International Journal of Engineering& Scientific Research
Vol. 6 Issue 3, March 2018
ISSN: 2347-6532 Impact Factor: 6.660
Journal Homepage: http://esrjournal.com, Email: esrjeditor@gmail.com Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Dimensionality Reduction for Biometric Face Image Recognition

Dhruva Adike (NIT Calicut)

Abstract:

The aim of the project is to reduce the dimensionality of the occluded biometric face images so that we could store and search efficiently. The main purpose is to use lesser space and better retrieval of images. For this sake, we will use the techniques used in locality sensitive hashing (LSH), minimal loss hashing (MLH) and iterative quantization (ITQ) where the data vector when randomly projected will be generating hyper-plane.

Key words- dimensionality reduction; MLH; LSH; ITQ; biometric recognition.

Introduction:

The term biometrics refers to the automatic recognition method for identifying and authenticating people based on the measurable characteristics and distinctive features of humans. Commonly used physiological biometrics are face image, fingerprint, iris pattern, hand geometry, vascular configuration of retina, and DNA. Unlike physiological biometrics that are based on body characteristics, behavioral biometrics are mostly based on muscle control and movement such as signature, keystroke and voice.

Biometric technologies have been commonly used for two main purposes: authentication and identification. Authentication based on biometrics aims at verifying if the individual is the same person as the claimed identity, whereas biometric identification attempts to determine identity of an un- known individual. Biometric authentication requires a one-to-one matching of the biometric identifier against the stored biometrics, while identification is a one-to-many matching of an identity against the whole database. We use feature selection and feature extraction techniques adopted to reduce the dimensions of the input feature space. Given the original set of input features, feature selection techniques choose a subset of most relevant features, but feature extraction techniques transform data into a lower dimensional space before the selection of discriminative features.

Motivation - Efficiency in terms of speed and ease of use along with accuracy and security are the major issues concerning the practicality of biometric systems. Byreducing the dimensions in images, we speed up the authentication and identification process. Reducing the dimensions of the biometric image implies in speeding up the authentication and identification processes done during biometric recognition. There are many problems when it comes to high dimensional images than low dimensional images as the space increases so fast that the available data becomes sparse which makes data organization inefficient. This project is to solve the problems by using algorithms that efficiently work on these high dimensional images.

Problem Statement - Aim of this project is to evaluate the dimensionality reduction techniques which converts biometric images(face) from real space to binary space while preserving the similarity using distance matrices. This is verified for invariance of biometric images which are scaled and blurred.

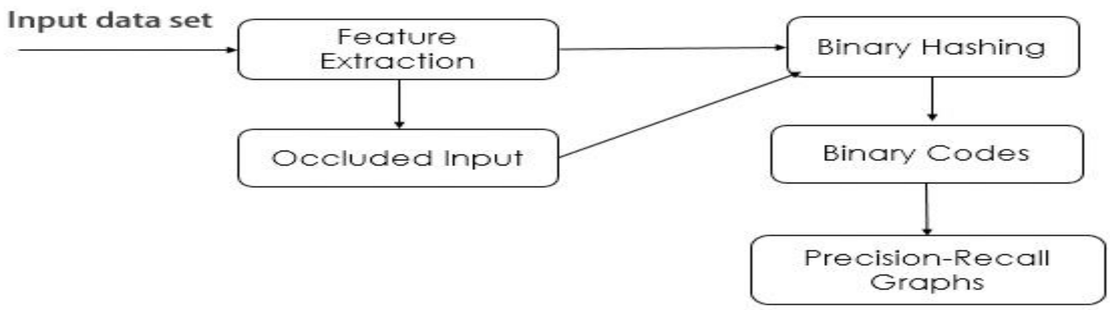
Related Works - The main focus is on learning compact codes. Early approaches showed how one might preserve semantically meaningful data abstractions. Multi- layer neural networks worked well for document retrieval. However, these methods exhibit poorer performance than more recent methods.

Recently, work is going on in the problem of learning similarity-preserving binary codes for representing large scale images. Encoding high dimensional images to compact binary codes can improve efficiency in terms of storage and speed for similarity search.

Presentation:

The dataset used is the labeled faces in the wild (LFW) dataset which consists of 13233 images of 5749 people, of which 1680 people had more than one image. The images are processed into real valued by the help of Gist descriptor in the first stage of the project and later by OverFeat in the later stage. This process is known as feature extraction. The idea is to develop a low dimensional representation of the scene, which does not require any form of segmentation. This algorithm takes an image as an input and process the image through various filters.

Figure 2.1: High level design



The tool used for implementing the gist descriptor is MATLAB in our project. When computing image similarities, it might be important to normalize the image size before computing the GIST descriptor. The dimensions of the matrix computed by the gist descriptor is 13233 x 512 where 13233 is the total number of images in the dataset and 512 is initially set in the algorithm of gist descriptor. This is obtained by convolving the image with 32 Gabor filters at 4 scales, 8 orientations producing 32 feature maps of the same size of the input images. Divide each map into 16 regions (by a 4 x 4 grid) and then average the feature values within each region. Concatenate the 16 averaged values of all 32 feature maps resulting in a 16 x 32 = 512 Gist descriptor.

