

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Spatial Domain Representation for Face Recognition

Toshnall Meenpal, Aarti Goyal and Moumita Mukherjee

Abstract

Spatial domain representation for face recognition characterizes extracted spatial facial features for face recognition. This chapter provides a complete understanding of well-known and some recently explored spatial domain representations for face recognition. Over last two decades, scale-invariant feature transform (SIFT), histogram of oriented gradients (HOG) and local binary patterns (LBP) have emerged as promising spatial feature extraction techniques for face recognition. SIFT and HOG are effective techniques for face recognition dealing with different scales, rotation, and illumination. LBP is texture based analysis effective for extracting texture information of face. Other relevant spatial domain representations are spatial pyramid learning (SPLE), linear phase quantization (LPQ), variants of LBP such as improved local binary pattern (ILBP), compound local binary pattern (CLBP), local ternary pattern (LTP), three-patch local binary patterns (TPLBP), four-patch local binary patterns (FPLBP). These representations are improved versions of SIFT and LBP and have improved results for face recognition. A detailed analysis of these methods, basic results for face recognition and possible applications are presented in this chapter.

Keywords: spatial domain representation, face recognition, scale-invariant feature transform, histogram of oriented gradients, local binary patterns

1. Introduction

Face recognition is a powerful biometric system in today's highly technological world. It is widely accepted over other biometric systems like, finger print, iris or speech recognition for security, surveillance, and commercial applications. Face recognition system is generally a procedure of multiple major stages: face detection, preprocessing, feature extraction and verification. A complete structure of face recognition system is shown in **Figure 1**. Face detection detects a single face or number of faces present in a given image. Viola-Jones face detection algorithms using Haar features [1], faster R-CNN face detector [2], and face detection based on Histograms of Oriented Gradient [3] are popular methods for detecting faces in an image. Generally, images are captured under unconstrained environment and hence needed to be preprocessed before feeding to feature extraction stage. Preprocessing mainly aims to reduce noise effect, difference of illumination, color intensity, background, and orientation. The correct recognition of image depends upon quality of captured image, lighting condition etc. [4]. Recognition rate can be improved by performing pre-processing on the captured image. Various pre-processing

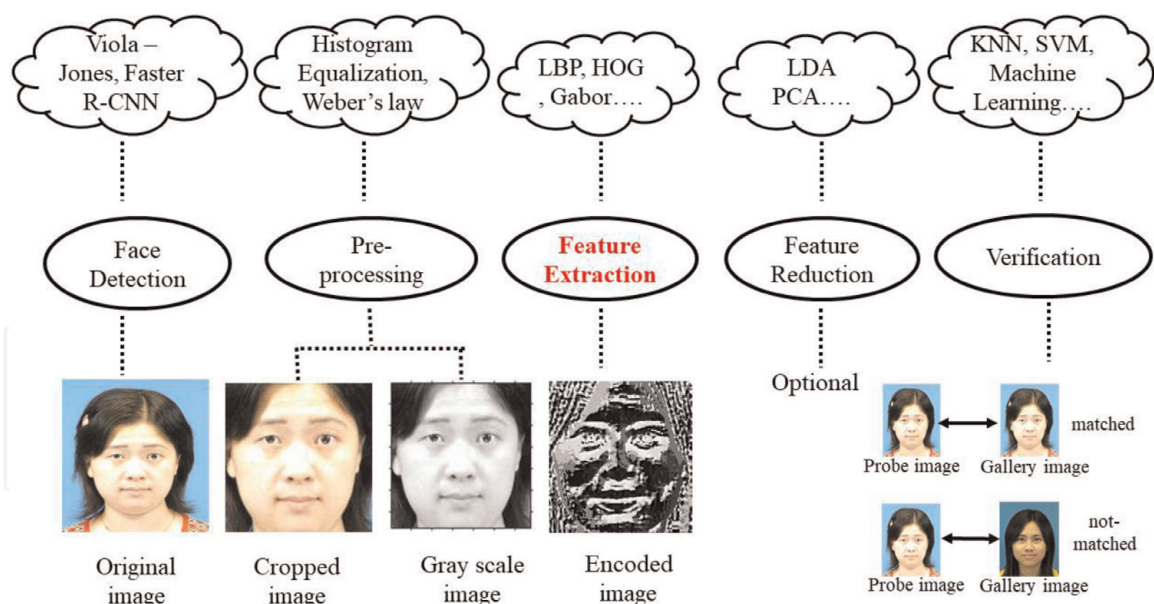


Figure 1.
A complete structure of face recognition system.

techniques are used in image processing to improve the recognition rate such as cropping, image resizing, histogram equalization and de-noising filtering as described below.

1. **Face Detection and Cropping**: - Face detection involves detecting face image from whole image. Cropping can be done based on one or more features of the image such as eyes, lips, nose etc.
2. **Image Resizing**: - Variation in face image size, shape, pose etc. raises difficulty for designing face recognition algorithms. So it is very important to resize image before feature extraction. For this, face images are cropped again into a standard size. Affine transformation can be applied on face with Bilinear Interpolation algorithm.
3. **Image Equalization**: - Illumination variation problem in the original resized image is overcome by using histogram equalization.
4. **Image De-noising and Filtering**: - Raw images are captured with many noise during the time of capturing the image and later also. Wiener filter and median filter are used to remove noises [5].

Next is feature extraction which is considered as the most prominent stage in face recognition system to extract discriminative facial features. Extracted features are then represented as feature vector and are fed to verification stage. Feature selection is an optional stage before verification which reduces feature vector dimensions using dimensional reduction techniques [6]. Final stage is verification to identify an unknown by finding closest matching in gallery.

2. Existing face databases

There are a number of benchmark face databases for fair face recognition evaluation by researchers. These databases are designed with images or videos of a

number of individuals with varying conditions and resolutions. A summary of benchmark face databases is tabulated in **Table 1**.

A detailed structure of some of these face databases are provided below.

2.1 A&T Database

A&T Database originally known as ORL database has face images captured in the interval April 1992 to April 1994. This database is collected by researchers of Cambridge University Engineering department for face recognition project. There are total 400 images in A&T database captured by taking 10 different images of 40 individuals. All images are captured in a dark homogeneous background with resolution 92×112 pixels. Different varying conditions under which images captured are- times, lighting, open eyes, closed eyes, smiling, not smiling, glasses, no glasses, some images also have rotation variation. This database has 40 different directories, each with 10 images of an individual stored as .pgm format. Samples of images of A&T database is shown in **Figure 2**.

2.2 CAS-PEAL-R1

CAS-PEAL-R1 Database is collected under sponsors of National Hi-Tech Program and ISVISION by the Face Recognition Group of JDL, ICT, CAS. This database contains 30,900 images of 1040 individuals captured under different conditions as such, variation in pose, facial expression, accessory, illumination, background, distance, and time. For pose variation, each of 1040 individuals has approximately 21 different poses. Facial expression is captured for 377 individuals with 6 different

Database	No. of individual	Conditions	Image Resolution	Images
A&T Database [7]	40 40 40 40 40	Lighting, Open eye, closed eyes, smiling, not smiling, glasses, no glasses	92×112	400
CAS-PEAL-R1 [8]	1040 377 438 233 297 296 66	Pose Facial expressions Accessory Illumination Background Distance Time	360×480	30,900
CMU Multi-PIE Database [9]	68	Pose Illumination Facial expressions	640×486	41,368
FERET [10]	1199	Pose Illumination Facial expressions Time	256×384	14,051
Korean Face Database (KFDB) [11]	1000	Pose Illumination Facial expressions	640×480	52,000
Yale Face Database B [12]	10	Pose Illumination	640×480	5850

Table 1.
Summary of benchmark face recognition databases.



Figure 2.
Samples of images of A&T database with 10 varying conditions [7].



Figure 3.
Samples of images of CAS-PEAL-R1 database [8].

expressions, similarly for accessory, 6 different images of 438 individuals with different accessory are used. Illumination variation has images of 233 individuals captured for minimum 10 and maximum 31 lighting variations. Background variation has images of 297 individuals for 2 to 4 different backgrounds. Further distance and time parameters have 296 and 66 individuals at an interval of 6-month. Samples of images of CAS-PEAL-R1 database are shown in **Figure 3**.

2.3 CMU Multi-PIE Database

CMU Multi-PIE Database is collected from October 2000 to December 2000 by taking 41,368 images of 68 individuals designed for 14 different poses, 43 illumination variation, and 4 different expressions. This database is known as CMU Multi-PIE by its varying conditions- pose, illumination, and expression. Image resolution is set to resolution 640×486 pixels. Samples of images of CMU Multi-PIE database is shown in **Figure 4**.

