International Journal of Sciences & Applied Research

International Journal of Sciences & Applied Research

www.ijsar.in

Using local binary pattern on high gradient of image for illumination invariant face recognition

Refath Ara Hossain*

Department of Computer Science and Engineering, Daffodil International University (DIU), Sobhanbagh, Dhanmondi, Dhaka, Bangladesh.

Correspondence Address: *Refath Ara Hossain, Department of Computer Science and Engineering, Daffodil International University (DIU), Sobhanbagh, Dhanmondi, Dhaka, Bangladesh.

Abstract

Local Binary Pattern (LBP) is a kind of discriminative texture descriptor for characterization of face patterns. However, the value of LBP operator is greatly changed under non-monotonic intensity transformations. In this paper, a novel illumination-invariant face recognition algorithm that applies LBP descriptor is proposed to overcome the performance degradation of LBP descriptor caused by varying illumination conditions. In our proposed method, the high gradient of image is first determined. The gradient of an image measures how it is changing. Using the image gradient the boundary of an image would be identified easily even if the illumination changes. Then, powerful LBP operator is applied only to the high gradient of image to derive a distinctive and robust representation for face patterns in images and the low gradient of images are discarded. We compared the recognition accuracy of the proposed algorithm with that of traditional PCA-based, Baysian-based, EBGM-based and raw LBP-based method on FERET face database. Another advantage of our proposed method is its computational simplicity, so it is very suitable for real-time manipulation.

Keywords: Face Recognition, Local Binary Pattern, Illumination Invariant Face Recognition, High Gradient Image

Introduction

Automated facial recognition involves the identification of an individual based on his or her facial geometry. Especially, face detection is an important part of face recognition as the first step of automatic face recognition. However, face detection is not straightforward because it has lots of variations of image appearance, such as pose variation (front, non-front), occlusion, image orientation, illuminating condition and facial expression. The motivation for this work comes from the observation that face images can be effectively described with the Local Binary Pattern (LBP) texture operator. This model for face description was first presented in papers (Ahonen et al, 2004) and (Ahonen et al, 2006). In Paper, (Ahonen et al, 2004) for the first time LBP was applied to face recognition, thus bringing a texture approach to face description.

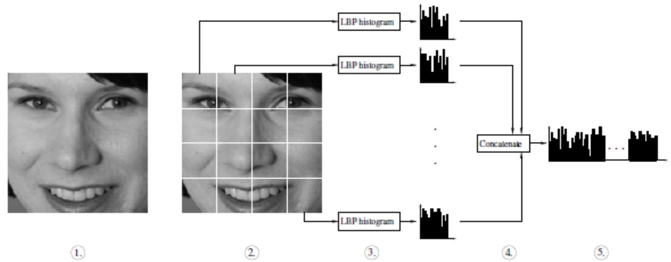


Figure 1: Face description with local binary patterns. 1. A registered face image is assumed. 2. The image is divided into local regions. 3. An LBP histogram is extracted independently from each region. 4. The histograms from regions are concatenated. 5. The concatenated histogram is used as descriptor for face recognition.

Generally speaking, current face recognition methods can be divided into two broad categories, i.e. holistic-based methods and local-based methods. Since local representation provides robustness to partial occlusion, local-based methods have dominated face recognition from the mid 1990s. For example, Pentland et al. (1994) extended the eigenface technique to a layered representation by combining eigenfaces and other eigenmodules, such as eigeneyes, eigennoses, and eigenmouths. A similar approach, named subpattern PCA (SpPCA), was studied by Chen and Zhu (2000). Local binary pattern (LBP), proposed by Ojala et al. (1996) was introduced to face recognition in 2004 (Ahonen et al, 2004). In LBP-based approaches, the face area was divided into a couple of small windows. LBP operators on each window were extracted and the weighted chi square statistic was adopted to compare LBP histograms. LBP operator achieves superior discriminative power on face images with expression, aging, pose variations as well as partial occlusion (Zhao and Pietikäinen, 2007) compared with other local features such as local PCA feature, Gabor feature, SIFT feature etc. By definition, LBP is invariant under any monotonic transformation of the pixel intensity. However, it is significantly affected by non-monotonic intensity transformations. Unfortunately, in contrast to the flat surfaces where texture images are usually captured, faces are not flat and non-monotonic intensity transformations occur in face images. Therefore, LBP may have problems to deal with non-monotonic illumination variations in face recognition.