The matrix obtained by the gist descriptor is taken as the input data. After the feature extraction process, we apply various techniques to the dataset to reduce the real valued data to compact binary code while preserving the similarity of the original dataset. Applying the techniques on the matrix obtained by gist yields binary valued matrix which is easier to compute as the matrix obtained is of low space.

The measure of similarity is obtained between binary valued matrix and original matrix by computing precision-recall graphs. These precision-recall graphs are helpful in choosing the algorithm that is the best alternative to a given dataset.

The accuracy of all the techniques can be computed by using the algorithm Information Theoretic Metric Learning(ITML). The efficiency in terms of time and space is achieved at the cost of reduced accuracy. So the algorithm can to used to decide which technique to be used for reducing the dimensions of the original dataset so that accuracy obtained remains closer to the accuracy of the original dataset.

Discussion:

We implemented MLH, LSH and BRE algorithms to the files that are obtained after applying Gist and OverFeat feature extractors. We then scaled down the original image dataset and applied the above algorithms. We applied ITML algorithm for comparing the accuracy of the results obtained in the previous methods.

Stage 1

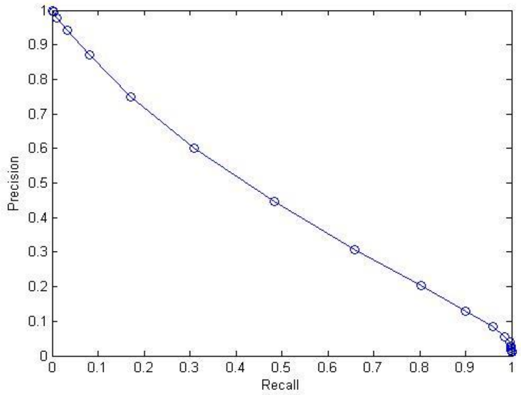
The dataset is taken from the labeled faces in the wild website as already mentioned in the high-level design. All the images in the dataset are of dimensions 250 x 250. Gist descriptor is applied to this dataset and a 13233 x 512dimensional matrix is obtained. This matrix is used for further conversion to binary compact codes.

This Input data is further used as a medium for various algorithms to convert it into Binary compact codes. The applied algorithms are as follows:

Iterative Quantization - We apply linear dimensionality reduction to the data and then perform binary quantization in the resulting space. Initially we follow maximum variants formulation which yields PCA projections. We then try to preserve the locality structure of the projected data by rotating it so as to minimize the discretization error. We adopt clustering method to find the local minimum of the quantization loss and specifically we use k-means clustering. We obtain a binary valued matrix while evaluating for minimum quantization loss. In this project, we considered taking 50 iterations is enough for computing the minimum quantization loss.

The evaluation of a performance is done by nearest neighbor search using Euclidean neighbors as the ground truth for computing the true positives and true negatives, and based on this we compute the mean average precision.

The following precision-recall graphs are plotted for a given number of iterations and code-length.



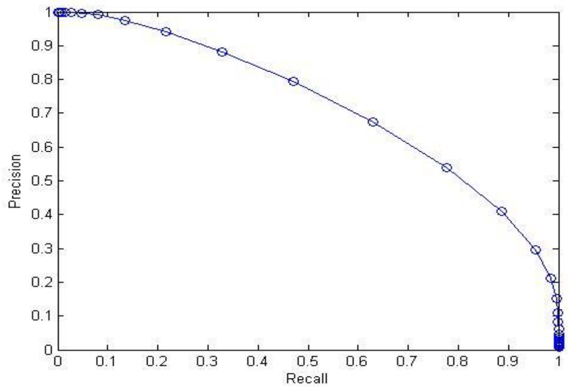
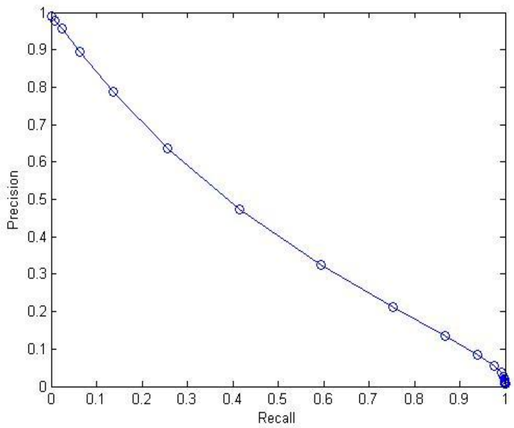


Fig4.1: No. of iterations=50 code-length=32 Fig4.2: No. of iterations=50 code-length=64