This chapter mainly focuses on feature extraction stage in face recognition. It presents some well-known and recently explored spatial domain representations for



Figure 4.
Samples of images of CMU Multi-PIE Database [9].

face recognition. Scale-invariant feature transform (SIFT), histogram of oriented gradients (HOG), and local binary patterns (LBP) are most commonly used spatial feature representations over past decade. Recently, other relevant feature representations, such as, spatial pyramid learning (SPLE), linear phase quantization (LPQ), variants of LBP such as improved local binary pattern (ILBP), compound local binary pattern (CLBP), local ternary pattern (LTP), three-patch local binary patterns (TPLBP), four-patch local binary patterns (FPLBP) are effectively used for face recognition.

3. Histogram of oriented gradients (HOG)

Histogram of oriented gradients (HOG) is introduced by Dalal et al. [13] in 2005 for human detection. HOG is an effective descriptor for face recognition by computing normalized histograms of face gradient orientations in dense grid [14]. Basically, HOG generates local appearance and shape of face rather than local intensity gradients. HOG is based on computation, fine orientation binning, normalization and descriptor blocks.

A detailed implementation for extracting HOG features for face recognition is given as:

1. Facial image is first divided into small regions called cells. For an image of size 64×64 , overlapping cells of 8×8 pixels are obtained. Gradient directions over pixels are computed for each cell. Simple 1-D derivatives are used in horizontal and vertical directions with the following masks:

$$D_x = [-1 \ 0 \ 1] \tag{1}$$

$$\text{and } D_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \tag{2}$$

Results for a sample facial image using horizontal (D_x) and vertical(D_y) derivative masks are shown in **Figure 5**.

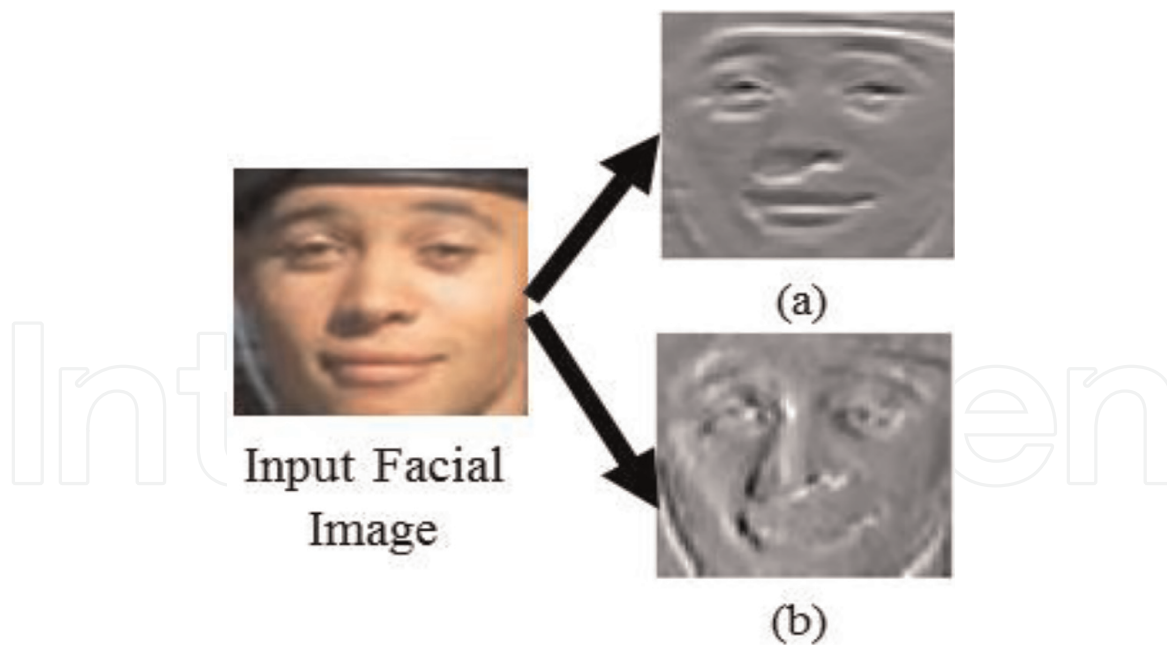


Figure 5.
Sample facial image and resultant derivatives. (a) Horizontal derivative. (b) Vertical derivative.

2. Next step is fine orientation binning for extracting HOG features. Histogram channels are evenly selected in the range $0-180^\circ$ for unsigned and $0-360^\circ$ for signed gradient. Each cell can contribute in the form of pixel magnitude, gradient magnitude, square root or square of magnitude. In general, gradient magnitude yields the best results while square root reduces the performance [13].
3. Gradients in each cell are normalized for local contrast normalization. Cell gradients are normalized from all blocks and are concatenated to form HOG feature vector. Dalal et al. [13] proposed 9 histogram channels (bins) to be computed for unsigned gradient. Hence, for 64×64 image, 1764 dimensional HOG feature vector is obtained representing full facial appearance. It can be explained as:

$$\frac{64 \times 64}{8 \times 8} \times 50\% \text{ overlapping} = 196 \text{ blocks} \quad (3)$$

$$196 \text{ blocks} \times 9 \text{ bin} = 1764 \text{ dimensional HOG vector} \quad (4)$$

4. Different normalization schemes are presented in [15] for block normalization. Let ν represents un-normalized block with $\|\nu\|_k$ as k^{th} norm for $k = 1, 2$ and

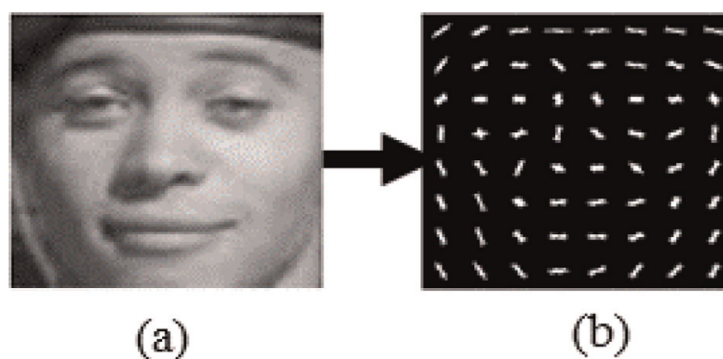


Figure 6.
Sample example of (a) Input facial image of size 64×64 . (b) Resultant HOG features (1764 dimensions).

q a small constant. Different normalization schemes used are L1-norm, L1-sqrt, L2-norm and L2-hys. Generally, L2-hys is used for block normalization. L2-hys is obtained by first computing L2-norm and then clipping such that maximum value of ν is limited to 0.2 and then renormalizing.

Sample input facial image and resultant HOG features are shown in **Figure 6**.

4. Scale invariant feature transform (SIFT)

Scale invariant feature transform (SIFT) is introduced by Lowe et al. [16] for extracting discriminative invariant features in an image. SIFT descriptor is widely used for facial feature representation by extracting blob-like local features [17]. These features are invariant to scale, translation and rotation resulting reliable matching. SIFT is described in four sections as: (1) Detection of scale-space extrema, (2) Detection of local extrema, (3) Orientation assignment, and (4) Keypoint descriptor representation.

4.1 Detection of scale-space extrema

First step is to identify keypoints in scale-space of grayscale input image $f(a, b)$ which is defined as:

$$L(a, b, \sigma) = G(a, b, \sigma) * f(a, b) \quad (5)$$

$$\text{such that, } G(a, b, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(a^2+b^2)/2\sigma^2} \quad (6)$$

where σ is standard deviation of Gaussian $G(a, b, \sigma)$.

Two closest scales of image with difference of multiplication factor k are used to effectively detect extrema in scale-space. Difference of Gaussian (DOG) is computed by taking difference of these two scaled versions of image convolved with original image given as:

$$\begin{aligned} D(a, b, \sigma) &= (G(a, b, k\sigma) - G(a, b, \sigma)) * f(a, b) \\ &= L(a, b, k\sigma) - L(a, b, \sigma) \end{aligned} \quad (7)$$

4.2 Detection of local extrema

Local extrema (maxima/minima) of $D(a, b, \sigma)$ is calculated by comparing sample pixel with eight neighbors in 3×3 patch as well as nine neighbors above and below scaled images. To select sample point as local minima, it should be smaller than all 26 neighbors whereas for local maxima, selected point should be larger than all neighbors. After keypoint localization, low contrast and poorly localized points are removed by computing $|D(a, b, \sigma)|$ and discarding points with lower value to defined threshold.

