A Robust Biometric Image Texture Descripting Approach

J Bhattacharya
Central Mechanical Engineering Research Institute
Durgapur
West Bengal , India

G Sanyal
National Institute of Technology
Durgapur
West Bengal , India

S Majumder
CMERI
Durgapur
West Bengal , India

ABSTRACT

The huge demand in online transactions calls for a secure, safe and accurate authentication system .The biometric system such as face, iris, fingerprint, gait has already replaced the existing manual inspection process and surveillance systems in many disciplines. Amongst all these biometrics, face is more attractive as it provides information such as identity, expression, gender, ethnicity and age of an individual.Specially for surveillance purposes,the data acquisition for face is much more simpler and can be obtained without the subjects knowledge and cooperation (simply by installing camera in public areas) when compared to fingerprints and iris data accumulation.In this paper an edge texture feature using different weight mask coding is utilized for face recognition.Five different sign and difference operators named LSH_LC, LSH_GC,LDH_LC, LDH_GC and LSH_LGV are developed and used to code the image texture feature.Each image is decomposed into four subset images which are used to generate the texture features. A decision fusion technique is then used for feature classification. The main advantage of this approach lies in the fact that they are computationally inexpensive when compared to most texture descriptors.The feature descriptor is applied for face biometric recognition to demonstrate the effectiveness of each approach in extracting textural features.It can also be tested with medical images or other pattern recognition applications.The dataset used for training and testing have considerable variances in lighting,viewpoint and other factors so that the potential of the feature extractor, when subjected to any kind of variations, can be judged.

Keywords:

Texture analysis, LBP, Classification, Feature extraction, Face recognitionifx

1. INTRODUCTION

Feature extraction and classification problem is one of the main task undertaken by the computer vision researchers.Development of high quality image acquisition systems for perceiving maximum amount of environmental details electronically were carried out in tandem with the advancement of information extraction and interpretation techniques.In short,the exact simulation of the task carried out by the visual,neural and cognitive systems of the human body is applied while designing its automatized replica.Any problem statement is thus reduced to a feature extraction and classification problem differing only on the types of features chosen and classification classes from case to case.Features in the visual domain can be defined as geometric primitives such as point, line, arc segments or some form of derived entities such as color and texture for example.Although a lot of work has been carried out on interest point detectors,shape descriptors,edge and line detectors,yet blobs or region patch detection always remained an area of focus.This was mainly due to the fact that these features often require a secondary level of corroboration such as color and texture to make it invariant.This issue is applicable even for human beings where any object is identified more by its color and texture rather than its shape.It is obvious that tasks like pattern recognition,vision based detection of diseased crop,cancer cell detection in medical imaging and many more rely more on textural features.Even in the field of face recognition, techniques based on Geometric matching [1] and Template matching [2] used in the beginning eventually gave way to texture feature extractors for better accuracy and recognition rates.Also, learning models like Support Vector Machines,Neural Network,Hidden Markov Model,which directly used pixel intensity as input components initially [3],[4],started relying on texture features [5].The same applies for techniques like Principal Component Analysis (PCA) , Linear Discriminant Analysis (LDA) ,Independent Component Analysis (ICA), though initially applied over pixel intensities [6] ,[7],[8] have extended to other features [9],[10].Gray-Level Co-occurrence Matrices (GLCM's), Law's texture measures ,autocorrelation ,primitive length, edge frequency , wavelets such as Haar, Morlet, Daubechis and steerable pyramids (GLCM's) are some commonly used texture modeling techniques [13, 14, 15, 16].Some notable texture feature extractors specifically designed and applied to face recognition tasks are Wavelets [18],Energy-Entropy features [19],Gabor filters [20] and Local Binary Pattern(LBP) [22, 23].Among these the latter two are mostly used at present . The Gabor filter has the capability of extracting large amount of discriminating local features due to its ability to operate in selective scales and orientations thus capturing the spacial locality and quadrature phase relationships.Usually the gabor kernel $\psi_{\gamma,\upsilon}$ is constructed using five scales $\upsilon = 0, 1, 2, 3, 4$ and eight orientations $\gamma = 0, 1, ..7$ hence resulting in a very high dimensional representation thus taking a huge time for computation.