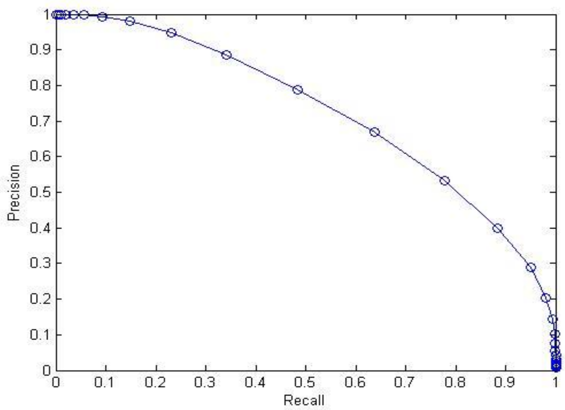
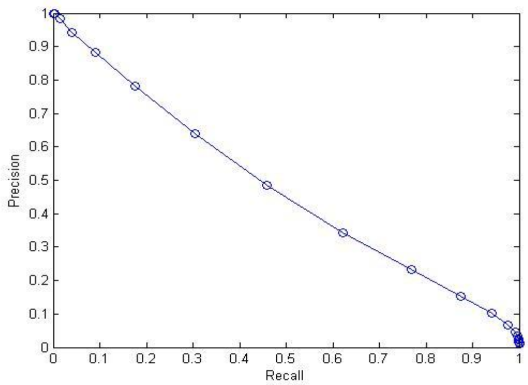


Fig 4.3: No. of iterations=40 code-length=32 Fig 4.4: No. of iterations=40 code-length=64



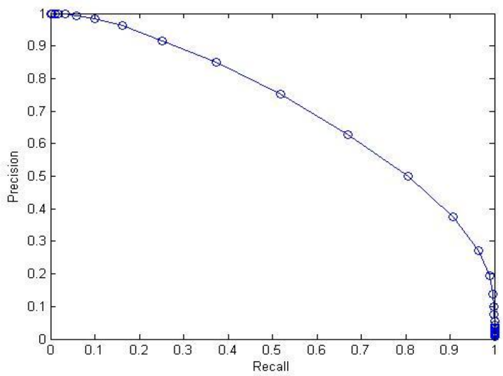
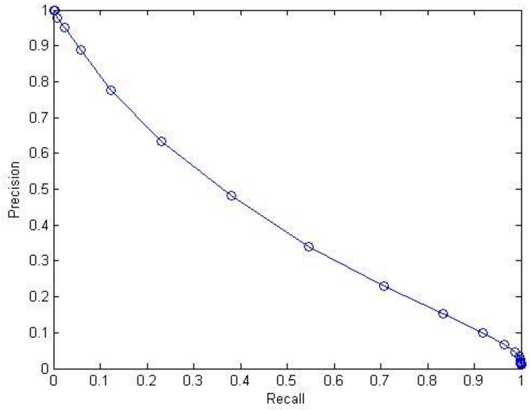


Fig 4.5: No. of iterations=30 code-length=32 Fig 4.6: No. of iterations=30 code-length=64



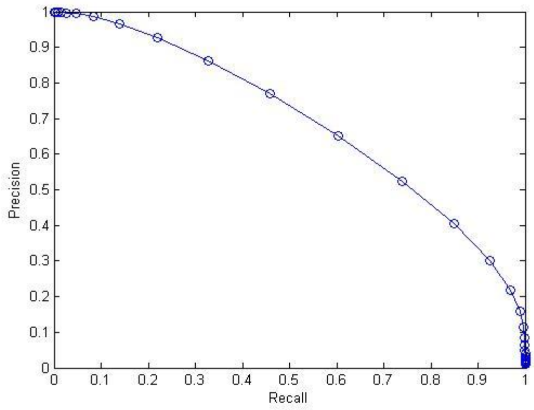


Fig 4.7: No. of iterations=20 code-length=32Fig 4.8: No. of iterations=20 code-length=64

We can infer that number of bits that are used to define a vector impact on the similarity of the matrix to the original. When we further increase the number of iterations from 50, the change observed is negligible. So we can infer that the optimal number of iterations to be 50.

Locality Sensitive Hashing - LSH is based on the simple idea that, if two points are close together, then after a projection operation these two points will remain close together. Locality sensitive hashing is a randomized algorithm that allows one to quickly find similar entries in large datasets. We start with a random projection operation that maps a data point from a high-dimensional point to a low- dimensional subspace.

First, we note which points are close to our query points. Second, we create projections from a number of different directions and keep track of the nearby points. We keep a list of these found points and note the pointsthat appear close to each other in more than one projection. At the core of LSH is the scalar projection given by the hash function which is the dot product of the query point which is in a high-dimensional space with a vector whose components are selected at random from a Gaussian distribution. This scalar projection is then quantized into a set of hash bins, with the intention that nearby items in the original space will fall into the same bin.

Minimal Loss Hashing - The basic function of MLH is to formulate the binary compact codes using the loss functions which are designed using hashing. Nearby data have similar codes and the dissimilar data have different code. The quality of the mapping is determined by loss function that assigns the cost of pair of binary codes and similarity label.

The points that are close-by have similar codes while that are farther have different codes.

Hashing function: b(x) = Threshold(Wx)

In the above hashing function, Threshold denotes the binary quantization function, W is a parameter matrix and x is input feature vector.

The i^{th}bit of the vector threshold(Wx) is 1 if and only if the i^{th}element of (Wx) is positive. In other words the i^{th}row of W determines the i^{th}bit of the hash function in terms of a hyper plane in the input space; 0 is assigned to points on one side of the hyper plane and 1 points on the other side.