4.3 Orientation assignment

Orientation assignment to each keypoint results in rotation invariance. For each Gaussian smoothed image $L(a, b)$, orientation is assigned by computing gradient magnitude $m(a, b)$, and gradient direction $\theta(a, b)$ by its neighbor using Eqs. (8) and (9) respectively.

$$m(a,b) = \sqrt{(L(a+1,b) - L(a-1,b))^2 + (L(a,b+1) - L(a,b-1))^2} \quad (8)$$

$$\theta(a,b) = \tanh(L(a,b+1) - L(a,b-1)) / (L(a+1,b) - L(a-1,b)) \quad (9)$$

4.4 Keypoint descriptor representation

Finally, each detected keypoint is represented as 128 dimensional feature vector. This is obtained by computing magnitude and orientation of gradient at each point in 16×16 sized patch of an image. Each 16×16 patch is subdivided into 4×4 non-overlapping regions such that each 4×4 region is represented by 8 bins. Hence, each keypoint descriptor is represented by $4 \times 4 \times 8 = 128$ length vector.

Figure 7 shows an example of assignment of SIFT descriptor for 8×8 neighborhood. Length of each arrow corresponds sum of gradient magnitude in a specific direction for 4×4 region.

Processing flow to generate SIFT features for face recognition is shown in **Figure 8**. Input original image is first preprocessed and difference of Gaussian

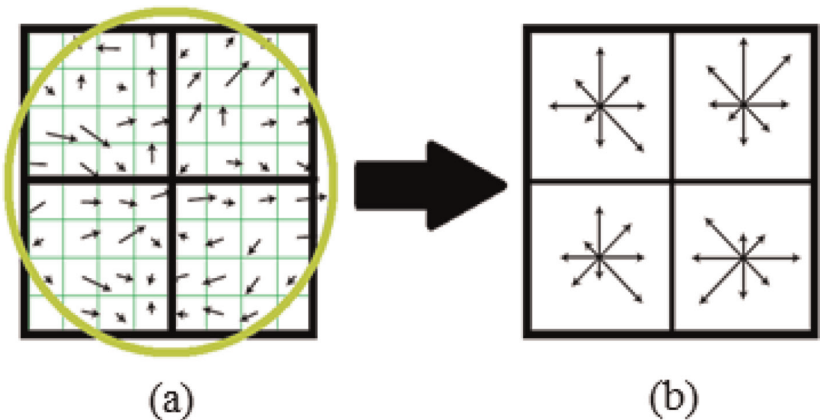


Figure 7.
Example of (a) Image gradients of 2×2 patch computed from 8×8 neighborhood. (b) Resultant SIFT keypoint descriptor.

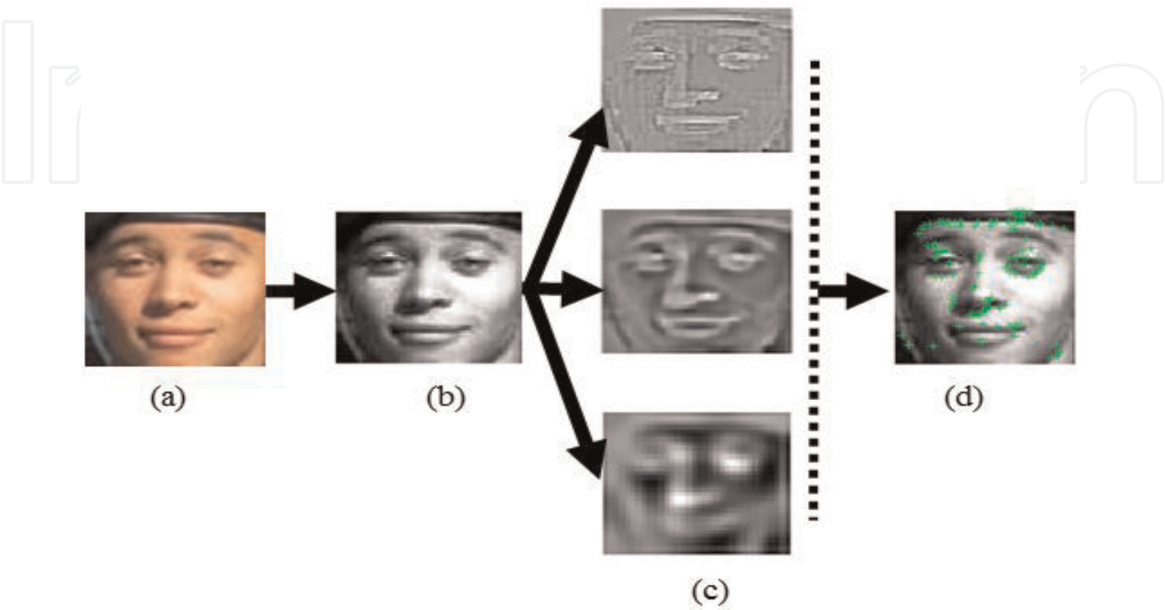


Figure 8.
Processing flow of SIFT for face recognition. (a) Original image. (b) Processed image. (c) Difference of Gaussian Pyramid. (d) SIFT keypoints.

pyramid is generated as in **Figure 8(c)**. Final resultant SIFT keypoints are then represented as feature vector to be fed to classifier for face recognition.

5. Linear phase quantization (LPQ)

Local phase quantization (LPQ) introduced by Ojansivu et al. [18, 19] is blur tolerant texture based descriptor. LPQ is based on blur invariance property of frequency domain phase spectrum of an image. LPQ for face recognition is investigated by Ahonen et al. [20] and reported improved results for blurred facial images.

LPQ on an image pixel is applied by using short-term Fourier transform (STFT) over $M \times M$ patch with image as center and four scalar frequencies. Imaginary and real components are then whitened and binary quantized to generate LPQ code for respective pixel. Complete process is detailed in **Figure 9** where LPQ code is obtained for an image pixel [21]. Similarly, final LPQ feature vector can be obtained by shifting $M \times M$ patch over the entire image.

Spatial blurring is performed by convolving grayscale input image $f(a, b)$ to point spread function (PSF). Frequency domain analysis can be represented as:

$$H(u, v) = F(u, v) \cdot P(u, v) \quad (10)$$

here, $F(u, v)$ and $P(u, v)$ are DFT of original image and PSF respectively. $H(u, v)$ is DFT of resultant blurred image.

Phase spectrum is obtained as:

$$\angle H(u, v) = \angle F(u, v) + \angle P(u, v) \quad (11)$$

Now, if PSF is positive and even, then $\angle P(u, v)$ must be either 0 or Π , such that $\angle P(u, v) = 0$ for $P(u, v) \geq 0$ while, $\angle P(u, v) = \Pi$ for $P(u, v) < 0$.

Since, shape of $P(u, v)$ generally selected is similar to Gaussian function, low frequency value of $P(u, v)$ is positive. This results $\angle P(u, v) = 0$ and Eq. (11)

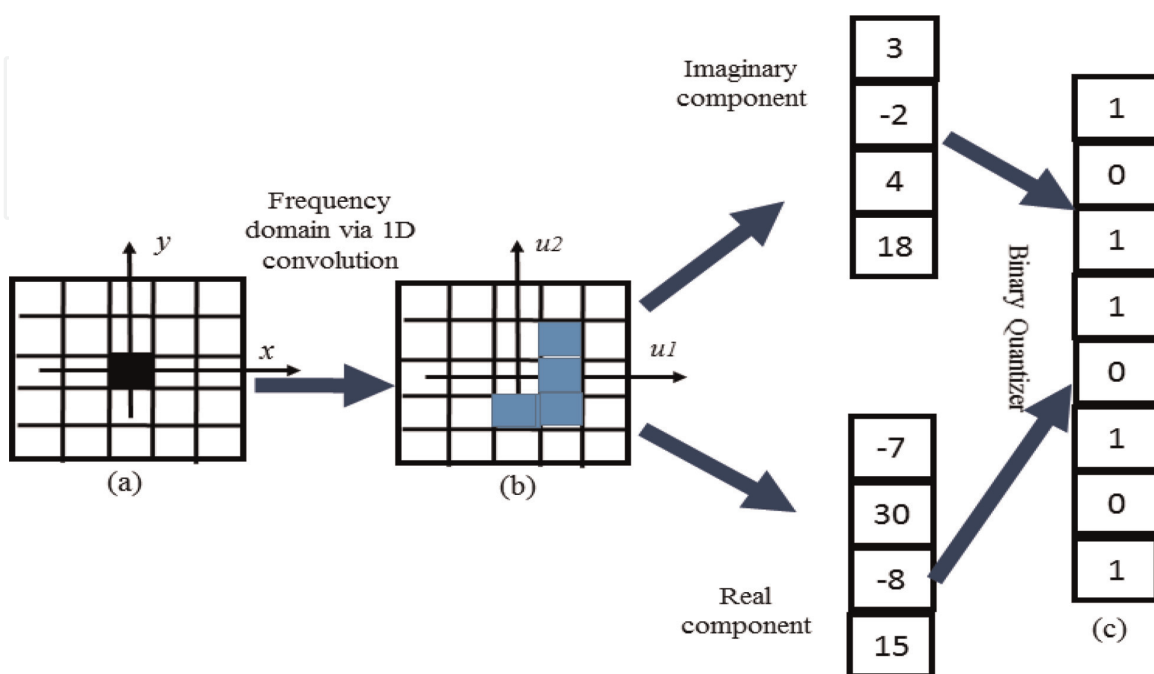


Figure 9. LPQ encoding scheme. (a) Input 5×5 patch. (b) Frequency domain representation. (c) LPQ code.

becomes $\angle H(u, v) = \angle F(u, v)$. Hence, it can be stated that LPQ possesses blur invariant property. Detailed mathematical analysis of LPQ can be obtained from [21].

