Effective Eye Localization using Local Binary Patterns

Shylaja S S
Department of Information Science and Engineering
P E S Institute of Technology, Bangalore, Karnataka, INDIA

K N B Murthy
Department of Information Science and Engineering
P E S Institute of Technology, Bangalore, Karnataka, INDIA

S Natarajan
Department of Information Science and Engineering
P E S Institute of Technology, Bangalore, Karnataka, INDIA

Nitin Kumar
Department of Information Science and Engineering
P E S Institute of Technology, Bangalore, Karnataka, INDIA

Ruby Agarwal
Department of Information Science and Engineering
P E S Institute of Technology, Bangalore, Karnataka, INDIA

ABSTRACT
Eyes are one of the most salient features of the human face, playing a critical role in understanding a person's desires, needs and emotional states. They are also considered to be non-deformable objects appearing under various poses and lighting conditions. Therefore, efficient eye localization is a necessary step in many face-related applications like face recognition, face registration, face validation, gaze tracking, blink detection and red eye detection. In this paper, a probabilistic eye localization method based on local binary patterns (LBPs) is presented. Local binary pattern generates a binary code that describes the local texture pattern by normalizing the intensity values in a neighborhood. These patterns provide a simple but powerful spatial description of texture, and are robust to the noise typical to various illumination conditions and pose. LBPs are used for their higher accuracy rate and lower complexity. For a given close-up image, the centre of the iris of two eyes is located. The complete system has been tested on the standard databases and web-cam pictures of people under different light conditions. The accuracy has been nearly 98 % (±1 pixel shift).

General Terms
Pattern Recognition, Computer Vision and Image Processing, et. al.

Keywords
Eye Localization, Face Recognition, Gaze Tracking, Blink Detection, Local Binary Patterns.

1. INTRODUCTION
The availability of numerous commercial eye localization systems attest to the significant progress achieved in the research field. Despite these achievements, eye localization continues to be an active topic in computer vision research.

Thisis due to the fact that current systems perform well under relatively controlled environments but tend to suffer when variations in different factors (such as pose, illumination etc.) are present. Therefore, the goal of the ongoing research is to increase the robustness of the systems against different factors. Ideally, it is aimed to develop an eye localizer which mimics the remarkable capabilities of human visual perception**.**

Eyes are important facial features due to their relatively constant interocular distance. Locating the eyes eases the problem of locating other facial features such as nose and mouth required for recognition tasks. Eye localization is invaluable in determining the orientation of the face and also the gaze direction. To find eyes in an image we can either search for them directly in the entire image, or rely on the output of a face detector indicating that eyes are present in the image. The first case is the eye detection problem, while in the second, more investigated case, eyes only need to be localized within the bounding box supplied by a face detector. In this paper we concentrate on the latter problem of eye localization.

Rest of the paper is organized as follows: section 2 covers the related work, section 3 describes about the proposed methodology used in this paper, experimental results are illustrated in section 4 and section 5 focuses on conclusions & suggestions for further work.

2. RELATED WORK
Machine vision should be a direct method for getting information of surroundings. For explaining, sorting, abstracting and comprehending the images, some appropriate image processing methods must be brought forward. As a crucial branch of it, the human eyes location problem has attracted significant interests in the last decades. Eye detection is often the first step in numerous applications, such as video surveillance, human computer interface, face recognition, and image database management. Eyes represent most essential and important physical information of face. In fact, according to physiology, as if a basic point of position, eyes are closely connected to other parts of face, such as nose, ears, mouth and eyebrows.

Existing eye localization methods differ both in the features, models and data source (e.g. grayscale, color or infrared images) used. In terms of methods, simple filtering techniques, morphology operations and heuristics have become popular [1], [2] and [3]. Other approaches adapt the idea of Eigenface to the eye localization problem (Eigen-eye) [1] or are based on horizontal and vertical projections of the edge image, on wavelet decompositions, on discrete cosine transformed mean-subtracted face images or on applying neural networks on color images.

Authors argue that the point on the nose between the eyes is easier to find than the eyes themselves, and therefore derive the eye positions based on the detection of this point. Some other algorithms for eye localization require active infrared illumination. These algorithms are only applicable in controlled environments where it is possible to install an IR illumination rig.

The success of AdaBoost [4] in the face detection context has inspired many researchers to apply AdaBoost to eye localization as well. However, as the method is known for its good trade-off between detection rate and complexity and not for accurate face localization, post-processing is required to compensate for imprecise initial eye localization. Next to the AdaBoost approaches, other methods are based on wavelet features.