$$\psi_{\gamma,\upsilon}(z) = \frac{\| \kappa_{\gamma,\upsilon} \|^2}{\delta^2} e^{-\frac{\|\kappa_{\gamma,\upsilon}\|^2 \|z\|^2}{2\delta^2}} [e^{\imath \kappa_{\gamma,\upsilon} z} - e^{-\frac{\delta^2}{2}}] \quad (1)$$

where $\kappa_{\gamma,\upsilon} = \kappa_\upsilon e^{\imath\phi_\gamma}$, $\delta = 2\pi$, $\kappa_\upsilon = \frac{\kappa_{max}}{f^\upsilon}$, $f = \sqrt{2}$, $\kappa_{max} = \frac{\pi}{2}$,$\phi_\gamma = \frac{\pi\gamma}{8}$, $z = (r, c)$ and $\| \|$ denotes the norm operation.It can be seen from table 1 that calculating gabor features for a set of 300 images takes almost 500 seconds which is nearly 10 times that of LBP features.Hence,in many cases the LBP operator is preferred as it is seen that the performance of both are nearly equal.The LBP may also give better result in some cases.For example , the dataset used for experimentation in the proposed work (300 images from the PIE database [24]) gives the highest performance for the LBP operator as shown in table 1.The PIE database created by CMU originally consists of $41,368$ images of 68 people,considering 13 different poses, 43 different illumination conditions and 4 different expressions for each person.The 300 images used for experimental analysis in the present work are selected such that there are images of 15 people with different poses,illumination and expression.The paper introduces some variations of the Local Binary Pattern op-

erator by using the difference magnitude as well as considering the local and global differencing.These texture extractors when applied on a face database shows that some of the techniques perform better than the common LBP operator.The next step lies in feeding the extracted feature information to the classifiers for desired interpretation. A large number of approaches ranging from euclidean or mahalanobis distance classifier to soft computing,linear programming or statistical tools like neural network , Hidden Markov model (HMM) [25], Support Vector Machine (SVM)[11] ,adaboost classifier,bayesian classifier [26],[27] have been developed by different researchers to solve the classification problem.The present method uses a modified mahalanobis distance classifier to classify the faces.

The contribution of this work can be divided into two parts.Firstly it introduces a new texture descriptor which can be used for different types of object detection and recognition applications.Secondly the decision fusion framework described here can be used for combining any other features for a better recognition rate.The paper is organized in the following manner. Section 2 provides details about the different developed edge texture operators. Section 3 deals with the image classification step.Section 4 shows the results obtained using PIE face database. Finally, conclusion is presented in section 5.

Table 1. Recognition results and Computation time for different features using the PIE Face Database.

Methods	Test (in %)	Computation time(sec)
LBP	82.6	52
Gabor features	76	500
Daubechies 4 wavelet	74	22.34
Edge Frequency	51.33	58
Laws texture measure	54	41

The results are obtained using LBP,Gabor,Wavelet,Edge Frequency [16],Laws Texture [16] as features and euclidean distance as classifier.

2. EDGE TEXTURE OPERATORS

It is a familiar fact that image edges correspond to regions which encompass the substantial information of the scene such as depth & surface orientation alterations and texture & lighting variations.Thus it is obvious that features obtained from these regions provide maximum cue.The operators discussed in this work are hence applied on edge images.Each image is decomposed into the low pas filtered image and the horizontal,vertical and diagonal edge image.The edge features $Im_n^{edge\_comp}$ are obtained using a kirsch operator over the original image Im.Generally a kirsch operator gives the edge coefficients in eight directions using 8 consecutive 45 degree rotations of the kernel k_1 thus yielding $k_n, n = 1 : 8$.Here the horizontal $f_h(\tau)$,vertical $f_v(\tau)$ and diagonal edge $f_d(\tau)$ elements are calculated by combining the 2 horizontal $(Im_1^{edge\_comp}, Im_5^{edge\_comp})$ and 2 vertical components $(Im_3^{edge\_comp}, Im_7^{edge\_comp})$ and the four diagonal components $(Im_{2,4,6,8}^{edge\_comp})$. The low pass filtered image $f_{lpf}(\tau)$ is computed by subtracting the composite edge image Im_{edge} from the original image,where the composite edge image is obtained using the maximum gradient value among all directions.The computation steps and results of applying them on an image are further given in equation 9 and figure 1 respectively.