6. Local binary patterns (LBP)

Local Binary Patterns (LBP) is introduced by Ojala et al. [22] as rotation invariant texture based feature descriptor. LBP as feature representation for face recognition is proposed by Ahonen et al. [23]. It stated that texture analysis of a local facial region represents its local appearance and fusion of all regions can generate an encoded global geometry of face.

Consider an input image and let $f(a, b)$ be its preprocessed version. Basic LBP operator on 3×3 neighborhood of $f(a, b)$ and generated decimal code for center pixel is shown in **Figure 10**. LBP operator replaces each pixel of $f(a, b)$ with a calculated decimal code resulting in LBP encoded image $f_{LBP}(a, b)$. It is done by thresholding each pixel of 3×3 neighborhood with its center pixel. Resultant is a binary code which is then converted into corresponding decimal code. Center pixel is then replaced by decimal code of generated binary stream. LBP code assigned to center pixel is given by Eq. (12). Here, i_c represents center pixel, c_n is gray level of neighbor pixels, and c_p is gray level of center pixel.

$$LBP_{P, R}(i_c) = \sum_{m=0}^{P-1} s(c_n - c_p) 2^m \quad (12)$$

$$s = \begin{cases} 1 & \text{if } c_n - c_p > 0 \\ 0 & \text{otherwise} \end{cases}$$

Ahonen et al. [23] proposed that LBP operator can be used with varying neighborhood size $M \times M$ and radius R to deal with different image scales. Notation (P, R) is used to represent P sampling points or neighbor pixels around center pixel for radius R . Thresholding is then performed by comparing center pixel with P neighbor pixels. Example of some selected values of (P, R) is shown in **Figure 11**.

LBP for face recognition processes by building local LBP descriptor to represent local region and then combined to obtain global representation for entire face. Encoded image $f_{LBP}(a, b)$ is evenly divided into non-overlapping blocks. Histogram for each block are calculated and final LBP feature vector is built by concatenating all regional histograms. LBP operator provides essential spatial information that plays a key role for face recognition. Complete processing flow to generate LBP feature vector is shown in **Figure 12**.

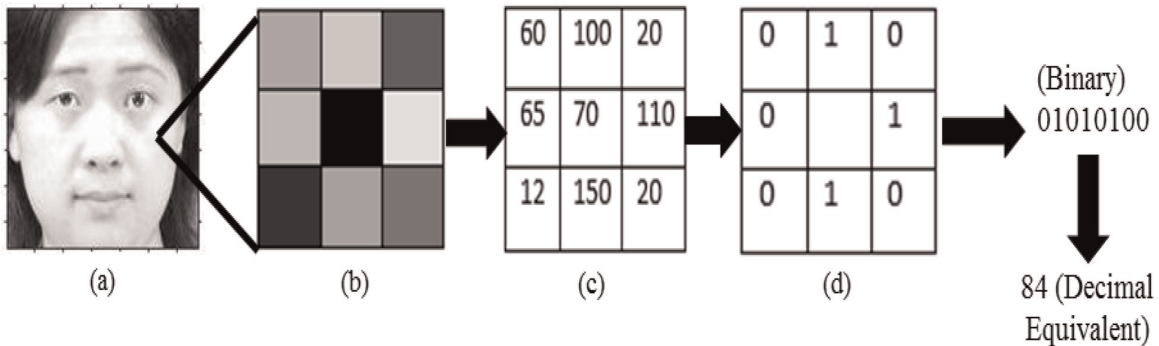


Figure 10.

Basic LBP operator on 3×3 neighborhood for $f(a, b)$. (a) Preprocessed image. (b) 3×3 Neighborhood. (c) Corresponding gray levels of each pixel. (d) Result after thresholding. Finally, center pixel is replaced by code 42.

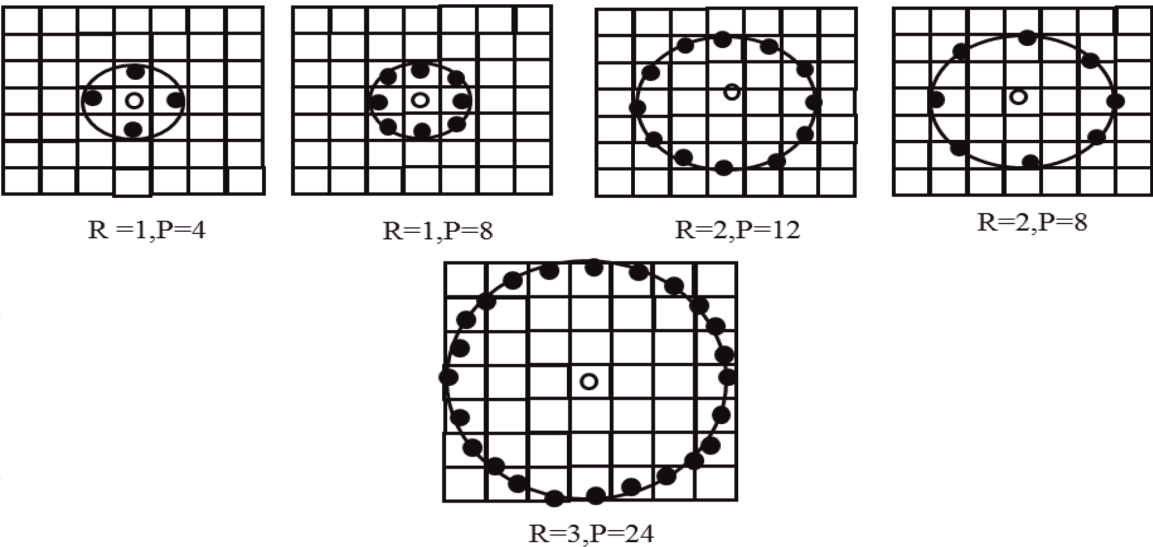


Figure 11.
Different P and R combinations for LBP operator.

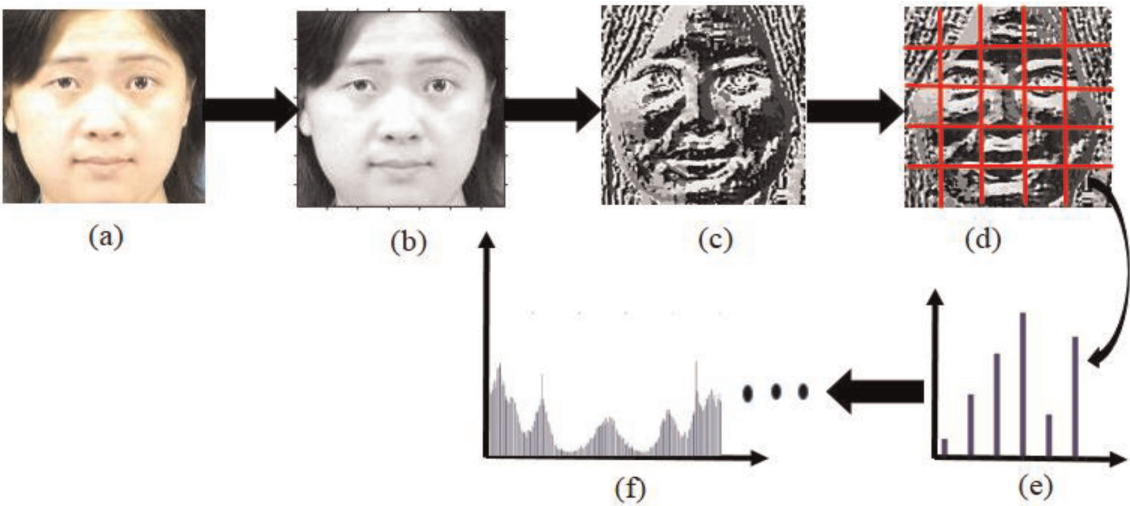


Figure 12.
Processing flow of LBP for face recognition. (a) Original input image. (b) Preprocessed image. (c) LBP Encoded image. (d) Divided non-overlapping patches for encoded image. (e) Histogram of selected non-overlapping patch. (f) Final LBP feature vector by concatenating histograms of all patches in image.