Hough Transform is proposed to find the circle shape of the eye irises and eyelids. However, the Hough Transform leads to heavy load of calculation. Bianchini and Sarti [5] refer to the eyes possess strong horizontal and vertical edges; the exploitation of gradient features is particularly suited to represent the image content. Therefore, a neural auto-associate can be trained to detect eye region by gradient features. In order to extract gradient features, Sobel filter is utilized.

More recently, Gabor wavelets techniques [6], where Gabor wavelet-based linear filters are used for eye corner detection and non-linear (Gaussian) filters are used for eye location. Moreover, the fractal model provides an excellent representation of the ruggedness of natural surfaces and has served as an image analysis tool for a variety of applications. The fractal dimension, which is the most commonly used fractal feature, has been applied to measure the irregularity of texture images and to describe these images. Lin ET Al. The lacunarity [7], which is a high-order fractal parameter, can assess the largeness of gaps or holes in images and describe the distribution of gaps within an image/region. For example, a method to estimate lacunarity can be based on a rectangular box instead of a square box. All these methods mentioned above closely depend on edge, shape, and model of skin color. There are many challenges to improve their robustness.

Also, there has been a lot of active research in this area, with algorithms based on Kalman filtering [8], but also more complex techniques, employing multiple phases of Bayesian classification, clustering and post processing or updated template matching [9]. Many of these approaches are very computationally intensive (requiring neural network trainings or large amounts of parallel processing), and many need color information.

Furthermore, in the initial phases of their processing, they make use of traditional edge detection techniques, like Canny or Laplace operators [10], which sometimes provide very bad results for human faces. All these attempts face challenges from issues like eye closure, eye occlusion, variability in scale and face orientation, and different lighting situations. A robust, accurate and non-intrusive eye detection and tracking mechanism remains a largely unresolved problem. Baluja ET. Al. [11]suggests a neural network based face detector with orientation normalization.

Approaches such as this require exhaustive training sets. Pitas ET. Al. [12] use thresholding in HSV color space for skin color extraction. However, this technique is sensitive to illumination changes and race. Feng ET. Al. [13] employ multi cues for eye detection on gray images using variance projection function. However, the variance projection function on a eye window is not very consistent. Pitas ET. Al.[12] adopt a similar approach using the vertical and horizontal reliefs for the detection of the eye pair requiring pose normalization.

P.Wang [14] compares fully automated eye localization and face recognition to the manually marked tests. The recognition results of those two tests are very close, e.g. 83.30% vs. 81.76% for experiment 1 and 97.04% vs. 96.38% for experiment 2. The performance of the automatic eye detection is validated using FRGC 1.0 database. The validation has an overall 94.5% eye detection rate, with the detected eyes very close to the manually provided eye positions.

ZhihengNiu et al. [15] introduced a framework of 2D cascaded AdaBoost for eye localization. This framework can efficiently deal with tough training set with vast and incompact positive and negative samples by two-direction bootstrap strategy. And the 2D cascaded classifier with cascading all the sub classifiers in localization period can also speed up the procedure and achieve high accuracy. The method is said to be very accurate, efficient and robust under usual condition but not under extreme conditions.

3. LOCALIZATION WITH LBP

In this paper, the proposed eye localizer produces the eye positions ($\hat{E}_L$, $\hat{E}_R$) for a given close-up image (I) and the corresponding face bounding box (f). where, f= (fx,fy,fw,fh) is the vector containing the main parameters of the face bounding box, namely the position of the center of the bounding box, width, and height, respectively and $\hat{E}_L = (\hat{E}_{Lx}, \hat{E}_{Ly})$, $\hat{E}_R = (\hat{E}_{Rx}, \hat{E}_{Ry})$ are the vectors indicating center position of the iris for left and right eyes, respectively.

It is assumed that the left eye is on the left side of the image ($\hat{E}_{Lx} < \hat{E}_{Rx}$) and that all sizes and positions are expressed in pixels. Also that both eyes are visible, which is in agreement with the focus on (near) frontal faces. The complete procedure for localizing the eyes in a given close-up image is primarily divided into three phases: training phase, learning phase and testing phase.