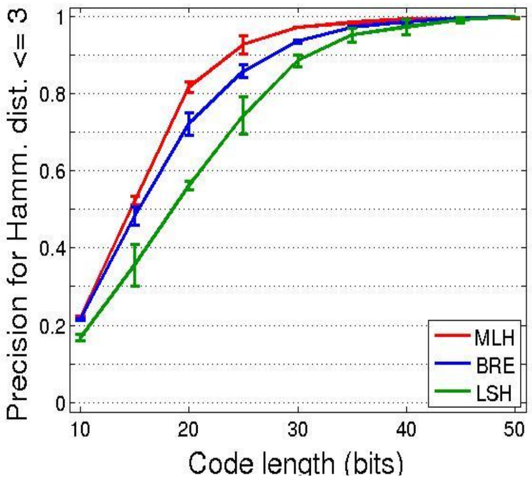
In this case the loss function is specific to learning binary hash functions, it includes a hyper-parameter p(rho), which is a threshold in the hamming space that differentiates neighbors from non-neighbors. We learn hash keys which are important for similar training points to map to binary codes that differ by no more than p(rho) bits, non-neighbors should map to codes no closer than p(rho) bits. This is a similarity preserving binary hashing for formulating compact binary codes as binary codes are storage efficient and takes a sub linear time to search.

Binary Reconstructive Embedding - It is a variant of MLH and uses a loss function that penalizes the difference between Euclidean distance in the input space and Hamming distance between binary codes.

The hash function is for this technique is found by empirical loss, i.e., the sum of the pairwise loss mentioned above, over training pairs.

Results of stage 1

The mentioned algorithms are executed using the gist descriptor feature extractor. The results obtained are used as a benchmark for comparing the results that will be obtained in the later stages.



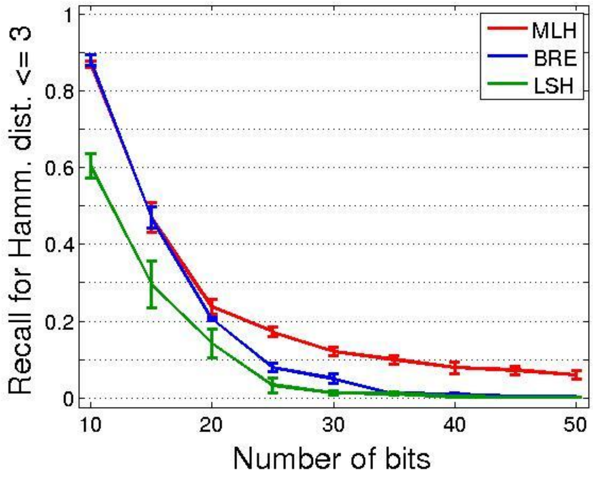
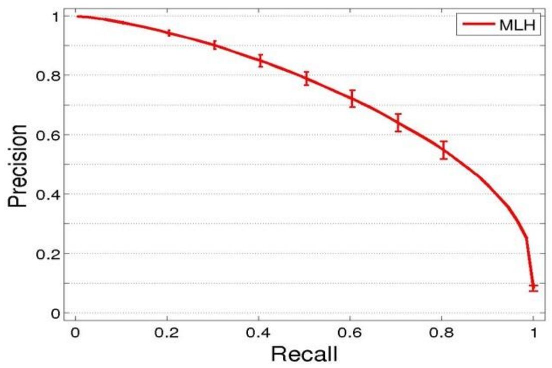


Fig 4.9: Precision formed from all curves Fig 4.10: Recall formed from all curves



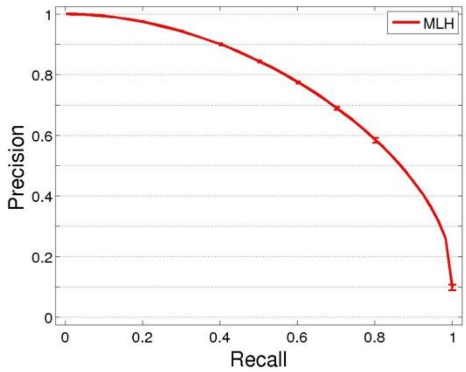


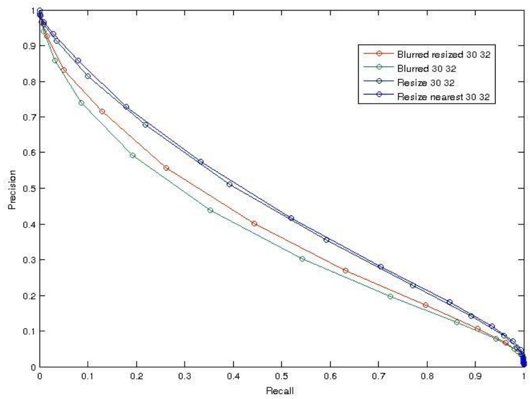
Fig 4.11: Precision Recall graph for 30 bits. Fig 4.12: Precision Recall graph for 50 bits.