Major advantages of LBP over other spatial feature representations are simple calculations, comparatively smaller feature vector size, more powerful towards noises and illumination balance. In recent years, various variants of LBP are widely implemented in texture analysis. Local ternary patterns (LTP) proposed by Tan et al. [24] is based on a ternary threshold operator. LTP is an improved LBP variant by using two LBP vectors for building one LTP representation. Other variants of LBP are compound local binary pattern (CLBP) [25], three-patch LBP (TPLBP) [26], four-patch LBP (FPLBP) [26] and improved local binary pattern (ILBP) [27]. These representations are verified to be more efficient than LBP against illumination and noise conditions.

7. Local ternary patterns (LTP)

Local ternary patterns (LTP) [24] is a generalization of LBP with reduced sensitivity to noise and illumination variations. LTP generates a 3-valued code by including a threshold around zero and improves resistance to noise. LTP works well for noisy images and different lighting conditions.

In LBP, neighbor pixels are compared with center pixel directly. Hence, a small variation in pixel values due to noise can drastically change LBP code. To overcome this limitation, LTP introduces a threshold $\pm t$ around center pixel i_c and neighbor pixels are compared to generate 3-valued ternary code as:

$$LTP_{P,R}(i_c) = \sum_{m=0}^{P-1} s(c_n - c_p) 2^m \quad (13)$$

$$s = \begin{cases} 1 & c_p \geq c_n + t \\ 0 & |c_p - t| < t \\ -1 & c_p \leq c_p - t \end{cases} \quad (14)$$

Here, c_p and c_n represent gray levels of center pixel and neighbor pixels respectively. Understanding of LTP encoding scheme to generate ternary LTP code is shown in **Figure 13**. Here, threshold t is set to 5, hence with center pixel value 40, the tolerance range is $[35, 45]$. Neighbor pixels with gray level values in this range is replaced by zero, those above are replaced by 1 and below are replaced by -1 as described in Eq. (14).

Resultant ternary LTP code is split into two sub-LTP codes which are treated as two separate channels as shown in **Figure 14**. Lower and upper sub-LTP codes are

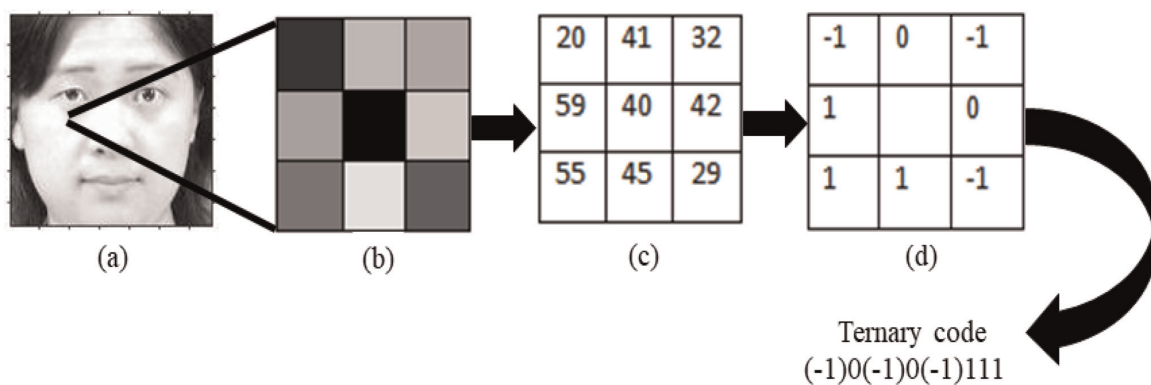


Figure 13.

LTP encoding scheme to generate ternary LTP code. (a) Preprocessed image. (b) 3×3 Neighborhood. (c) Corresponding gray levels of each pixel. (d) Ternary LTP code after thresholding.

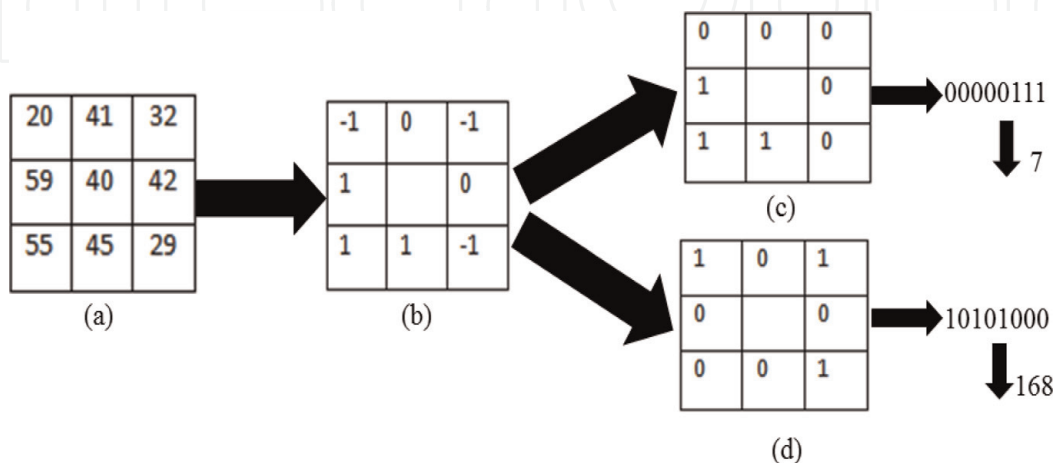


Figure 14.

Splitting of ternary LTP code to generate lower and upper sub-LTP codes. (a) 3×3 neighborhood of an image. (b) Ternary LTP code. (c) Lower sub-LTP code. (d) Upper sub-LTP code. Finally, lower and upper sub-LTP codes obtained are 7 and 168 respectively.

generated by replacing ‘-1’ in original ternary code to ‘0’ and ‘1’ respectively. Hence, LTP represents each original image by two encoded images.

8. Compound local binary pattern (CLBP)

Compound local binary pattern (CLBP) proposed by Ahmed et al. [25] is an improved variant of LBP using $2P$ bits code. CLBP overcomes limitation of LBP by improving performance in case of flat image. LBP results poor for images with bright spots or dark patches i.e. in case of flat image LBP fails as shown in **Figure 15**.

Original LBP generates P bits code by taking gray level difference between center pixel and P neighbor pixels (sampling points). CLPB is an extension to LBP by generating $2P$ bits code for P neighbor pixels. Here, extra P bits encode magnitude information of difference between center pixel and P pixels. This way, CLBP increases robustness of texture representation mainly in case of flat images.

To generate $2P$ bits code, CLBP represents each neighbor pixel with two bits for sign and magnitude information. The first bit is same as LBP bit and represents sign of difference between center pixel and respective neighbor pixel. Second bit encodes magnitude of difference with respect to a calculated threshold M_{ab} . This threshold is obtained by taking mean of magnitudes of difference between center pixel and all P pixels.

First bit is set to ‘1’ if gray level of neighbor pixel is greater than or equals to center pixel and ‘0’ otherwise. Second bit is ‘1’ if absolute magnitude of difference between neighbor pixel and center pixel is greater than threshold and ‘0’ otherwise. CLBP

$$CLBP_{P,R}(i_c) = \sum_{m=0}^{P-1} s(c_n - c_p) 2^m \tag{15}$$

$$s = \begin{cases} 00 & c_n - c_p < 0, \quad |c_n - c_p| \leq M_{ab} \\ 01 & c_n - c_p < 0, \quad |c_n - c_p| > M_{ab} \\ 10 & c_n - c_p \geq 0, \quad |c_n - c_p| \leq M_{ab} \\ 11 & \text{otherwise} \end{cases} \tag{16}$$

CLBP encoding scheme to generate $2P$ bits code for 3×3 neighborhood of an image is shown in **Figure 16**. A 16-bits CLBP code is generated after thresholding using Eq. (16). Resultant CLBP code is then split into two 8 bits sub-CLBP codes to reduce possible binary patterns from 2^{16} to (2×2^8) . First 8-bits code is

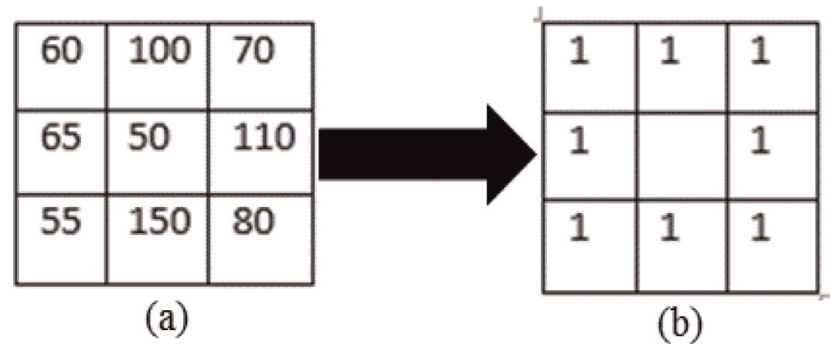


Figure 15.
LBP code for flat image. (a) 3×3 Neighbourhood of an image. (b) LBP encoded image.