3.1 Training Phase

During the training phase, actual eye positions (E_{Li},E_{Ri}) and the detected face bounding boxes, $f_i$of each image in the training set are used to estimate prior eye locations ($\tilde{E}_L$, $\tilde{E}_R$). The actual eye positions (E_{Li},E_{Ri}) are obtained manually for every image in the training set.The estimated eye positions ($\tilde{E}_L$, $\tilde{E}_R$), called the prior estimation vector, is calculated as follows:

$$\begin{bmatrix} \tilde{E}_L \\ \tilde{E}_R \end{bmatrix} = PE\left[\begin{bmatrix} E_L \\ E_R \end{bmatrix} \middle| f\right]$$

which can be modelled as,

$$\begin{bmatrix} \widetilde{E}_L \\ \widetilde{E}_R \end{bmatrix} = f_w \begin{bmatrix} K_L \\ K_R \end{bmatrix} + \begin{bmatrix} f_x & f_y \\ f_x & f_y \end{bmatrix}$$

where, vector K = [K_L , K_R] is calculated as,

$$\begin{bmatrix} K_L \\ K_R \end{bmatrix} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{f_{wi}} \begin{bmatrix} E_{Li} \\ E_{Ri} \end{bmatrix} - \begin{bmatrix} f_{xi} & f_{yi} \\ f_{xi} & f_{yi} \end{bmatrix}$$

where, Nis the number of images in the training set.

The complete procedure for the eye localization is explained in the figure Fig. 3.1

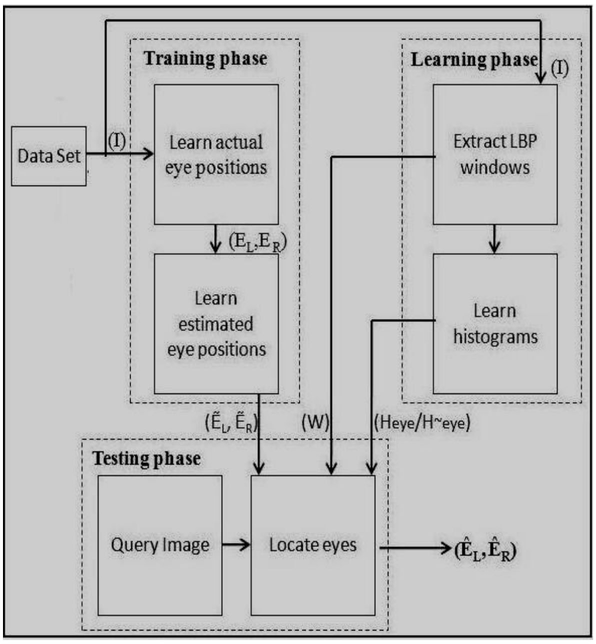


Fig. 3.1 Eye Localization

3.2 Learning Phase

During the learning phase, all images $I_i$in the training set are transformed and processed by the LBP (local binary pattern) extractor to extract LBP windows $\mathbf{W}_{x,y}$ (nXm). For a given window (nXm) of intensity values of the pixels around the point (x, y), the LBP extractor transforms them to corresponding LBP values and generates a LBP window $W_{x,y}$ (nXm). For position (x,y), on the actual eye position (E_L,E_R) and different non-eye positions around it, respective eye and non-eye LBP windows are generated. The LBP texture analysis operator is defined as an illumination invariant texture measure, derived from a general definition of texture in a local neighborhood

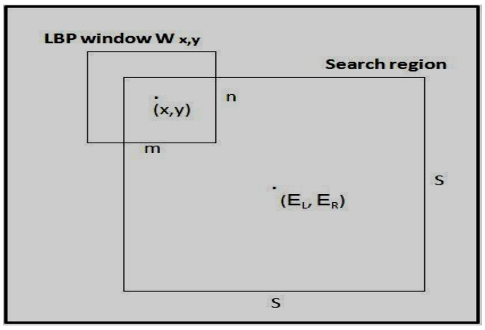


Fig. 3.2 LPB Window

The basic idea is to calculate a binary code that describes the local texture pattern built by thresholding a neighborhood by the intensity value of its center. The LBP operator labels the pixels of an image by thresholding a 3X3 neighborhood of each pixel with the center value and considering the results as a binary number.

Formally, given a pixel at (x_c,y_c), the resulting LBP can be expressed in the decimal form as

$$LBP(x_c, x_y) = \sum_{n=0}^{7} s(i_n - i_c)2^n \quad s(x) = \begin{cases} 1, if \quad x \geq 0 \\ 0, otherwise \end{cases}$$

where, n runs over the 8 neighbors of the central pixel, i_c and i_n are the gray-level valuesof the central pixel and the surrounding pixel, and s(x) is 1 if x > = 0 , and 0 otherwise.