Local Binary Patterns

The Original Local Binary Pattern

The original LBP operator, first introduced in paper (Ojala et al, 1996), is a powerful means of texture description. The operator assigns a label to every pixel of an image by thresholding the 3 x 3-neighborhood of each pixel with the center pixel value and considering the result as a binary number. Then, the histogram of the labels can be used as a texture descriptor. Figure 2 shows an illustration of the basic LBP operator.

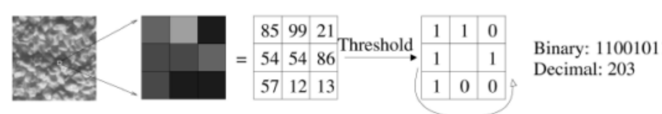


Figure 2: The basic LBP operator

Uniform and Rotation Invariant Patterns

To be able to deal with textures at different scales, the LBP operator was later extended to use neighborhoods of different sizes (Ojala et al, 2002). Defining the local neighborhood as a set of sampling points evenly spaced on a circle centered at the pixel to be labeled allows any radius and number of sampling points. Bilinear interpolation is used when a sampling point does not fall in the center of a pixel. In the following, the notation *(P, R)* will be used for pixel neighborhoods which mean P sampling points on a circle of radius of R. Figure 3 shows an example of circular neighborhoods.

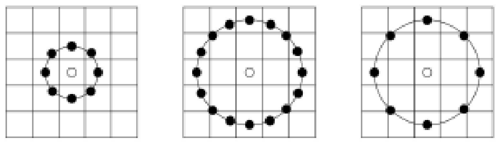


Figure 3: The circular (8,1), (16,2), and (8,2) neighborhoods. The pixel values are Bi-linearly interpolated whenever the sampling point is not in the center of a pixel.

Let us denote the value of the center pixel *(x, y)* by $g_c = I(x, y)$ and the gray values of the P sampling points by $g_1 = I(x_1, y_1)$, $g_2 = I(x_2, y_2)$,….., g_P, where the sampling points lie at coordinates $(x_p, y_p) = (x + R\cos(2\pi\, p/P),\ y - R\sin(2\pi\, p/P))$. Now the generic $LBP_{P,R}$ operator is defined as

$$LBP_{P,R} = \sum_{p=1} s(g_p - g_c)\, 2^{p-1}, \qquad (1)$$

where

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \qquad (2)$$

Another extension to the original operator uses so called uniform patterns (Ojala et al, 2002). For this, a uniformity measure of a pattern is used: U (.pattern.) is the number of bitwise transitions from 0 to 1, or vice versa, when the bit pattern is considered circular. A local binary pattern is called uniform if its uniformity measure is at most 2. For example, the patterns 00000000 ($U = 0$), 01110000 ($U = 2$) and 11001111 ($U = 2$) are uniform, whereas the patterns 11001001 ($U = 4$) and 01010011 ($U = 6$) are not. In uniform LBP mapping, there is a separate output label for each uniform pattern, and all the non-uniform patterns are assigned to a single label.

Image Gradient

An image gradient is a directional change in the intensity or color in an image. Mathematically, the gradient of a two-variable function (here the image intensity function) at each image point is a 2D vector with the components given by the derivatives in the horizontal and vertical directions. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction. To get the full range of direction, gradient images in the x and y directions are computed. The gradient of an image is given by the formula:

$$\nabla f = \frac{\partial f}{\partial x}\hat{x} + \frac{\partial f}{\partial y}\hat{y} \qquad (3)$$

where:

$\frac{\partial f}{\partial x}$ is the gradient in the x direction

$\frac{\partial f}{\partial y}$ is the gradient in the y direction.