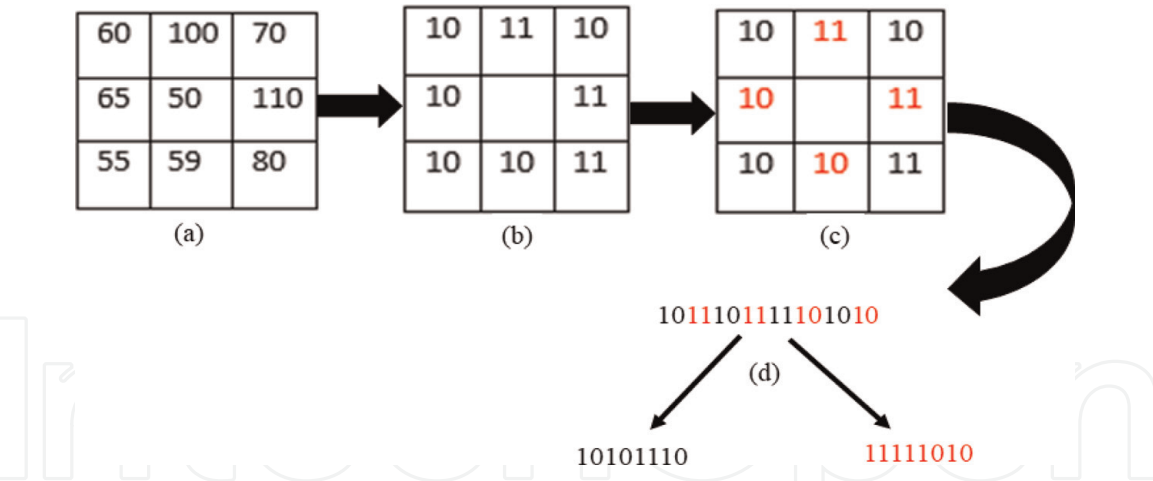


Figure 16. CLBP encoding scheme to generate 2P bits code. (a) 3×3 neighborhood of an image. (b) 2P bits CLBP code after thresholding. (c) Separated sub-CLBP codes. (d) Resultant two 8-bit sub-CLBP codes.

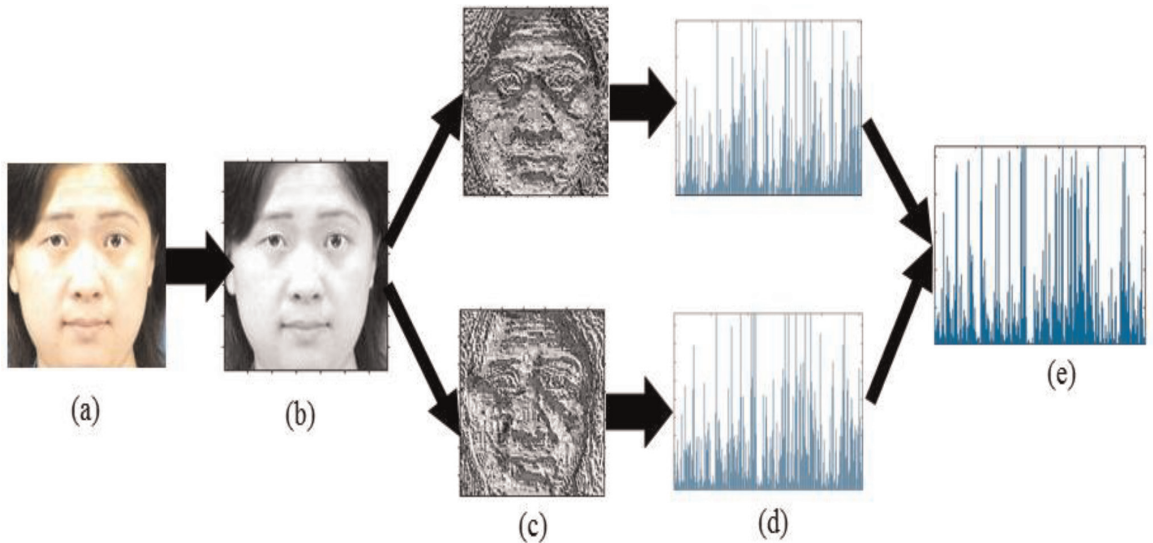


Figure 17. Processing flow of CLBP for face recognition. (a) Original image. (b) Preprocessed image. (c) Separated sub-CLBP encoded images. (d) Respective histograms of each encoded image. (e) Concatenated histogram.

concatenation of bits from pixels marked red in **Figure 16(c)**. Again, second 8-bits code is obtained by concatenating bit values from left over pixels. Finally, these sub-CLBP codes are treated as channels for final feature vector representation.

Processing flow to generate histograms of CLBP encoded image for face recognition is shown in **Figure 17**. It explains how each pixel of original image is converted into CLBP encoded image. **Figure 17(c)** shows two sub-CLBP encoded images. Histogram of each encoded image are obtained as in **Figure 17(d)**. These histograms can be individually used as separate feature vectors for face recognition or can be concatenated as a single final vector.

9. Three-patch LBP (TPLBP)

Original LBP and different variants of LBP generate 1-bit value or 2-bit value (for CLBP) by comparing two pixels, one as center pixel and other as one of the P neighbor pixels. Wolf et al. [26] proposed two different variants of LBP, namely, Three-patch LBP (TPLBP and Four-patch LBP (FPLBP) by comparing center pixel with more than one neighbor pixels.

TPLBP assigns each neighbor pixel in encoded image with 1-bit value by comparing gray level of three patches. For each center pixel i_c , $M \times M$ patch is considered and P additional same sized patches with center at distance of radius R is selected. Center pixel i_c is compared with center pixels of two patches at δ distance apart along the ring of radius R . This way, TPLBP generates P bits code for i_c as:

$$TPLBP_{P,R,M,\delta}(i_c) = \sum_{m=0}^{P-1} f(d(c_m, c_p) - d(c_{m+\delta \bmod M}, c_p))2^m \tag{17}$$

here, c_p , c_m and $c_{m+\delta \bmod M}$ are gray level of i_c , gray levels of center pixel of m^{th} and $(m + \delta)^{th}$ patches respectively. $d(.)$ is L_2 norm and f is given as:

$$f(a) = \begin{cases} 1, & a \geq \tau \\ 0, & a < \tau \end{cases} \tag{18}$$

τ is a user-specific threshold selected slightly greater than zero (say $\tau=.01$) to obtain stability in flat regions. **Figure 18** shows a sample example to generate TPBLP code for selected $P = 8$, $\delta = 2$, $M = 3$. TPLBP code generation for given sample using Eq. (17) is as:

$$\begin{aligned} &f(d(c_0, c_p) - d(c_2, c_p))2^0 + f(d(c_1, c_p) - d(c_3, c_p))2^1 + \\ &f(d(c_2, c_p) - d(c_4, c_p))2^2 + f(d(c_3, c_p) - d(c_5, c_p))2^3 + \\ &f(d(c_4, c_p) - d(c_6, c_p))2^4 + f(d(c_5, c_p) - d(c_7, c_p))2^5 + \\ &f(d(c_6, c_p) - d(c_0, c_p))2^6 + f(d(c_7, c_p) - d(c_1, c_p))2^7 \end{aligned} \tag{19}$$

Processing flow to obtain TPLBP feature vector for face recognition is shown in **Figure 19**. Input facial image of size 64×64 is first represented as TPLBP encoded image as in **Figure 19(c)**. TPLBP encoded image is then divided into non-overlapping patches of same size and histogram for each patch is obtained. These

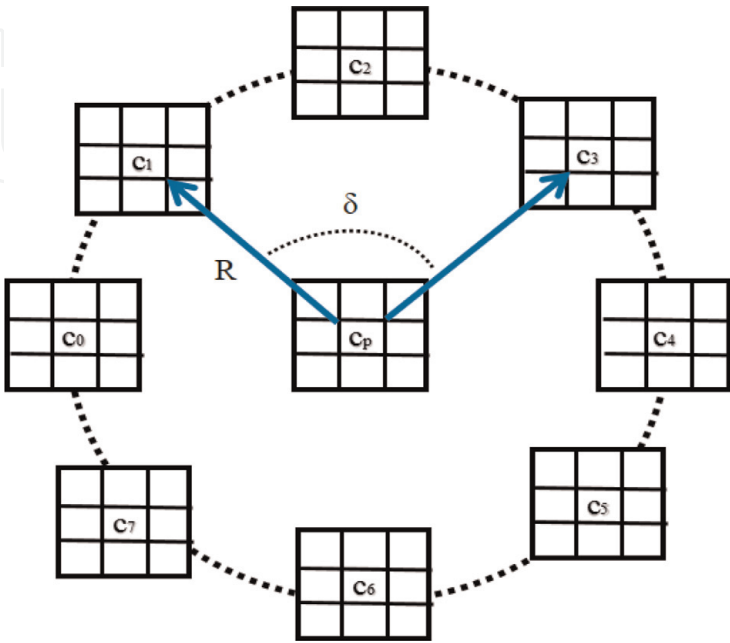


Figure 18.
TPLBP code generation for selected $P = 8$, $\delta = 2$, $M = 3$.