For example, as shown in figure Fig. 3.3, the LBP value for the intensity value 100 is calculated by comparing the value 100 with its eight neighboring intensity points – 130, 120, 40, 42, 192, 14, 19 and 130. The intensity value is replaced by the LBP transformed value 201.

LBP values limit the intensity values of the pixels within the LBP window to the range of 0 to 255. It eliminates the illumination variations. The LBP windows, $W_{x,y}$ (nXm) obtained for both the eye and non-eye positions are used to generate the eye and non-eye histograms respectively.

Fig. 3.4 shows the LBP transformed images. Irrespective of the illumination conditions, the structure of the face is maintained as same in all the images.

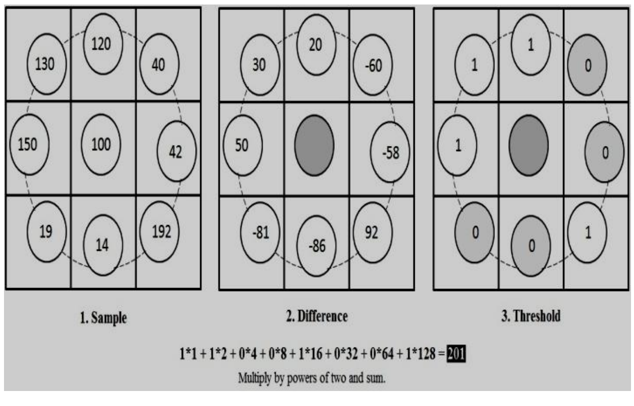


Fig. 3.3 LBP Calculation

Fig 3.4 LBP Transformed Images

A histogram (nXmX256) counts how often pattern occurs at each pixel position within the LBP window across all the images in the training data set. Here the histogram comprises of pixel positions (x, y) co-ordinates along the x and y axes; LBP values ranging from 0 to 255 along the z axis. The function f(x,y,z) represents the count value at (x,y) for LBP value z.

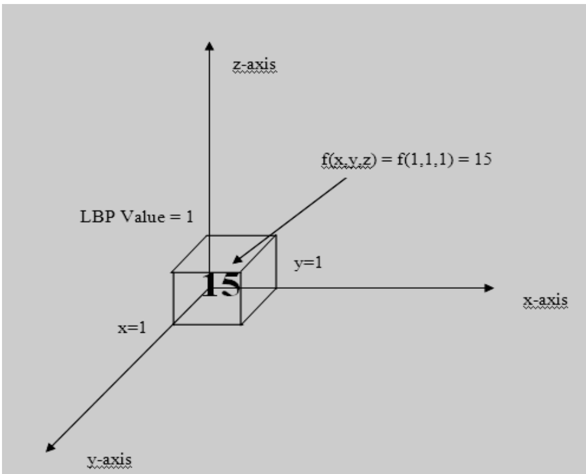


Fig 3.5 Histogram Representation

The histogram for LBP window at eye position is H_{eye} (nXmX256) and for non eye position is $H_{\sim eye}$ (nXmX256) where nXm is the size of the LBP window $W_{x,\ y}$ and 256 refers to the total number of possible LBP values(0 to 255). For one eye (left or right) for the facial images present in the training set, one Heye is created pertaining to the LBP window created for actual eye position. However, eight H~eye are created pertaining to eight LBP windows created for eight non-eye positions around the actual eye position.

The plots Fig 3.6 & Fig 3.7 given below, represent the total count of each LBP value ranging from 0-255 (plotted as 1-256) occurring at each pixel position within the LBP window. It was found that the sum of the count of all the LBP values within the LBP window for eye position equals the sum of the count of all the LBP values within the LBP window for non eye positions. However, the distribution of the LBP values within the LBP windows for both the eye and non eye positions was different, i.e., same LBP value occurs with different frequency at the same pixel position in the LBP windows for eye and non-eye regions. The differences in the distribution of LBP values help to distinguish between eye and non-eye regions.