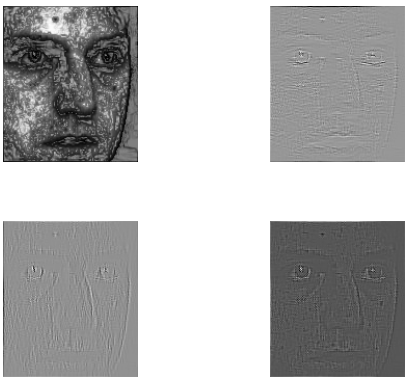

Fig. 1. The lowpass filtered image along with the horizontal,vertical and diagonal edge components.

$$k_1 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \quad (2)$$

$$Im_n^{edge\_comp} = Im \otimes k_n \quad (3)$$

$$Im(z)_{edge} = \max_{n=1:8} Im(z)_n^{edge\_comp} \quad (4)$$

$$f_{lpf}(\tau) = Im - Im_{edge} \quad (5)$$

$$f_h(\tau) = \sum_{n=1,5} Im_n^{edge\_comp} \quad (6)$$

$$f_v(\tau) = \sum_{n=3,7} Im_n^{edge\_comp} \quad (7)$$

$$f_d(\tau) = \sum_{n=2,4,6,8} Im_n^{edge\_comp} \quad (8)$$

$$f(\tau) = [f_{lpf}(\tau) \quad f_h(\tau) \quad f_v(\tau) \quad f_d(\tau)] \quad (9)$$

The edge texture feature referred as $\mathbf{x}_{j,i}$ is then extracted from each subset image of $f(\tau)$ individually . Here $i \in \{$ lowpass image (lp) ,horizontal edge components (h) , vertical edge components (v), diagonal edge components (d) $\}$. The size of j depends on the number of images used.The feature set for the j^{th} image can thus be written as $\{\mathbf{x}_{j,lp}, \mathbf{x}_{j,h}, \mathbf{x}_{j,v}, \mathbf{x}_{j,d}\}$. The different operators are further described in the subsequent subsections.

2.1 Local Sign Histogram with Local Center: LSH_LC

The sign code of a given pixel f_c is computed by finding its local difference with its neighbors f_p and generating the weight code c_s as shown in equation 10.Figure 2 shows the sign and difference value calculation for a particular pixel.

$$c_s = \sum_{p=0}^{7} \varpi_p . s_p$$

$$\text{where } s_p = \begin{cases} 1 & (f_p - f_c) > 0 \\ 0 & (f_p - f_c) < 0 \end{cases} \quad (10)$$

Four types of weight code ϖ namely power weight code,gaussian weight code,unity weight code and ordered weight code are used here.These are further elaborated in section 2.6.For obtaining a finer local detail the image is divided into blocks as shown in figure 3 and the frequency magnitude fm of all the b blocks are concatenated to generate the final code.The code frequency magnitude is calculated as the product

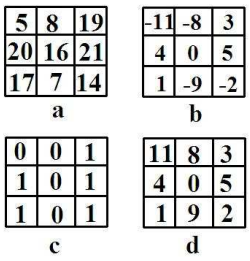

Fig. 2. (a) A 3×3 sample window. (b) local difference using the central pixel. (c) sign components for the window. (d) absolute difference components for the window.

Fig. 3. Face partitioned in equal sized blocks and code vector is obtained by concatenating the code of all the blocks.

of the number of occurrences f of a particular code with the code value c_s as shown in equation 11 .The frequency referred here is basically the histogram bin value where each bin corresponds to a particular code.

$$fm(c_s) = f(c_s) \times c_s \tag{11}$$

$$\mathbf{x}_{j,i} = [\mathbf{fm}_1 \quad \mathbf{fm}_2 \quad ..... \quad \mathbf{fm}_b] \tag{12}$$

where $\mathbf{fm}_n$ is the frequency magnitude vector of a particular block n.This operator is used with the power,gaussian and ordered weight codes.

2.2 Local Sign Histogram with Global Center: LSH_GC

In this operator the sign components are computed for each 3×3 window of a block using the global mean g_c of the image.Equation 10 is hence modified as equation 13.Similar to the LSH_LC operator, the LSH_GC operator is also used along with the power,gaussian and ordered weight codes.A 5×5 window can also be used in the same way.Specially for a gaussian weight code,5×5 windows are preferred over 3×3 window as a gentle slope accumulates greater variations thus giving a better result.

$$c_s = \sum_{p=0}^{7} \varpi_p . s_p$$

$$\text{where } s_p = \begin{cases} 1 & (f_p - g_c) > 0 \\ 0 & (f_p - g_c) < 0 \end{cases} \tag{13}$$

The feature set $\mathbf{x}_{j,i}$ is calculated in the same way as mentioned in equation 12.