The gradient direction can be calculated by the formula:

$$\theta = atan2\left(\frac{\partial f}{\partial y}, \frac{\partial f}{\partial x}\right) \qquad (4)$$

Let us represent an image by an array *A*, in which each element of the array corresponds to the gray level of an image. If the gray levels are in pixel counts, then the numbers might range from 0 to 255 for an eight-bit per pixel image. The gradient is the change in gray level with direction. This can be calculated by taking the difference in value of neighboring pixels. Let us construct a new array *B* that contains the values of the gradient from *A*. The horizontal gradient is formed by taking the differences between column values.

$$B(j, k) = A(j, k + 1) - A(j, k) \qquad (5)$$

This can be represented by a filter array as shown below:

-1	1

A problem with this filter is that the location of the gradient in the array *B* is shifted somewhat to the left. With an even number of pixels in the computation it is impossible to locate the result in the center of the cells used to produce it. It is therefore most common to use an odd number of cells. This can be accomplished by doing the calculation over cells that are separated by step.

$B\,(j, k) = A\,(j, k + 1) – A\,(j, k - 1)$ (6)

This can be represented by the array shown below:

-1	0	1

An example of a horizontal gradient calculation can be demonstrated on the image shown in Figure 4, which has strong vertical and horizontal structures. The result is shown in Figure 5.

Figure 4: Detail of a building at 256 gray levels.

Figure 5: Image produced by the horizontal gradient calculation.

Horizontal edges would be detected by calculating the vertical gradient. The equation for the separated vertical difference is

$B\,(j, k) = A\,(j +1, k) – A\,(j - 1, k)$ (7)

For an image in which the row coordinates are counted from the bottom edge upward, the corresponding filter array is

1
0
-1

An example of a vertical gradient calculation is shown in Figure 6.

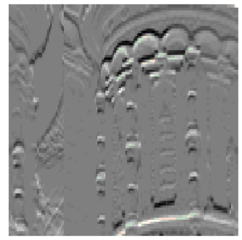


Figure 6: Edge produced by vertical gradient calculation.

Sobel Operator

Sobel filter is a simple approximation to the concept of gradient with smoothing. The 3x3 convolution mask is usually used to detect gradients in X and Y directions. The operator consists of a pair of 3×3 convolution kernels as shown in Figure 7. One kernel is simply the other rotated by 90°.

+1	+2	+1
0	0	0
-1	-2	-1

Gx

1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 7: Masks used by Sobel Operator

The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient (Matthews, 2002). The gradient magnitude is given by:

$|G| = \sqrt{Gx^2 + Gy^2}$ (8)

Typically, an approximate magnitude is computed using:

$|G| = |Gx| + |Gy|$ (9)

which is much faster to compute. The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$\theta = \arctan\,(Gy/Gx)$ (10)

Figure 8: Original image

Figure 9: Sobel

Face Recognition

Given a picture taken from a digital camera, we'd like to know if there is any person inside, where his/her face locates at, and who he/she is. Towards this goal, we generally separate the face recognition procedure into three steps: Face Detection, Feature Extraction, and Face Recognition (shown at Figure 10).

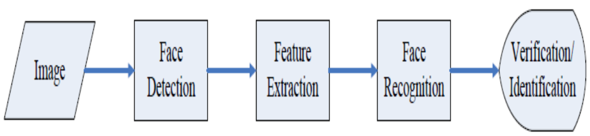


Figure 10: Configuration of a general face recognition structure

After formulizing the representation of each face, the last step is to recognize the identities of these faces. When an input face image comes in, we perform face detection and feature extraction, and compare its feature to each face class stored in the database. In figure 11, we show an example of how these three steps work on an input image.

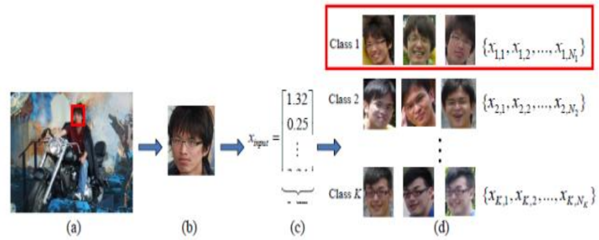


Figure 11: An example of how the three steps work on an input image. (a) The input image and the result of face detection (the red rectangle) (b) The extracted face patch (c) The feature vector after feature extraction (d) Comparing the input vector with the stored vectors in the database by classification techniques and determine the most probable class (the red rectangle). Here we express each face patch as a *d*-dimensional vector, the vector as the vector in the class, and as the number of faces stored in the class.