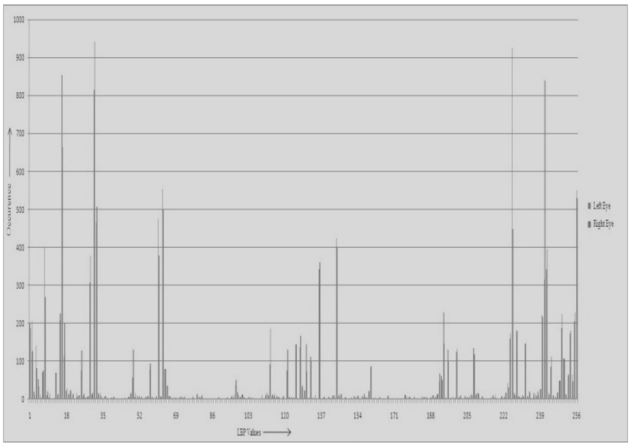


Fig.3.6. Occurrence vs. LBP values within the LBP window

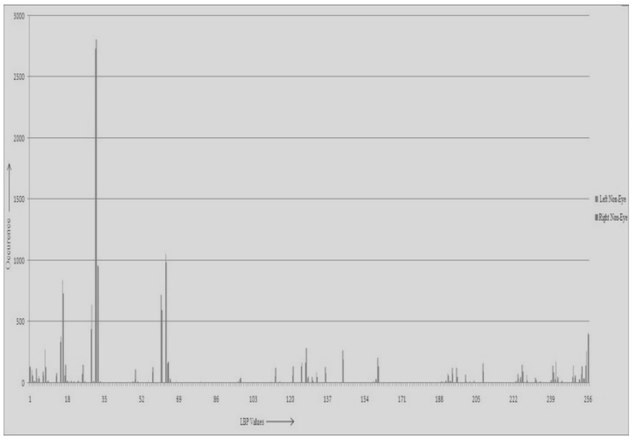


Fig. 3.7. Occurrence vs. LBP values within the LBP window for non-eye position

Fig 3.8 – Fig 3.11 represent the histogram pattern for one of the LBP values-31 for eye & non-eye position for both left and right eyes of the facial image. The pattern describes the count of LBP value 31 at every pixel position within the LBP window for all the images in the training set. The LBP window size (nXm) is 11X11 which is represented in X and Y axis. Z axis represents the count for the LBP value 31 at every point in the LBP window for all images in the training data set.

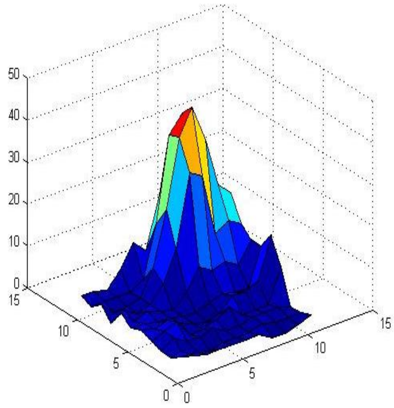
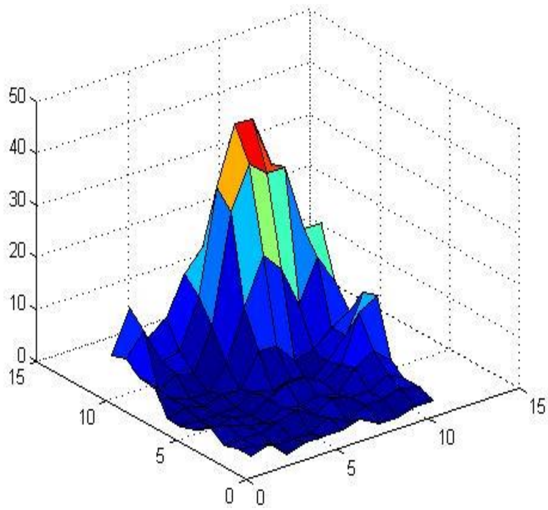
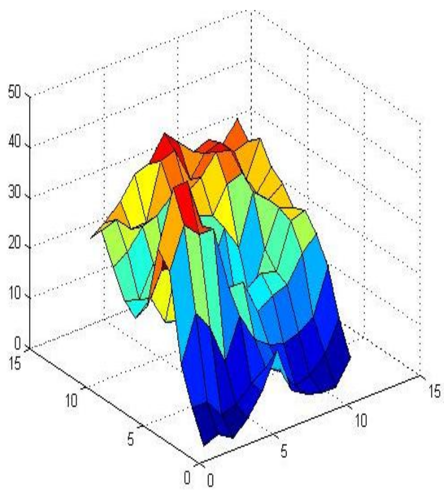
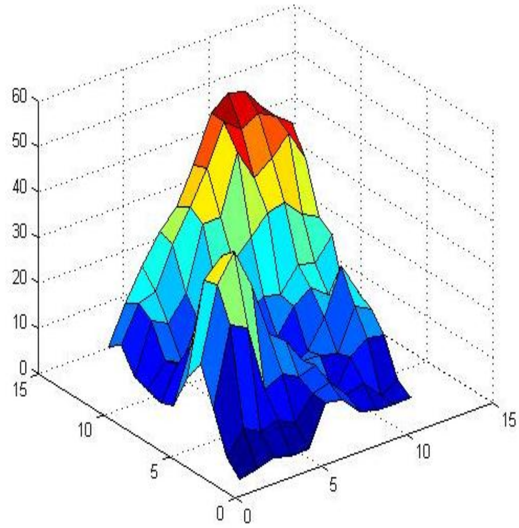


