFAR dataset X_v consists of 500 images i.e. 50 images selected randomly from the subset of images belonging to each class. The precision and Mean Average Precision (MAP) in K nearest neighbors is considered as the performance measure as in [12]. The MAP is computed as follows

$$\mathrm{MAP}(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk})$$

Average Precision takes the mean of precision at each recall point (i.e. location of relevant result) in the retrieved results. MAP takes the mean of Average Precision over all queries. Q represents the set of query, m_j is the number of relevant retrieved results for query $q_j \in Q$. The R_{jk} represents the ranked retrieval results from the top to k^{th} relevant results. When a retrieved document in R_{jk} is not relevant, the precision value in the above equation is taken to be 0 [7]. The selection for parameter λ is done with some preliminary experiment such that the contribution of terms $\mathbf{J}$ and $\lambda\mathbf{V}$ in $\mathbf{F}$ (Equation (7)) is of same order. The value of λ is selected as 0.025 and K as 20 for performance evaluation. The experimental results for both datasets is following.

5.1 Results with CIFAR-10

The experiments on the dataset have been performed using the GIST feature and the proposed texture feature. The GIST feature represents the orientation energies in the image at different scales an orientation [8]. The GIST descriptor defines each dataset image by a vector having 384 element. The set of base kernels for experiments with GIST feature included a linear and set of Gaussian kernels var: 0.1, 1, 2, 5.

Considering the small size of the dataset images the neighbor window parameter W is fixed as 7, i.e. Local texture feature (LWV) is computed using a neighbor window of 7×7. For visual *visual vocabulary* generation for the dataset, we experimented with 25, 40 and 50 visual words and finally selected 40 visual words for computation of Indicator vectors. The results for various hash table parameters (L, K) is in table 5.1. The set of base kernels for experiments with texture feature included set of Gaussian kernels {var: 0.1, 1, 5, 25, 100}. In experiments for combining the two features, the set of base kernels applied individually is ORed. The performance of the texture feature is poor in comparison with GIST feature. The low density of the key points because of the low resolution (32x32) of dataset images is the primary reason. However, the complementary nature of information present in both the features improves the indexing performance by learning the optimal combination through MKL.

5.2 Results with MNIST dataset

For experiments with MNIST, we have selected raw images representing its gray scales intensities as the feature values. Thus each image will represented by a 784 dimensional vector. The set of base kernels included a linear and set of Gaussian kernels {var: 1, 10, 100}. The experimental results in comparison with basic DBH for different hash table parameters is presented in table 5.2.

Table 3: Results with CIFAR-10 dataset

$L = 20$, $k = 18$			
	MAP	**Precision**	**Avg. Comps.**
MKLDBH{GIST}	0.42	0.46	3208
DBH{GIST}	0.40	0.42	3981
MKLDBH{LWV}	0.19	0.23	2208
DBH{LWV}	0.19	0.20	2781
MKLDBH{GIST + LWV}	0.44	0.48	3173
$L = 24$, $k = 25$			
	MAP	**Precision**	**Avg. Comps.**
MKLDBH{GIST}	0.41	0.49	3012
DBH{GIST}	0.40	0.42	3523
MKLDBH{LWV}	0.24	0.24	2132
DBH{LWV}	0.23	0.22	2531
MKLDBH{GIST + LWV}	0.44	0.49	3037

Table 4: Results with MNIST dataset

$L = 20$, $k = 18$			
	MAP	**Precision**	**Avg. Comps.**
MKLDBH	0.70	0.73	1827
DBH	0.68	0.69	1923
$L = 24$, $k = 25$			
	MAP	**Precision**	**Avg. Comps.**
MKLDBH	0.70	0.75	1699
DBH	0.69	0.72	1789

The increase in number of hash bits k increases the precision, but the collision probability $\{\mathrm{P}\{g(x) = g(y)\}, g = [h_1, h_2, ..., h_k]\}$ decreases with increase in k. However the increase in number of tables compensates the decrease in recall at the increasing cost of search time complexity (number of comparisons).

6. CONCLUSIONS

The paper presents novel hash function learning framework through MKL. The DBH is applied for hashing which preserves the distance between objects while projection. The framework proposes novel application of Genetic Algorithm for solving MKL optimization. A novel texture based image representation using *bag of words* model is presented. The experimental results of the proposed framework is presented for two standard dataset. The framework also demonstrates the capability to combine multiple features for indexing. The evaluation of the proposed framework for wider range of datasets is the future work.

7. REFERENCES

[1] A. Andoni and P. Indyk. Near optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

[2] V. Athitsos, M. Potamias, P. Papapetrou, and G. Kollios. Nearest neighbor retrieval using distance based hashing. *In Proceedings of IEEE International Conference on Data Engineering*, pages 327–336, April 2008.

[3] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. *In the Proceedings of International Conference on Computer Vision*, pages 1–8, 2009.

[4] G. R. G. Lanckriet, T. D. Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

[5] D. G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*, 2:1150–1157, 2000.

[6] A. Mallik, P. Pasumarthi, and S. Chaudhury. Multimedia ontology learning for automatic annotation and video browsing. *In The Proceeding of the 1st ACM international conference on Multimedia information retrieval*, pages 387–394, 2008.

[7] C. D. Manning, P. Raghavan, and H. SchÃijtze. *An Introduction to Information Retrieval.* Cambridge University Press, Cambridge, 2009.

[8] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[9] A. Rakotomamonjy, F. bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. *In The Proceedings of International Conference on Machine Learning*, 772:775–782, 2007.

[10] S. Sonneburg, G. Ratsch, C. Schafer, and B. Scholkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, December 2006.

[11] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. pages 606–613, September 2009.

[12] J. Wang, S. Kumar, and S.-F. Chang. Sequential projection learning for hashing with compact codes. *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[13] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. *In The Proceedings of NIPS*, pages 1753–1760, 2008.