Stage 2

In the second stage, we scaled down the original dataset by resizing and blurring using the tools available in MATLAB. Im-resize is the tool used to resize the image. Fspecial tool is used to blur the image which creates a pre-defined 2-D filter of the specified type and the type we selected is Gaussian low-pass filter. We have also used Nearest neighbor method on im-resize where the output pixels are assigned the value of the pixel that the point falls within. No other pixels are considered. We applied Gist descriptor to all the scaled down images obtained by the above techniques.

Results of stage 2

The results obtained by all the four scaled algorithms are compared after applying ITQ techniques for a given number of iterations and code-length.



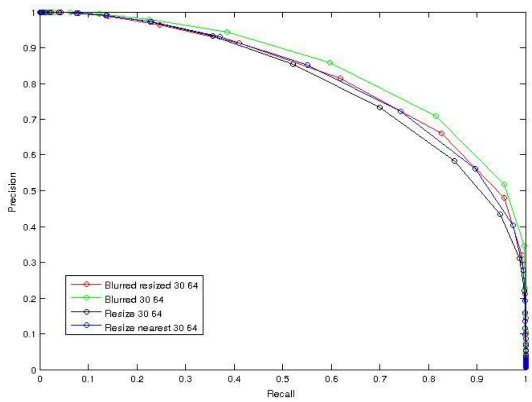
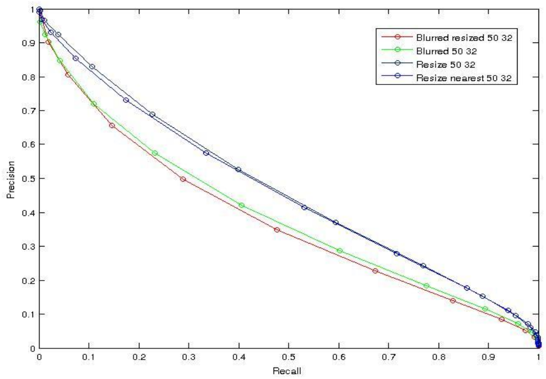


Fig4.13: No. of iter. =30; code-length=32 bitsFig 4.14: No. of iter.=30; code-length=64 bits



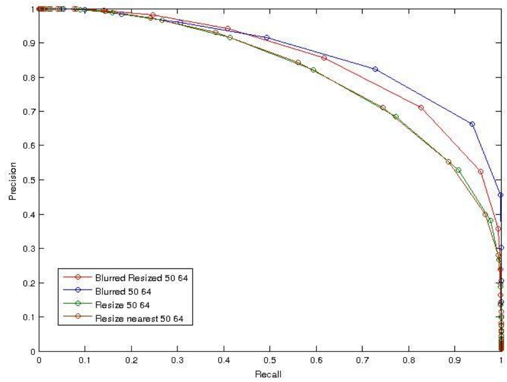
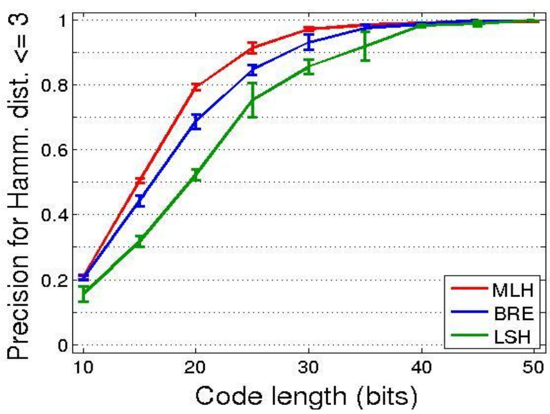


Fig 4.15: No. of iterations=50; length=32 bits Fig 4.16: No. of iterations=50; length=64 bits

From the above graphs we can infer that blurred dataset is performing better when compared to the other datasets when the code-length is 64bits. So, we have applied MLH, BRE and LSH techniques to the blurred dataset which yielded the precision-recall graphs as in figures 4.17-4.20.



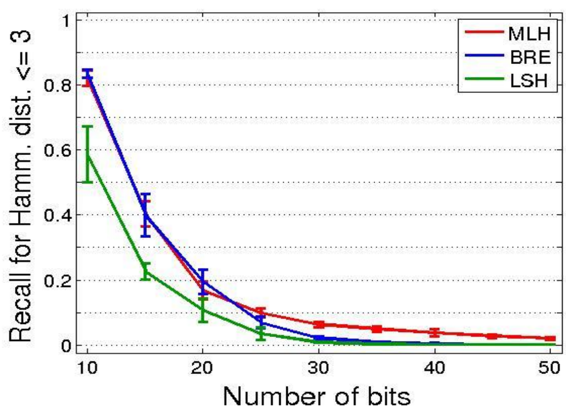
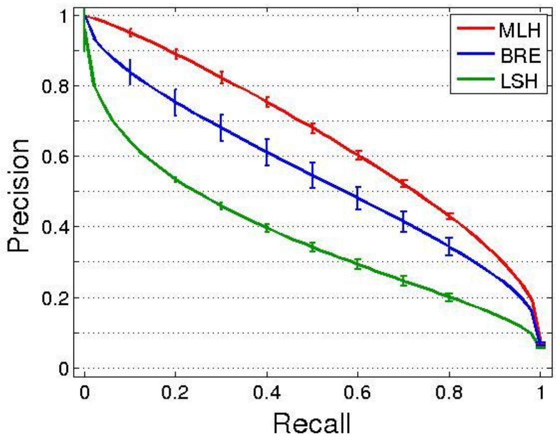