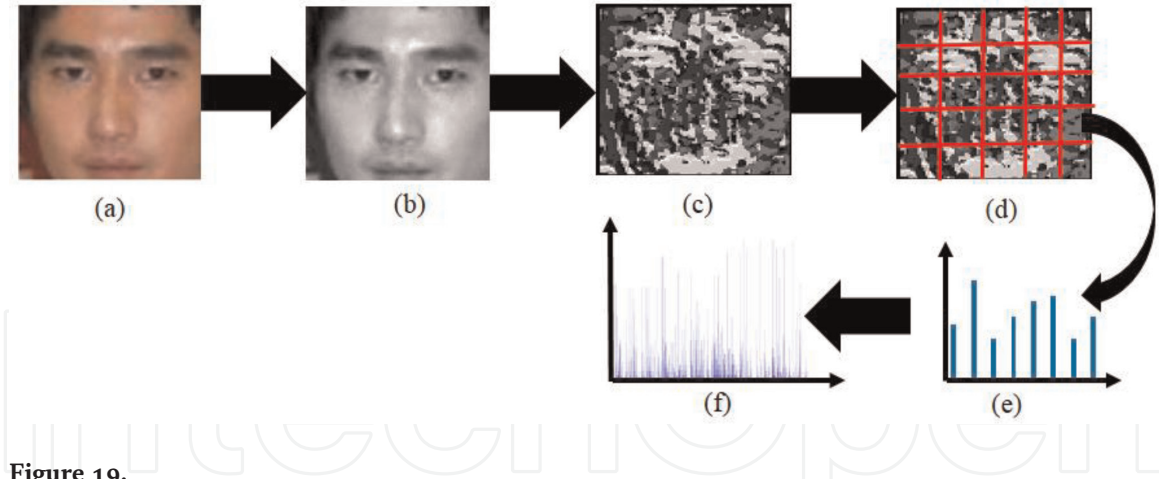


Figure 19. Processing flow of TPLBP for face recognition. (a) Original input image. (b) Preprocessed image. (c) TPLBP Encoded image. (d) Divided non-overlapping patches for encoded image. (e) Histogram of selected non-overlapping patch. (f) Final TPLBP feature vector by concatenating histograms of all patches in image.

histograms are then normalized and truncated to value 0.2. Finally, TPLBP feature vector is obtained by concatenating all histograms.

10. Four-patch LBP (FPLBP)

Four-patch LBP (FPLBP) [26] is an extension to TPLBP by comparing center pixels of four patches to generate 1-bit value. Two different rings with radius $R1$ and $R2$ ($R1 < R2$) and P patches of size $M \times M$ for each ring are selected around center pixel i_c . Two patches with center symmetric are selected in inner ring and compared with corresponding patches in outer ring at distance δ along a circle. This way, FPLBP generates $P/2$ bit code for i_c by obtaining $P/2$ pairs as:

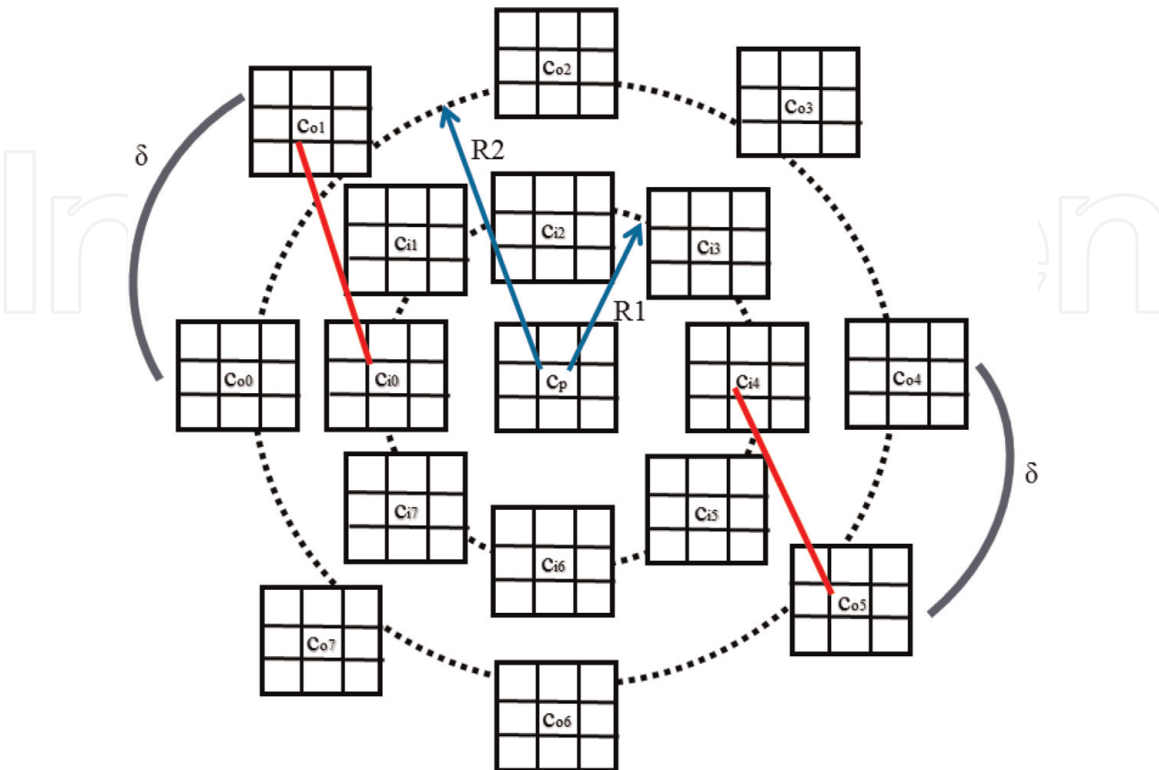


Figure 20. FPLBP code generation for selected $P = 8, \delta = 2, M = 3$.

$$FPLBP_{P,R1,R2,M,\delta}(i_c) = \sum_{m=0}^{P/2-1} f(d(c_{i,m}, c_{o,m+\delta \bmod M}) - d(c_{i,m+P/2}, c_{o,m+P/2+\delta \bmod M}))2^m \tag{20}$$

here, $c_{i,m}$ and $c_{o,m+\delta \bmod M}$ are gray levels of center pixel of m^{th} patch in inner ring and $(m + \delta)^{th}$ patch in outer ring respectively. Again, $c_{i,m+P/2}$ and $c_{o,m+P/2+\delta \bmod M}$ are gray levels of center pixel of center symmetric $(m + P/2)^{th}$ patch in inner ring and $(m + P/2 + \delta)^{th}$ patch in outer ring respectively. **Figure 20** shows a sample example to generate FPBLP code for selected $P = 8, \delta = 2, M = 3$. Also FPLBP code generation for given sample using Eq. (20) is as:

$$\begin{aligned} &f(d(c_{i0}, c_{o1}) - d(c_{i4}, c_{o5}))2^0 + f(d(c_{i1}, c_{o2}) - d(c_{i5}, c_{o6}))2^1 \\ &f(d(c_{i2}, c_{o3}) - d(c_{i6}, c_{o7}))2^2 + f(d(c_{i3}, c_{o4}) - d(c_{i7}, c_{o8}))2^3 \end{aligned} \tag{21}$$

Processing flow to obtain FPLBP feature vector for a sample facial image similar to TPLBP is shown in **Figure 21**.