Chi Square Classification

Following Face Detection Scheme once the enhanced feature vectors are obtained for the training samples, some kind of faces and non-faces models are needed in order to complete training stage and to be used in classification stage. In a first stage more simple classifiers have been used in order to investigate its potential as face descriptor. The first one is Chi Square dissimilarly metric due to the fact that it is easy to implement and that it is a good technique for histogram comparison. After features extraction step, a mean model of the Enhanced Feature Vectors for faces and non-faces, respectively, is calculated: *Mface* and *Mnon-face*. Then, as a first approach to be able to compare a new input sample ***S*** with these models, Chi Square is proposed as a simple dissimilarly metric to compare this sample with both classes, faces and non-faces. The main idea consists of finding the minimum quadratic difference normalized between samples in the feature vector to be evaluated, ***S***, with samples of a given model features vector, ***M***:

$$x^2\,(S,M) = \sum_i \frac{(S_i - M_i)^2}{|S_i + M_i|} \qquad (11)$$

In this case, two models exist: faces (***Mface***) and non-faces (***Mnon-face***); and each model is calculated as the mean value of its training feature vectors:

$$M_{face_i} = \sum_{j=1}^{Numfaces} \frac{S_{face_i,j}}{Numfaces} \qquad (12)$$

$$M_{non-face_i} = \sum_{k=1}^{NumNonfaces} \frac{S_{non-face_i,k}}{NumNonfaces} \qquad (13)$$

where:

- ***M*** *face i* is the *i*-sample of the face model feature vector, ***M****face*.
- ***M*** *non-face i* is the *i*-sample of the non-face model feature vector, ***M****non-face*.
- ***S*** *face i,j* is the *i*-sample of face number *j* feature vector.
- ***S*** *face i,j* is the *i*-sample of non-face number *j* feature vector.
- ***i*** is the feature vector sample out of a total of 203.
- ***j*** is a sample of the face training set.
- ***k*** is a sample of the non-face training set.

NumFaces can be different from NumNonFaces.

Proposed Method

We propose an enhanced LBP feature-based method for face recognition under varying illumination conditions. In our proposed method, the high gradient of image is first determined. The gradient of an image measures how it is changing. It provides two pieces of information. The magnitude of the gradient tells us how quickly the image is changing, while the direction of the gradient tells us the direction in which the image is changing most rapidly. The image gradient is important in boundary detection because images often change most quickly at the boundary between objects.