Fig 4.17: Precision of curves in blurred dataset Fig 4.18: Recall of curves in blurred dataset



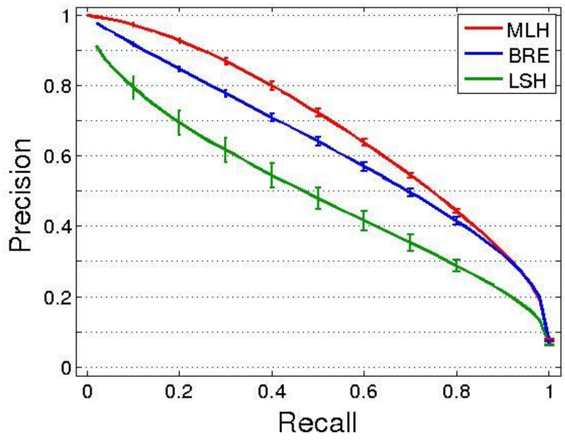


Fig 4.19: Precision-recall graphs for 30 bits Fig 4.20: Precision-recall graphs for 50 bits

Stage 3

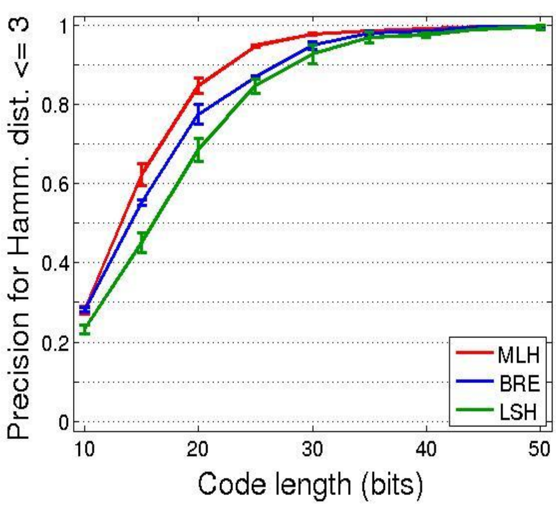
In this stage we are concerned with the accuracy that is obtained by the various scaling techniques mentioned above by an algorithm ITML and we also apply a new feature extractor OverFeat to the original dataset.

OverFeat- In the final stage, we use a different feature named OverFeat which generates 4096 feature vectors while Gist generates 512 feature vectors. OverFeat is a convolutional network based image classifier and feature extractor. So OverFeat reduces the information loss when compared to the loss incurred in Gist feature descriptor.

Information Theoretic Metric Learning - We use Information Theoretic Metric Learning(ITML) algorithm in this stage to compute the accuracy obtained in the above techniques. In this algorithm we find a suitable metric for a given set of data-points with information regarding distances between data-points. ITML runs metric learning with cross-validation and k-nearest neighbors. It uses Mahalanobis distance function and learns the associated parameters using Bregman'scyclic projection algorithm.

Results of stage 3

Results obtained from the ITML are as follows : KNN cross validated accuracy on the gist applied original dataset is 0.901836, on resized images using 50 bits is 0.818606, blurred images with 50 bits is 0.775507, and on blurred resized images with 50 bits is 0.738228. Here we can observe that the accuracy obtained in blurred and resized blurred images is less than the accuracy obtained in resized images. Hence for our dataset resized image dataset is giving better result in terms of accuracy. The results obtained from the OverFeat feature extraction of the original dataset images are plotted in the figures 4.21-4.24.



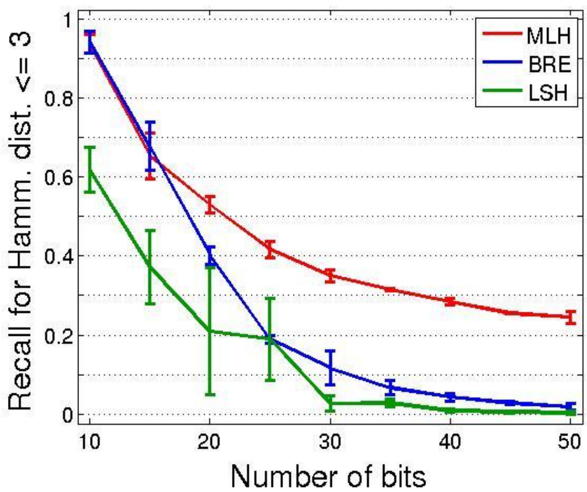
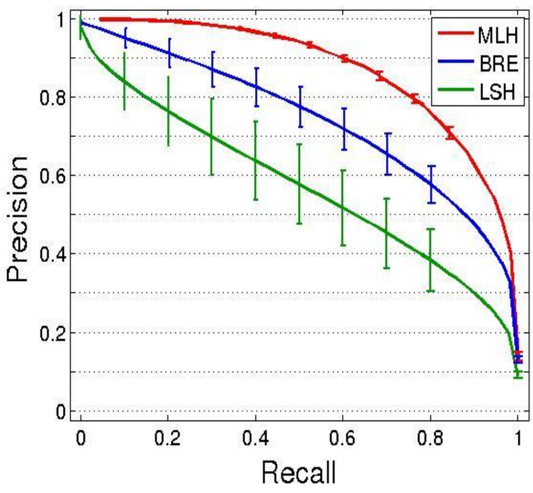