2.3 Local Difference Histogram with Local Center: LDH_LC

Generally only the sign component is used for the LBP operator while the difference magnitude is discarded.[28] used the difference magnitude along with the sign component by coding it like the latter as shown in equation 14.The LDH_LC operator is tested with a different coding technique where the frequency is updated after calculating the difference components(as shown in figure 2d) for each 3×3 window of a block as shown in equation 15.

$$c_m = \sum_{p=0}^{7} 2^p . m_p$$

$$\text{where } m_p = \begin{cases} 1 & \|f_p - f_c\| - g_c > 0 \\ 0 & \|f_p - f_c\| - g_c < 0 \end{cases} \tag{14}$$

$$H(d_p) = H(d_p) + 1, \forall p = 0, ..., 7$$
$$\text{where } d_p = \|f_p - f_c\| \tag{15}$$

$H(d_p)$ gives the histogram of d_p.As the value of d_p ranges from $0 - 255$,its frequency is generated directly without using any weight code.For generalization it is considered that it uses the unity weight code(which implies no weight multiplication and no summation).

$$\mathbf{x}_{j,i} = [H(d_p)_1 \quad H(d_p)_2 \quad ..... \quad H(d_p)_b] \tag{16}$$

2.4 Local Difference Histogram with Global Center: LDH_GC

In this operator the difference components are computed for each 3×3 window of a block using the global mean g_c of the image.The frequency is then updated as shown in equation 17.Using the global mean while calculating the difference components is much less error prone than the local center case as the latter is highly sensitive to noise.

$$H(d_p) = H(d_p) + 1, \forall p = 0, ..., 7$$
$$\text{where } d_p = \|f_p - g_c\| \tag{17}$$

As in LDH_LC,LDH_GC also uses the unity weight code. The feature set $\mathbf{x}_{j,i}$ is calculated in the same way as mentioned in equation 16.

2.5 Local Sign Histogram with Local & Global Variation: LSH_LGV

Noise sensitivity can further be reduced if relative differences are considered.In LSH_LGV two difference components are calculated as shown in equation 18.First the local difference d_p is calculated and then the difference m between the local center f_c and the global mean g_c is calculated.The sign code is then calculated using the difference between these two values as shown in equation 19.

$$d_p = \|f_p - f_c\|$$
$$m = \|g_c - f_c\| \tag{18}$$

$$c_s = \sum_{p=0}^{7} \varpi_p . d$$

$$\text{where } d = \begin{cases} 1 & d_p > m \\ 0 & d_p < m \end{cases} \tag{19}$$

This operator is used with the power,gaussian and ordered weight codes. The feature set $\mathbf{x}_{j,i}$ is calculated in the same way as mentioned in equation 12. Figure 2.5 gives an illustration of the

amount of information depicted by the sign and difference components using local and global center differencing and the sign components using local and global variations.

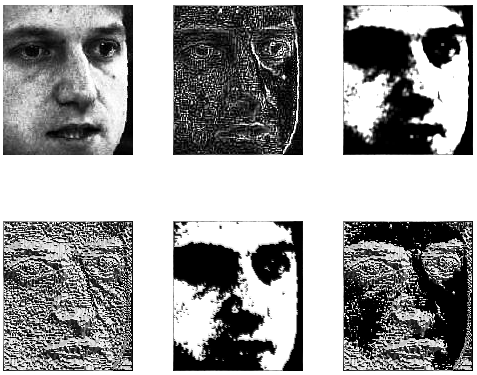

Fig. 4. The top row shows the original image , difference components using local & global differencing from left to right. The bottom row depicts the sign components using local & global differencing and local global variation from left to right.