Fig. 3.9. Histogram Pattern for Left Non Eye Regio







3.3 Testing phase

During the testing phase, the query facial image is processed by the LBP extraction part and the localizer. Because the actual eye positions are not available, the transformation is now determined by the prior estimation vector ($\tilde{E}_L$, $\tilde{E}_R$). For both the eyes in the query image, a search window is created around the estimated eye positions. To determine the accurate centre of the iris within the search window, histograms H_{eye} and $H_{\sim eye}$ are used. For every pixel inside the search window, a eye texture analysis window of size nXm is created around it. LBP extractor transforms the pixel values within the eye texture analysis window to their respective LBP values and hence a same dimensional LBP transformed eye search matrix is obtained. Similarly, for the regions present outside the eye texture analysis window, non eye texture analysis windows of the same size (nXm) are created. LBP extractor is applied on these non eye texture analysis windows to generate the respective LBP transformed non eye search matrices. The LBP transformed eye search matrix is compared against all the LBP transformed non eye search matrices.

The comparison of LBP transformed eye search matrix against every LBP transformed non eye search matrix is done at pixel-by-pixel basis. For each pixel in the LBP transformed eye and non eye search matrices, same co-ordinate position (i, j) and the LBP transformed value at the respective co-ordinate position ($W_{x,\,y}(i,j)$) in the two matrices are fed to the eye and non eye histograms respectively. The histograms, H_{eye} and $H_{\sim eye}$ return the respective frequency of occurrence of the LBP value at the given co-ordinate for eye and non eye search matrices. A logarithmic ratio is taken for the two frequencies obtained from the histograms. This is repeated for all the pixels present in the search matrices. A summation of all such logarithmic values is calculated. This is called initial transformation value, $T_{initial}$.

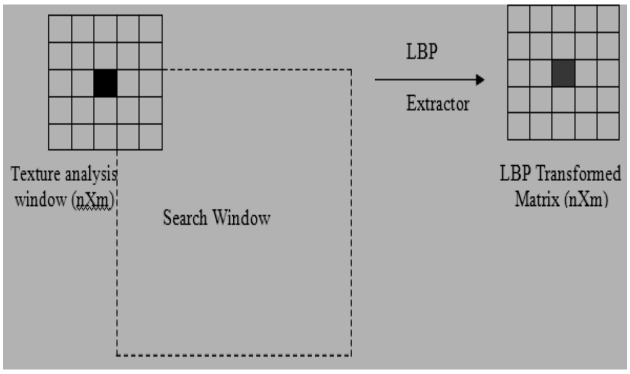


Fig. 3.12 Testing Phase

$$T_{initial} = \sum_{i,j=0}^{n,m} \log \frac{H_{eye}\left(i, j, W_{x,y}(i, j)\right)}{H_{\sim eye}\left(i, j, W_{x,y}(i, j)\right)}$$

Similar initial transformation values are obtained by comparing the same LBP transformed eye search matrix with different LBP transformed non eye search matrices. A summation of all the initial transformation values is calculated. This scalar is referred to as the final transformation value T_{final}. The final transformation value is attached to the pixel in the search window for which the texture analysis window was created.

$$T_{final} = \sum_{k=0}^{7} T_{initial_k}$$

Where, k runs from 0 to 7 for every different LBP transformed non eye search matrices and $T_{initial(k)}$ refers to the initial transformation value obtained by comparing LBP transformed eye search matrix with different LBP transformed non eye search matrices.Among all the pixels in the search window created around the prior estimation points ($\tilde{E}_L$, $\tilde{E}_R$), the pixel with the highest value for the final transformation value T_f is considered as the most precise eye position vector ($\hat{E}_L$, $\hat{E}_R$) for the query image.