Fig 4.21: Precision of OverFeat feature extractedFig 4.22: Recall of OverFeat feature extracted



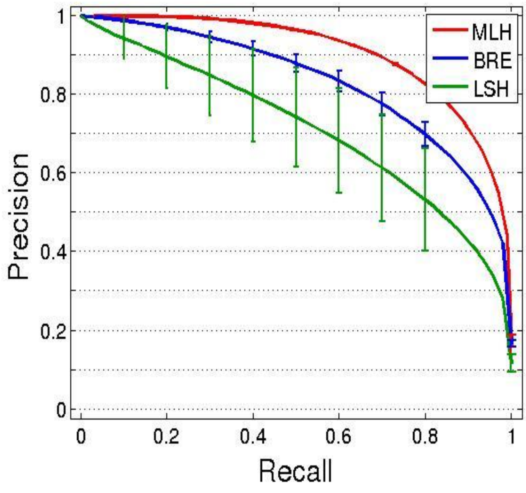


Fig 4.23: PR graph by OverFeat with 30 bits Fig 4.24: PR graph by OverFeat with 50 bits

Comparing the precision-recall graphs obtained by Gist on blurred dataset and OverFeat on the original dataset.

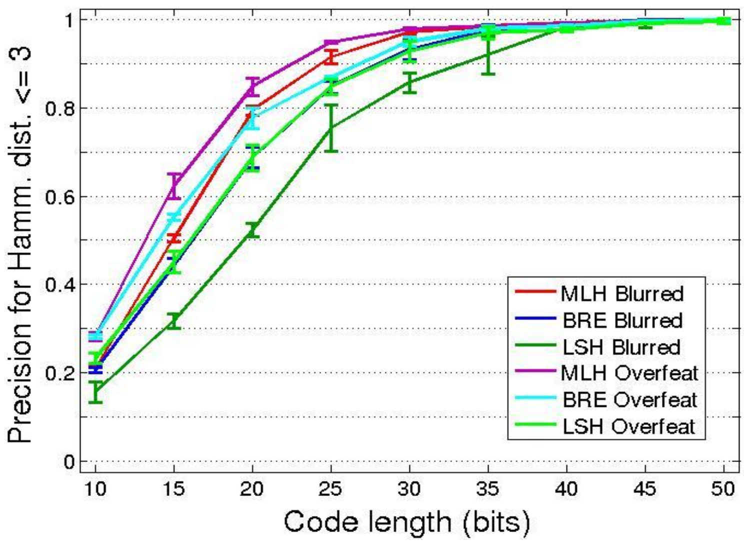


Figure 4.25: Precision graph obtained by OverFeat vs Gist on blurred dataset

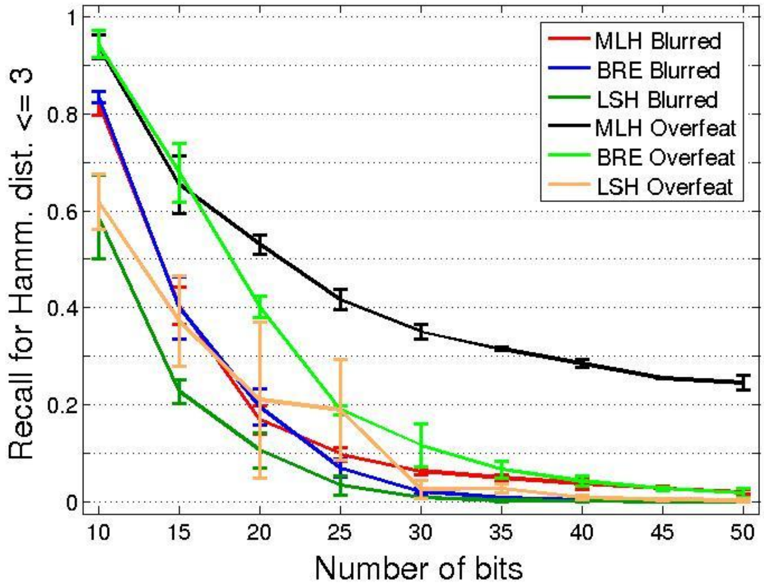


Figure 4.26: Recall graph obtained by OverFeat vs Gist on blurred dataset

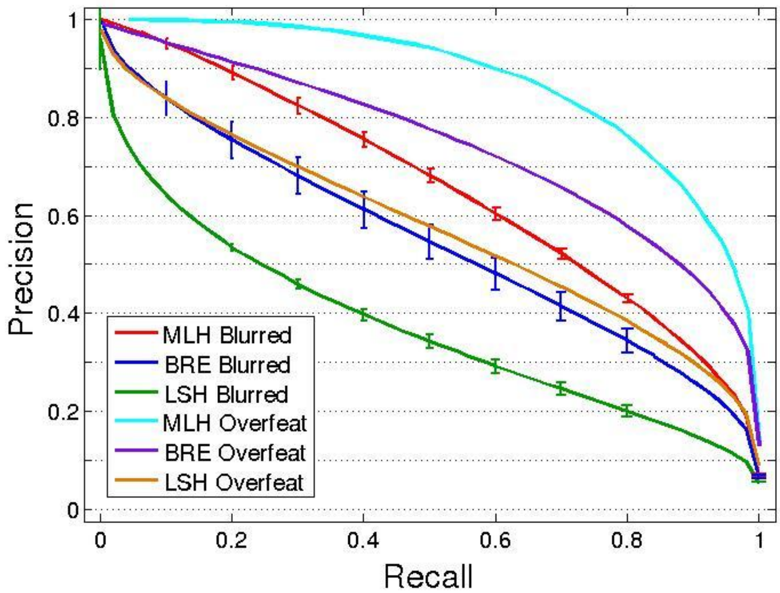


Figure 4.27: Precision-recall graph for 30 bits obtained by OverFeat vs Gist on blurred dataset

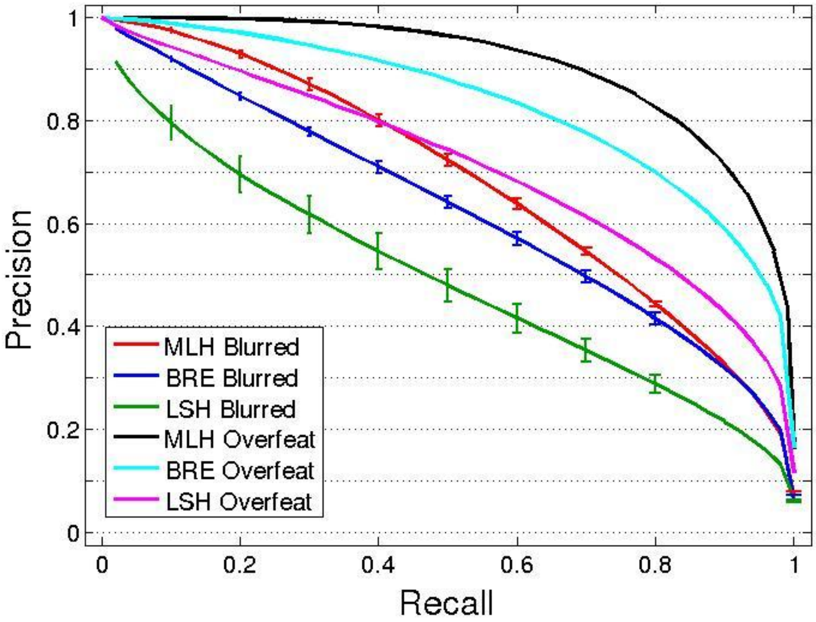


Figure 4.28: Precision-recall graph for 50 bits obtained by OverFeat vs Gist on blurred dataset

The figures 4.25 to 4.28 shows the compromise on the precision recall graphs obtained in case of applying OverFeat to original dataset and Gist on blurred dataset.

Conclusion:

Encoding the original image dataset to binary mapping of the real valued vectors improves both time and space by preserving the similarity of the binary codes to the original image dataset. From the results obtained in the form of precision-recall graphs, it can be inferred that similarity is preserved better when the images are scaled down. In this dataset we have used scaling down using blurring technique which preserved the similarity even better. Results obtained by OverFeat are better than the results obtained by Gist as the number of feature