2.6 Weight Codes

As mentioned previously four different types of weight codes are used as shown in figure 5.In case of a power weight mask,the mask coefficients are generated using the power of real numbers r as $\varpi = r^p$,where $p = 0, 1, ..7$.The binary $(r = 2)$ code commonly used in LBP is an example of the power weight code. For ordered weight codes the pixel position $1 - 8$ can be weighted using natural numbers $1 - 8$,even numbers $2 - 16$,odd numbers $1 - 15$,fibonacci series $1, 1, 2, 3, .., 21$ or prime numbers $1, 2, 3, ..17$ while the gaussian weight codes uses a gaussian kernel as weights.The size of the feature vector also depends on the type of code used.For example,for a power code of $r = 1.5$ the feature size of each block is 50 .This size is actually obtained from the summation of the kernel (shown in figure 5) used.Thus the total size for an image divided into 16 blocks gives a size of 800 for each feature (lp, h, v, d). The performance variation on the basis of the *r value* for power kernels,*s value* for gaussian masks and *type of numbering* for ordered weights are later discussed in section 4.Five variations are discussed for each type as shown in table 2.

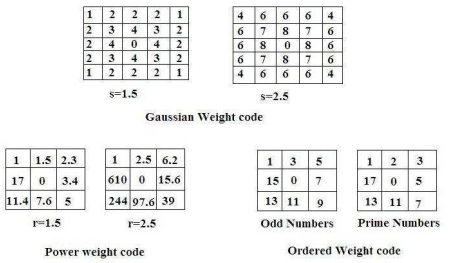

Fig. 5. Examples of some weight code kernels for gaussian (standard deviation s=1.5 and 2.5),power weights (r=1.5 and 2.5) and ordered weights.

Table 2. Variations of each type of weight code.

	Power Weight (r)	Gaussian Weight (s)	Ordered Weight
1	1.5	1.5	Odd numbers (1:15)
2	1.75	1.75	Even numbers(2:16)
3	2	2	Natural numbers (1:8)
4	2.25	2.25	Prime numbers (1:17)
5	2.5	2.5	Fibonacci (1:21)

3. IMAGE CLASSIFICATION

The feature descriptor **x** is further transformed in the eigen space using equations 21 to 24.

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_{1,lp} & \mathbf{x}_{1,h} & \mathbf{x}_{1,v} & \mathbf{x}_{1,d} \\ \mathbf{x}_{2,lp} & \mathbf{x}_{2,h} & \mathbf{x}_{2,v} & \mathbf{x}_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}_{m,lp} & \mathbf{x}_{m,h} & \mathbf{x}_{m,v} & \mathbf{x}_{m,d} \end{pmatrix} \tag{20}$$

Here, each element is referred as $\mathbf{x}_{j,i}$ i.e. the i^{th} feature set of the j^{th} image .An entire row $\mathbf{x}_{j,:}$ represents all the features of the j^{th} image $\{\mathbf{x}_{j,lp}, \mathbf{x}_{j,h}, \mathbf{x}_{j,v}, \mathbf{x}_{j,d}\}$ whereas an entire column $\mathbf{x}_{:,i}$ denotes the i^{th} feature of all the m images.Each feature $\mathbf{x}_{:,i}$ here is transformed to the eigen space individually and their dissimilarity σ_i is computed.These are then fused as shown in equation 26 to determine the fused dissimilarity $sigma_{comb}$. The mean vector of each feature i is calculated using equation 21.

$$\mu_i = \sum_{j=1}^{m} \mathbf{x}_{j,i} \tag{21}$$

The covariance $\mathbf{C}_i$ of the feature i in the training set is calculated as shown in equation 22

$$\mathbf{C}_i = \frac{1}{m} \varphi_i^T \times \varphi_i \tag{22}$$

where φ_i is the mean adjusted data for feature i .The feature is then transformed to the eigen space using the eigen vectors ω_i of the covariance matrix $\mathbf{C}_i$ as shown in equation 23.It should be noted that the subscript i here on is used to refer a particular feature set generated using all the images.Hence the following computations is to be carried out for all the four features before determining the combined dissimilarity vector.

$$\Omega_i = \omega_i^T . \varphi_i^T \tag{23}$$

For any test image the transformed vector in the eigen space is calculated from the feature $\hat{\mathbf{x}}_{k,i}$ using equation 24.