4. EXPERIMENTAL RESULTS

For testing, a set of total 155 images was used. Experiments were conducted over 124 test database images and 31 web-cam images. The accuracy level attained was nearly 98% (± 1 pixel shift). The total number of images used in the training set was 103 and prior eye positions were calculated using the images in the training set. The above graphs indicate images vs. accuracy for 123 test database images and 31 webcam images.

The graphs 4.3 & 4.4 indicate the accuracy level obtained for each of the images tested. The test data set comprised of facial images obtained under various lighting conditions and facial images with and without the spectacles. The tables 4.1 & 4.2 below indicate the test results.

Table: 4.1 Test Results for Left Eye

Database Name	No of Images	Detection (precision)	False Alarm
Test database	124	105 (98.34%)	19
Web-cam	31	30 (98.10%)	1

Table: 4.2 Test Results for Right Eye

Database Name	No of Images	Detection (precision)	False Alarm
Test database	124	113 (98.42%)	11
Web-cam	31	29 (98.27%)	2

The test images below indicate both true and false alarms:

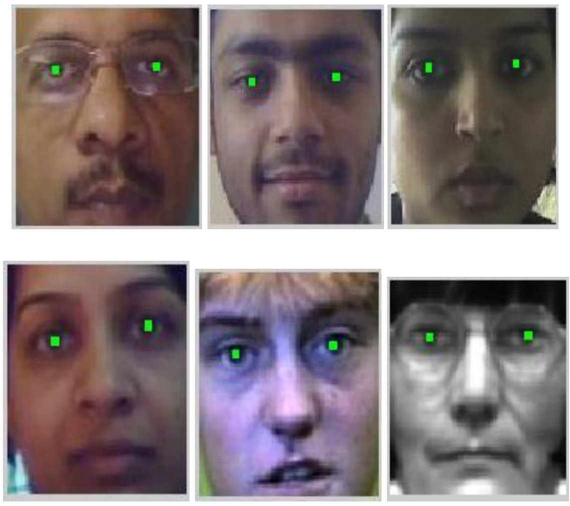


Fig 4.1 True Alarms

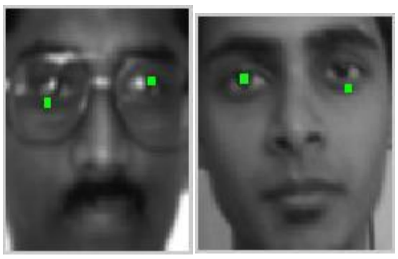


Fig 4.2 False Alarms

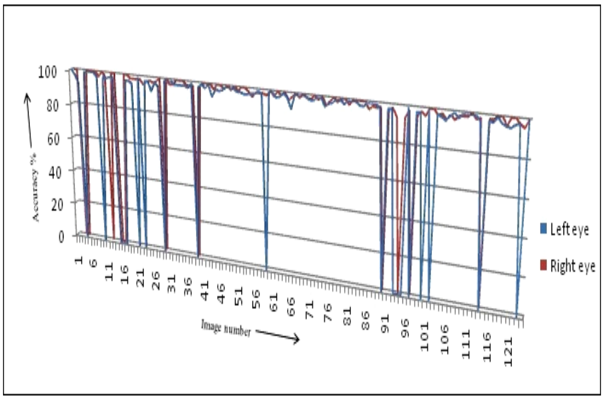


Fig. 4.3 Accuracy vs. Images for Test database

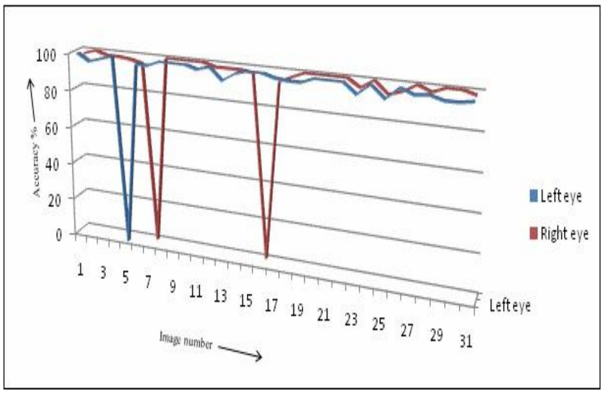


Fig.4.4 Accuracy vs. Images for Webcam Images

5. CONCLUSIONS AND FUTURE ENHANCEMENTS

In this paper we described a method for eye localization based on local binary patterns. The features used, namely LBPs, have a compact representation, are simple to compute and provide a good description of the spatial texture repartition. The primary contribution is a model which is able to achieve both high spatial accuracy and robustness by using spatial LBP histograms. It shows to be particularly robust for noise typical for the low and standard definition content, while being computationally efficient. The model can be extended to make use of multi-scale LBPs for higher accuracy.