vectors generated by OverFeat is more than that generated by Gist. From all the results observed above MLH performed better than both BRE and LSH. MLH algorithm is online, efficient, and scales well to large code lengths and empirical results on different datasets suggest that MLH outperforms exiting methods. Conversion of real values to binary compact code by Minimal Loss Hashing and usage of randomized algorithms like Locality Sensitive Hashing helps in significantly improving the computational performance.

Scope of research is high as we can modify the algorithms mentioned above so that the efficiency in terms of time and space can be improved which in turn improves the efficiency of the Biometric identification and authentication.

References:

1. Piotr Indyk and Rajeev Motwani, Approximate nearest neighbours: towards removing the curse of dimensionality, STOC 1998, pp. 604-613.
2. Yunchao Gong, Svetlana Lazebnik, Albert Gordo, FlorentPerronnin, Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (12): 2916-2929 (2013).
3. Mohammad Norouzi, David J. Fleet, Minimal Loss Hashing for Compact Binary Codes, Proceedings of the 28th International Conference on Machine Learning (ICML-2011), pp. 353-360 (2011)
4. Information Theoretic Metric Learning Algorithm (ITML), http://www.cs.utexas.edu/users/pjain/itml, Accessed 4 May 2017.
5. Overfeat, https://github.com/sermanet/OverFeat, Accessed 4 May 2017.
6. Modeling the shape of the scene: a holistic representation of the spatial envelope,
    a. http://people.csail.mit.edu/torralba/code/spatialenvelope, Accessed on 4 May 2017.