$$\widehat{\Omega}_{k,i} = \omega_i^T . (\widehat{\varphi}_{k,i})^T \tag{24}$$

where

$$\widehat{\varphi}_{k,i} = \hat{\mathbf{x}}_{k,i} - \mu_i \tag{25}$$

The dissimilarity vector σ_i $(\sigma_i = \{\sigma_{1,i}, \sigma_{2,i}, ..., \sigma_{m,i}\})$ is calculated by finding the mahalanobis distance between each training image projection component vector $\Omega_{j,i}$ and test image projection component vector $\widehat{\Omega}_{k,i}$. The combined dissimilarity σ_{comb} is then obtained using equation 26 and the best match is given by the image producing the least dissimilarity.

$$\sigma_{comb} = (\sum \frac{1}{\sigma_i})^{-1} \tag{26}$$

It can be proved that using this multimodal combination ,the dissimilarity σ_{comb} obtained by combining all the variables is less than the individual dissimilarity σ_i for each i. This can be proved easily using proof by contradiction.It is assumed that

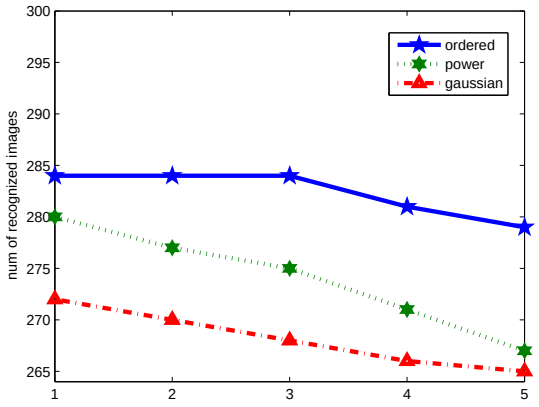

Fig. 6. The graph represents the performance of the weight codes with the different parameter types.The parameter selection is carried out using the LSH_GC operator over the lowpass filtered image.

$\sigma_{comb} > \sigma_1$. Thus

$$\frac{1}{\frac{1}{\sigma_1}+\frac{1}{\sigma_2}+\cdots+\frac{1}{\sigma_n}} > \sigma_1$$
$$\Leftrightarrow \frac{1}{\sigma_1} > \frac{1}{\sigma_1}+\frac{1}{\sigma_2}+\cdots+\frac{1}{\sigma_n}$$

The above identity is wrong for all values of σ_1 as σ_1 is a square root of a squared term and hence is always positive.Thus the above assumption is wrong and it can be stated that $\sigma_{comb} < \sigma_1$. σ_{comb} is similarly less than σ_2, σ_3..... Hence it is seen that σ_{comb} is less than the dissimilarities of all the variables and also decreases with the increase in the number of variables.The basic aim of using a multiple feature set is to reduce the false alarms or wrong identification obtained while depending on a single feature.A multi dimensional feature set can extract more salient information thus solving the problems of viewpoint,scale,illumination and other changes of the scene to some extent.However, feature fusion often hampers the overall performance in cases of samples which gives a good detection rate for one feature but a poor one for another.The above approach serves the advantage of using a multimodal feature set safely avoiding the general drawback as explained above.

4. EXPERIMENTAL RESULTS AND DISCUSSION

The performance analysis of the operators is carried out using the PIE face database with a set of 300 images (15 different individuals with 20 images each).For all the cases 10 out of the 20 images were used for training.The face images were normalized to a dimension of 150 x 130 pixels before use.The performance variation among the five types of parameters for each weight code (as discussed in table 2) is provided in figure 6.The LSH_GC operator is used for the results shown in tables 4 and 5 and figures 6 and 7. It is seen that for the power weights and gaussian weights the performance decreases when the $rvalue$ or standard deviation (as applicable) is increased.The optimal performance is attained at $r = 1.5$ and $s = 1.5$ for the power and gaussian weight codes respectively while the the odd and even ordered weight codes give a more or less similar performance.Using these optimal parameters for each weight code the different operators are evaluated as shown in table 3. It is observed that all the operators gives a more or less similar performance,however the LSH_GC gives the maximum recognition rate for the current dataset.From table 3 and figure 6 it can be inferred that the performance of

Table 3. Performance Comparison of the Operators (in%).