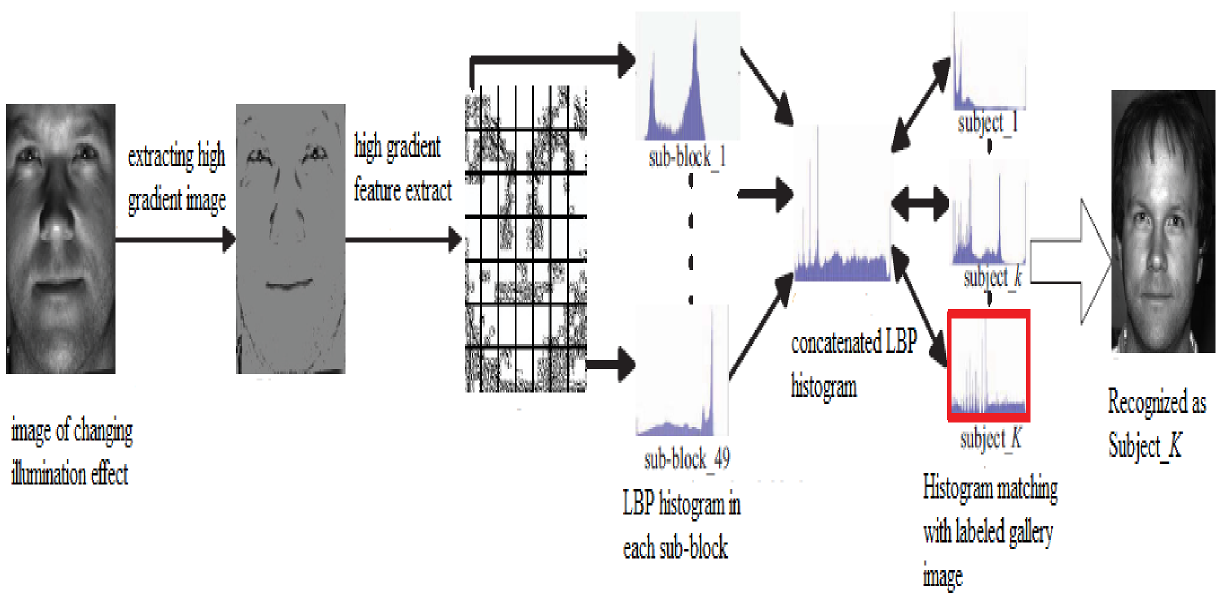


Figure 12: Framework of the proposed method for illumination invariant Face recognition.

Using the image gradient the boundary of an image would be identified easily even if the illumination changes. Then, powerful LBP operator is applied only to the high gradient of image to derive a distinctive and robust representation for face patterns in images and the low gradient of images are discarded. We significantly improve the recognition performance of the current raw LBP feature-based face recognition system. Another attracting feature of the proposed method is the computational simplicity, which make our method very suitable for real-time face recognition.

Experimental Results

The performance of proposed method is tested in the face recognition problem in accordance with the Colorado State University Face Identification Evaluation System (Beveridge et al, 2005) using images from the FERET (Phillips et al, 1998) database. Many approaches, such as Principle Component Analysis (PCA), Bayesian approach, Elastic Bunch Graph Match (EBGM) etc. are available for comparisons. After choosing the images, automatic face cropping and resizing have been done from the original image by utilizing the positions of two eyes, mouth and nose with equal height and width. Finally images are resized into 100 × 100 pixels. In the FERET database, the ground-truth data of eye, mouth and nose are provided. For the real-time system, we can use an existing eye detection technique that provides good detection accuracy (Niu et al, 2006). For automatic face cropping we have used the horizontal distance between two eyes (termed as *dx*) and vertical distance between mouth and nose (termed as *dy*). A distance of 1/3*dx* between the boundaries and both eyes has been maintained. The upper boundary has maintained a distance of 1/2*dy* from the eye position and the bottom boundary also has maintained a distance of 1/3*dy* from the mouth position. The whole cropping measurement is shown in figure 13. Since our proposed method is robust in illumination change, no attempt is made to remove illumination changes. Five example cropped images, one from each of those five groups is shown in figure 14. In our experimental setup, every image is partitioned into $M \times N$ subblocks. We used *fa* image set as gallery image and other four sets (*fb*, *fc*, *dupI* and *dupII*) as probe images. One image from probe set is compared using mentioned dissimilarity measure with all the images from gallery image set (*fa* set). The classification result is achieved through the nearest neighbor classification method. Table 1 shows recognition performance of proposed method along with other methods which ascertain the superiority of the proposed method.

Figure 13: Original face and cropped region of the face image.

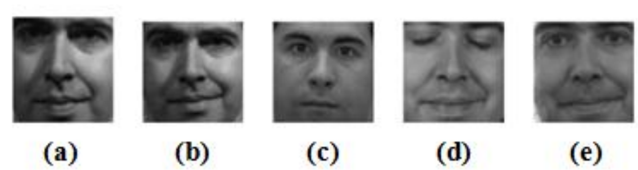
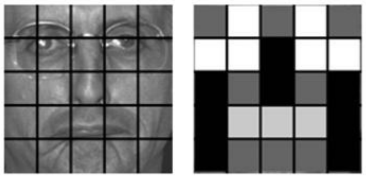


Figure 14: Example face image from FERET database. (a) Image from *fa* set, (b) Image from *fb* set, (c) Image from *fc* set, (d) Image from *dupI* set, (e) Image from *dupII* set.