6. ACKNOWLEDGMENTS

The authors extend their heartfelt sincere gratitude to PES Management for all the support extended in carrying out this research work.

7. REFERENCES

[1] Bart Kroon, Sander Maas, Abri Boughorbel, and Alan Hanjalic, **“**Eye localization in low and standard definition content with application to face matching”, *In the proceedings of international conference on Content-based Image and Video Retrieval*, pp.379-387, 2008.

[2] W. Huang, Q. Sun, C.-P. Lam, J.-K. Wu, “A robust approach to face and eyes detection from images with cluttered background”, *In the proceedings of the 14th International Conference on Pattern Recognition*, Vol. 1, pp. 110 -113, 1998.

[3] K. Peng, L. Chen, S. Ruan and G. Kukharev, A robust and efficient algorithm for eye detection on gray intensity face”, *In the proceedings of Pattern Recognition and Image Analysis*, pp. 302- 308. 2005.

[4] Y. Freund, “Boosting a weak learning algorithm by majority, Information and Computation”, *In the proceedings of Information and Computation*, Volume 121, Issue 2, pp. 256-285, 1995.

[5] M. Bianchini and L. Sarti, “An Eye Detection System Based on Neural Autoassociators”, *In the proceedings of Artificial Neural Networks in Pattern Recognition*, Lecture Notes in Computer Science, Volume 4087/2006, pp. 244-252, 2006.

[6] H. Kim, Lee, H.J., Kee, S.C. Seok, “A Fast Eye Localization Method For Face Recognition”, *In the 13th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 241- 245, 2004.

[7] E.J. Koh and P.K. Rhee, “Image Context-Driven Eye Location Using The Hybrid Network Of K-Means and RBF”, *In the proceedings of Advances in Natural Computation*, Lecture Notes in Computer Science, 2006, Volume 4222/2006, pp. 540-549, 2006.

[8] X. Liu, F. Xu, K. Fujimura, “Real-Time Eye Detection and Tracking for Driver Observation under Various Light Conditions”, *In the proceedings of Intelligent Vehicle Symposium*, pp. 334 -351, 2002.

[9] S. Kawato, N. Tetsutani, “Detection and Tracking for Gaze-camera Control”, *In the proceedings of 15th International Conference on Vision Interface*, pp. 1031- 1038 2002.

[10] A.Perez, M.L. Cordoba, A. Garcia, R. Mendez, M.L.Munoz, J.L.Pedraza, F. Sanchez, “A Precise Eye Detection and Tracking System”, *In the Journal of Computer Vision and Image Understanding*, Vol. 98, Issue 1, 2005.

[11] H. Rowley. **S.** Baluja. and T. Kanade, "Neural Network-Based Face Detection," *In the proceedings of lEEE International Conference on Computer vision and Pattern Recognition*, pp. 23 – 38, 1998.

[12] K. Sobattka and I. Pitas, "A Novel Method for Automatic Face Segmentation. Facial Feature Extraction and Tracking," *In the proceedings of Signal Processing: Image Communication*, Volume 12, Issue 3, pp. 263-281, 1998.

[13] Guo Can Feng and Pong C. Yuen. "Multi-cues eye detection on gray intensity image", *In the proceedings of Pattern Recognition*, Volume 34, Issue 5, pp.1033-1046, 2001.

[14] P.Wang, M. Green, Q. Ji, and J.Wayman., “Automatic Eye Detection and Its Validation”, In the proceedings

of *IEEE Conference on. Computer Vision and Pattern Recognition*, 2005.

[15] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao. "2D Cascaded AdaBoost for Eye Localization". In the *Proceedings. of the 18th International Conference on Pattern Recognition*, 2006.

[16] M.R. Everingham and A. Zisserman, "Regression and Classification Approaches to Eye Localization in Face Images", In the *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FG2006)*, pp. 441-448, 2006.