	Power	Unity	Gaussian	Ordered
LSH_LC	85.33		80.67	86.67
LSH_GC	86.67		82.67	89.33
LDH_LC		84		
LDH_GC		86.67		
LSH_LGV	85.33		80	88

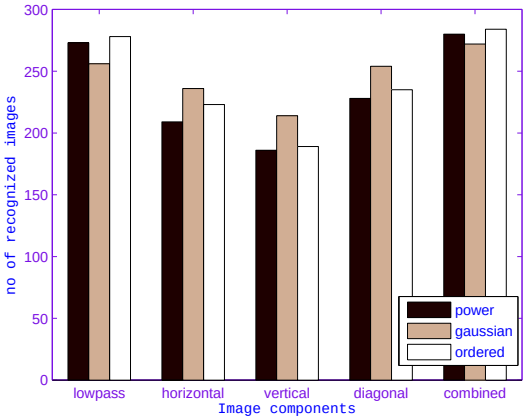

Fig. 7. The graph depicts that the three edge components give a higher recognition rate with gaussian weights than the ordered or power weights.

the gaussian weights is slightly lower than the power and ordered weights.Figure 7 further shows the recognition rate for each weight code obtained using the individual image components as well as the combined rate.A few conclusions are obtained from this result.Initially,it is noticed that the recognition rate from the lowpass image is less for the gaussian weight code when compared to the other two,hence resulting in the overall performance decrease.However the performance of the other edge components are slightly better than those obtained from the power and ordered weight codes.Thus combining a power/order coded lowpass image with gaussian coded edge components may result in an effective performance hike (seen in table 4).It can also be confirmed that the multimodal distance metrics gives a satisfactory result as it is not influenced by the low performance of particular features (for example the vertical component gives a low recognition for all the weight codes,yet its removal lowers the combined performance by 1.8% approximately) and increases the overall performance considerably (as distinctly noticed in case of gaussian weight code).

Table 4. Performance Comparison of Different Weight Combinations (in%).

	$G1.5\sigma$	$G1.75\sigma$	$G2\sigma$
Ordered natural no.	92.67	92.67	90.67
Ordered odd no.	93.33	92.67	90.67
Ordered even no.	92.67	92.67	90.67
Ordered prime no.	93.33	92.67	90.67
Ordered fibonacci no.	92	91.33	90
Power r=1.5	92.67	90	90

It is seen that the performance of the gaussian weights increase significantly on using the frequency instead of the frequency magnitude.Thus when the combination shown in table 4 is applied on the modified gaussian weight descriptor the total recognition rate reaches 95.33% as shown in table 5.However the

power and ordered codes work better with the frequency magnitude descriptor.Also,there is no improvement while combining the different components of the power and ordered codes.

Table 5. Performance Comparison with modified gaussian (in%).

	$G1.5\sigma$	$G1.75\sigma$	$G2\sigma$
Frequency magnitude	81.33	80	77.33
Frequency	89.33	89.33	88.67
Freq mag with Ordered natural no	92.67	92.67	90.67
Freq with Ordered natural no.	95.33	93.33	92.67

5. CONCLUSION

The experimental results verify that all the operators give a fairly high performance in classifying the images reliably with negligible false alarm or wrong identification with a high accuracy and low processing time.The feature set calculation time for 300 images is approximately 90 sec which is much less when compared to the statistical textural descriptors .The LSH_GC operator gives the best performance for the current dataset.It is also seen that the gaussian weights perform better on the edge components while the power/ordered weights give a better recognition rate for the lowpass filtered image.Combining the gaussian and power/ordered weights give a fairly high recognition rate (95.33) when compared to the LBP operator which gives a recognition rate of 82.6 for grayscale intensity images and 84.67 when used on edge images used for the present work.On a detailed observation,it was found that the main difference in these performances occur in cases of expression variations,especially eye region.The present technique is more expression invariant than the traditional LBP technique.The present code is tested on an Intel Pentium 4 machine, with CPU 3GHz and Memory 1GB using Matlab 7.0.As the face database used for testing has huge variations in pose,expression,orientations and lighting conditions;it can be inferred that the proposed technique is resilient to these variations and can give a robust performance in any kind of environment.The technique can be applied to different kind of detection and recognition tasks ranging from pattern recognition like face or gesture to medical imaging applications. Work is in progress to apply the feature extractor for medical imaging operations.

6. REFERENCES

[1] S. Tamura, H. Kawa, and H. Mitsumoto, "Male/Female identification from 8x6 very low resolution face images by neural network," *Pattern Recognition*, vol. 29, pp. 331-335, 1996.