(a) (b)

Figure 15: Weight assignment in different block (a) A facial image divided in 5×5 sub-regions, (b) The weights assigned for the weighted $\chi2$ dissimilarity measure. Black, dark gray, light gray and white indicates weight of 0.0, 0.5, 1.0 and 1.5 respectively.

Table 1: The recognition result of the proposed method and comparison algorithm in FERET database.

Methods	fb	fc	dupI	dupII
Proposed method, weighted	0.97	0.82	0.72	0.69
Proposed method, un-weighted	0.97	0.80	0.70	0.66
LBP	0.97	0.79	0.66	0.64
PCA	0.85	0.65	0.44	0.22
Bayesian	0.82	0.37	0.52	0.32
EBGM	0.90	0.42	0.46	0.24

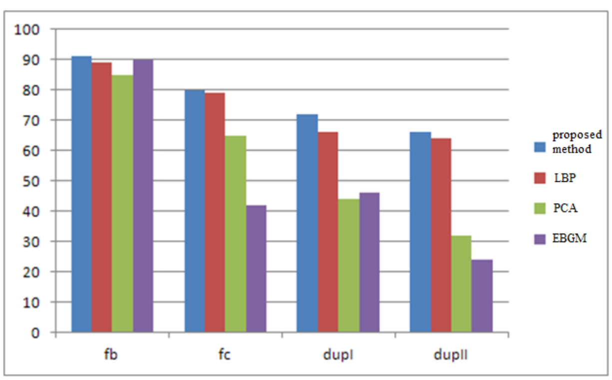


Figure 16: The recognition result of the proposed method and other algorithms in FERET database.

Our proposed method is quite robust with respect to dividing the whole image into different number of sub-regions as long as it divides the face into prominent facial components like eye, lip, nose etc. Recognition performances with different number of sub-regions are shown in Table 2. In that experiment we divide the image into 3x3, 5x5, 7x7 and 9x9 sub-regions or blocks. Experimental result shows that 7x7 block produces the optimal result for face recognition in FERET database.

As we are taking the high gradient pixel values from each block and discarding the low gradient pixels, it is also important to check that what percentage of high gradient of pixels should be chosen from each block that will produce optimum recognition rate. We have tested our proposed method using 80%, 85%, 90% and up to 95% of high gradient pixels. Experimental result shows that if 90% of high gradient pixels are selected then it will produce the best recognition rate comparing to 80% and 85% of high gradient pixels. But if higher than 90%, for example, 95% of high gradient pixels are taken, then the recognition rate will gradually reduces because 95% of high gradient pixels will definitely include the low gradient pixels which will start to change the recognition rate. The Recognition performances with different percentage of high gradient pixels taken are shown in Table 3.

Table 2: The recognition result using different sub-region size in FERET database.

Block Number	fb	fc	dupI	dupII
3x3 block	0.96	0.78	0.66	0.61
5x5 block	0.96	0.79	0.68	0.65
7x7 block	0.97	0.82	0.72	0.69
9x9 block	0.94	0.80	0.69	0.65

Table 3: The recognition result using different percentage of High Gradient taken in FERET database.

% of High Gradient Taken	fb	fc	dupI	dupII
80%	0.96	0.77	0.67	0.63
85%	0.96	0.78	0.68	0.67
90%	0.97	0.82	0.72	0.69
95%	0.94	0.80	0.70	0.66

When comparing different distance measures, the $\chi 2$ measure was found to be performed better than histogram intersection or log-likelihood distance. The recognition result of various dissimilarly measures are also shown in Table 4.

Table 4: The recognition result using different similarity measure in FERET database.

Similarly Measure	fb	fc	dupI	dupII
Histogram Intersection	0.95	0.78	0.66	0.66
Log-Likelihood	0.95	0.80	0.66	0.65
Chi-Square	0.97	0.82	0.72	0.69

In actual face recognition systems face detection is performed prior to face recognition. Automatic face detection method does not provide exact location of the face and it contains some small error in localization step. So proposed face recognition method should provide robust enough performance in presence of small face localization error. Our proposed face recognition method generated histograms over the local regions of the face image. So small changes of face position will only lead to changes in the labels which come from border of the local regions. But labels of inner part of the local region, which are major part of those regions, will not change. Therefore it is expected that the proposed method is robust in presence of small changes in face localization steps. This assumption is further verified with the figure 17.

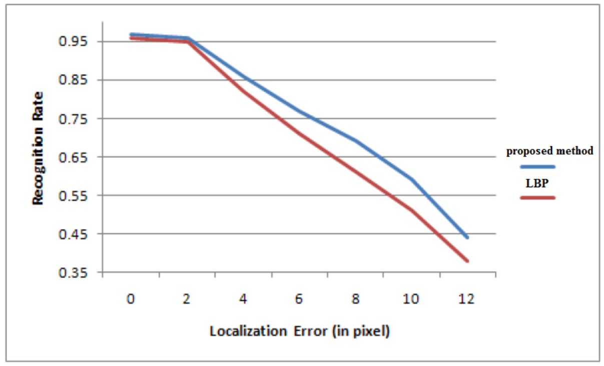


Figure 17: The recognition rate for the images of *fb* probe set with different face localization error.

Conclusion

This research paper is about designing and implementing an efficient face recognition system which performs well under structured surrounding. Extensions have also been made to make the performance of the system effective in relatively unorganized surrounding. Unorganized surrounding mostly refers to lightening. We significantly improve the recognition performance of the current raw LBP feature-based face recognition system. Another attracting feature of the proposed method is the computational simplicity, which make our method very suitable for real-time face recognition. Future work includes perform an extensive evaluation on more face datasets with severe illumination changes. It is important to note that each application related to face analysis has different requirements in the analysis process. This is the main reason why almost all algorithms and approaches for face analysis are application dependent and a standardization or generalization is very difficult, at least for the moment. Therefore, and due to the rapid evolution of technology that makes it possible, the recent trend is moving towards multimodal analysis combining multiple approaches to converge towards more accurate and satisfactory results.

References

[1] A. Pentland, B. Moghaddam, and T. Starner (1994): View-Based and Modular Eigenspaces for Face Recognition. In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 84–91.

[2] Ahonen T, Hadid A & Pietikäinen M (2004): Face recognition with local binary patterns. Proc. European Conference on Computer Vision (ECCV 2004), Prague, Czech Republic. Lecture Notes in Computer Science 3021, Springer: 469-481.

[3] Ahonen T, Hadid A & Pietikäinen M (2006): Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12): 2037-2041.

[4] J. Beveridge, D. Bolme, B. Draper, and M. Teixeira (2005): The CSU face identification evaluation system: Its purpose, features, and structure. Machine Vision and Applications, 16(2): 128–138.

[5] Matthews J. (2002): An introduction to edge detection: The sobel edge detector. Available http://www.generation5.org/content/2002/im01.asp.

[6] Ojala T, Pietikäinen M & Mäenpää T (2002b): Multi-resolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7): 971-987.

[7] Ojala T, Pietikäinen M, Harwood D (1996): A comparative study of texture measures with classification based on feature distributions. Pattern Recognition 29: 51–59.

[8] P. Phillips, H. Wechsler, J. Huang, and P. Rauss (1998): The FERET database and evaluation procedure for face recognition algorithms. Image and Vision Com- puting, 16(10): 295–306.

[9] Pietikäinen M. (2005): Image analysis with local binary patterns. Image Analysis, 115-118.

[10] S. C. Chen, and Y. L. Zhu (2000): Subpattern-Based Principle Component Analysis. Pattern Recognition, 37(5): 1081–1083.

[11] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao (2006): 2D cascaded adaboost for eye localization. International Conference on Pattern Recognition, 1216–1219.

[12] Zhao G. & Pietikäinen M. (2007): Dynamic texture recognition using volume local binary patterns. Dynamical Vision, 165-177.