[2] R. Bruneli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1042-1052, 1993.

[3] N. Jamil, S. Iqbal, N. Iqbal, "Face Recognition Using Neural Networks," *IEEE Proc. of INMIC technology for the 21st century*, pp. 277-281, 2001.

[4] P. J. Phillips, "Support vector machines applied to face recognition," *Advances in Neural Information Processing Systems 11*, MIT Press, 1999.

[5] Bhaskar Gupta , Sushant Gupta , Arun Kumar Tiwari,Face Detection Using Gabor Feature Extraction and Artificial Neural Network

[6] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal Cognitive Neuroscience*, vol. 3, pp. 71-86, 1991.

[7] Martinez, A.M., Kak, A.C., "PCA versus LDA," *IEEE Transactions Pattern Analysis and Machine Intellgence* Vol. 23 (2), pp. 228-233, 2001.

[8] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, Face Recognition by Independent Component Analysis, *IEEE Transaction on Neural Networks*, Vol 13, pp. 1450-1464, 2002.

[9] Xiaoyang Tan and Bill Triggs,Fusing Gabor and LBP Feature Sets for Kernel-Based Face Recognition,Springer Verlag,2007

[10] YUCHUN FANG, ZHAN WANG,IMPROVING LBP FEATURES FOR GENDER CLASSIFICATION,Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong

[11] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, Vol.2, pp. 121-167, 1998.

[12] Guoqin Cui, Wen Gao, Feng Jiao, and Shiguang Shan, "Face Recognition Based on Support Vector Method," *Asian Conference on Computer Vision*, January 2002.

[13] M. J. Nassiri and A. Vafaei and A. Monadjemi and Pwaset, " Texture Feature Extraction Using Slant-Hadamard Transform",World Academy Of Science, Engineering And Technology, 2006.

[14] Yuxin Liu and Yanda Li, " Image Feature Extraction And Segmentation Using Fractal Dimension",IEEE Conference on Information, Communications and Signal Processing, 1997.

[15] Hua Yuan and Xiao-Ping Zhang and Ling Guan, " A Statistical Approach For Image Feature Extraction In The Wavelet Domain",IEEE Conference on Electrical and Computer Engineering, 2003.

[16] Mona Sharma,Markos Markou,Sameer Singh, " Evaluation of Texture Methods for Image Analysis",Pattern Recognition Letters.

[17] Cootes, T.F., Edwards, G.J., Taylor, C.J., "Active Appearance Models," Proc. European Conf. on Computer Vision, Vol. 2, pp. 484-498, Springers, 1998.

[18] Yi Liu,Tao Sun , Huang Yang , Yongmi Yang , Xinhong Zhou ,Wavelet-Based Face Recognition Method by Using Support Vector Machine,Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference, 2009

[19] Cunjian.chen Jiashu.zhang," Wavelet Energy Entropy as a New Feature Extractor for Face Recognition", Fourth International Conference on Image and Graphics,IEEE ,2007.

[20] Chengjun Liu, Harry Wechsler, "A Gabor Feature Classifier for Face Recognition," *Proceedings of ICCV*, pp. 270-275, 2001.

[21] T. Ojala, M. Pietikinen, T. Menp, "Multiresolution grayscale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971-987, 2002.

[22] T. Ahonen, A. Hadid, and M. Pietikinen, "Face Recognition with Local Binary Patterns," *Proc. Eighth European Conf. Computer Vision*, pp. 469-481, 2004.

[23] T. Ojala, M. Pietikinen, T. Menp, "Multiresolution grayscale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971-987, 2002.

[24] CMU,PIE Face Database,$http : //www.ri.cmu.edu/research_project_detail.html$

[25] F. Samaria and F. Fallside, "Face identification and feature extraction using hidden markov models," *Image Processing: Theory and Application*, Elsevier, 1993.

[26] S. Park and J. K. Aggarwal, "A Hierarchical Bayesian Network For Event Recognition Of Human Actions And Interactions" Multimedia System, 2004.

[27] S. Avidan, "Support Vector Tracking," In IEEE Conference On Computer Vision And Pattern Recognition (Cvpr), 2001.

[28] Zhenhua Guo, Lei Zhang, and David Zhang, " A Completed Modeling of Local Binary Pattern Operator for Texture Classification",IEEE Transactions on Image Processing.