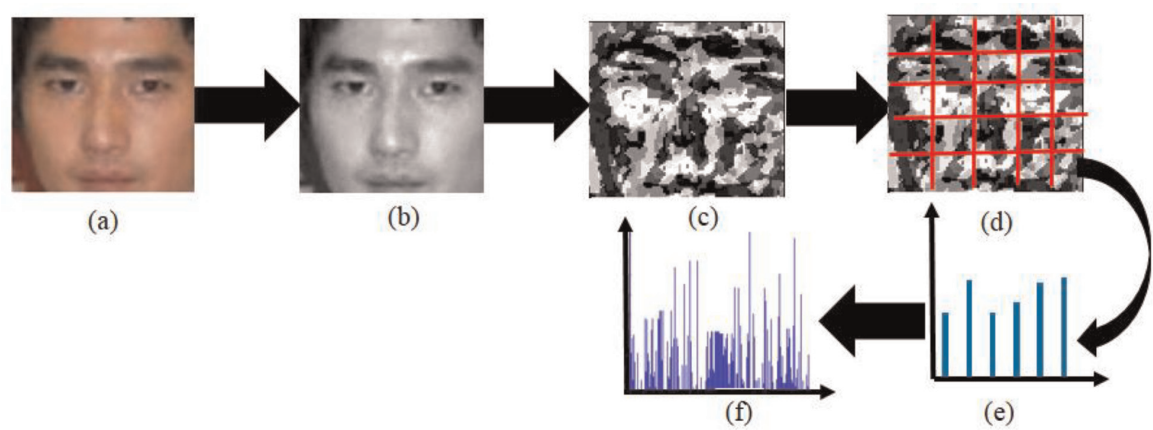


Figure 21.
Processing flow of FPLBP for face recognition. (a) Original input image. (b) Preprocessed image. (c) FPLBP Encoded image. (d) Divided non-overlapping patches for encoded image. (e) Histogram of selected non-overlapping patch. (f) Final FPLBP feature vector by concatenating histograms of all patches in image.

11. Improved LBP (ILBP)

Improved LBP (ILBP) originally named as CLBP (complete LBP) is proposed by Guo et al. [27]. It is termed as ILBP to distinguish its abbreviation from compound LBP (CLBP). In ILBP, neighbor pixels are represented by its center pixel and a local difference sign-magnitude transform (LDSMT). A complete processing flow to

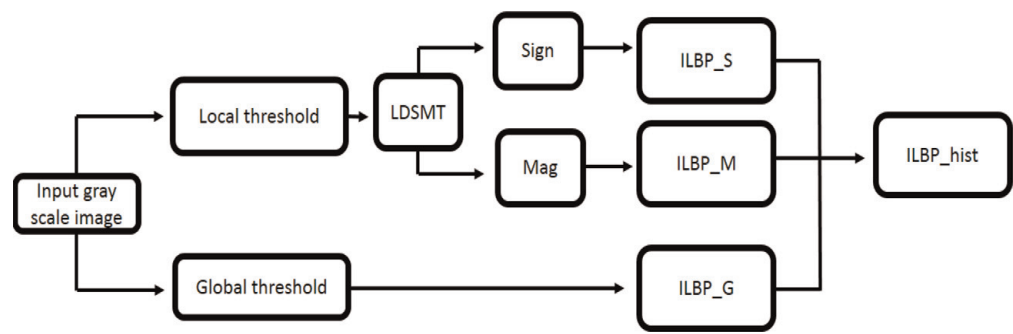


Figure 22.
Complete processing flow to generate ILBP code.

generate ILBP code is shown in **Figure 22**. ILPB generates $3P$ bits code for P neighbor pixels. An original image is first represented in terms of local threshold and global threshold. Local threshold is then further decomposed into sign and magnitude components. Consequently, three representations of P bits are obtained namely, ILBP_Sign (ILBP_S), ILBP_Magnitude (ILBP_M) and ILBP_Gobal (ILBP_G) and combined to form $3P$ bits ILBP code.

Let c_p and c_n represent gray levels of center pixel i_c and P neighbor pixels respectively. Local threshold is generated by taking difference $s_p = c_n - c_p$. Subtracted vector s_p is further divided into components, namely, magnitude of subtraction (m_p) and sign of subtraction (q_p) as:

$$s_p = q_p * m_p, \text{ where } \begin{cases} q_p = \text{sign}(s_p) \\ m_p = |s_p| \end{cases} \quad (22)$$

$$q_p = \begin{cases} 1, & s_p \geq 0 \\ -1, & s_p < 0 \end{cases} \quad (23)$$

Understanding of ILPB encoding scheme to generate $3P$ bits ILBP code is shown in **Figure 23**. **Figure 23(a)** shows 3×3 neighborhood with center pixel value 50. ILBP encoded image after local thresholding is shown in **Figure 23(b)** as $[-38, -15, 20, 20, 15, 22, -6, -41, 35]$. After LDSMT, sign and magnitude vectors are obtained. It is clearly seen that original LBP uses only sign as LBP encodes -1 as 0 in sign vector representation. LBP code for above sample block is $[0, 0, 1, 1, 1, 0, 0, 1]$. Hence, LBP considers only sign components of subtraction while ILBP combines three representations, ILBP_S, ILBP_M and ILBP_G. Local region around center pixel is represented by LDSMT, assigning threshold value w.r.t sign leads ILBP_S and assigning threshold value w.r.t. magnitude leads ILBP_M. Similarly, image is also encoded using global threshold is termed as ILBP_G.

A comparative analysis of various spatial domain feature representations is given in **Table 2**.

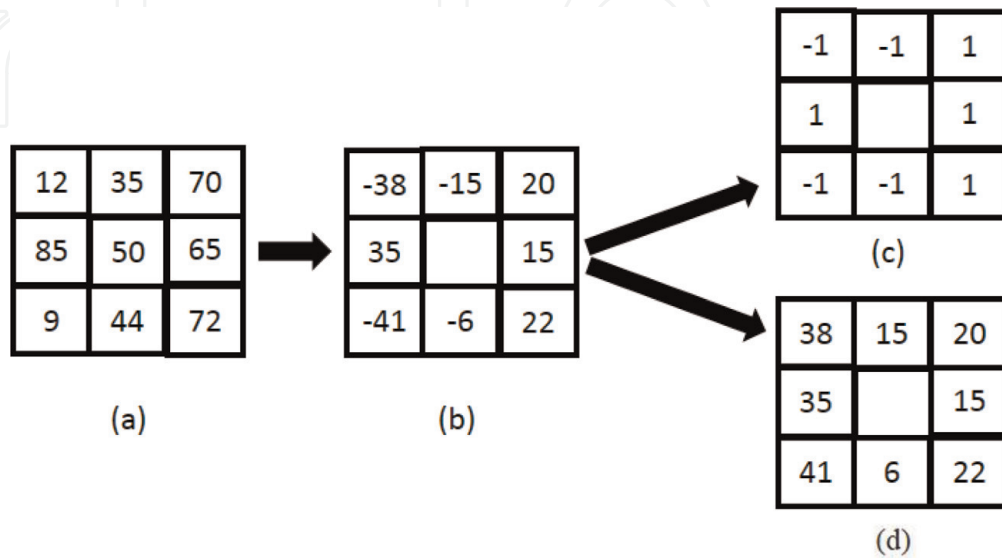


Figure 23. ILBP encoding scheme. (a) 3×3 neighborhood of an image. (b) ILBP encoded image after thresholding. (c) Sign component. (d) Magnitude component.

Feature	Advantages	Disadvantages
HOG	<ul style="list-style-type: none">• Rotation and scale invariant.	<ul style="list-style-type: none">• Very sensitive to image rotation. Not good choice for classification of textures or objects.
SIFT	<ul style="list-style-type: none">• Rotation and scale invariant.	<ul style="list-style-type: none">• Mathematically complicated and computationally heavy.• It is not effective for low powered devices.
LBP	<ul style="list-style-type: none">• High discriminative power.• Computational simplicity.	<ul style="list-style-type: none">• Not invariant to rotations.• Size of feature vector increases exponentially with number of neighbors leading to an increase of computational complexity in terms of time and space.• The structural information captured by it is limited. Only pixel difference is used, magnitude information ignored.• Performance decreases for flat images.
LPQ	<ul style="list-style-type: none">• Performance is better as compare to LBP in case of blurred illumination and facial expression variations images.	<ul style="list-style-type: none">• LPQ vector is about four times longer than an LBP vector with 8 neighbor pixels.
CLBP	<ul style="list-style-type: none">• It gives better performance as compared to LBP as it uses both difference sign and magnitude.	<ul style="list-style-type: none">• Feature vector is too long so it increases computational time.
LTP	<ul style="list-style-type: none">• Resistant to noise.	<ul style="list-style-type: none">• Not invariant under gray-scale transform of intensity values as its encoding is based on a fixed predefined thresholding.
TPLBP	<ul style="list-style-type: none">• Rotation invariant for texture descriptor.• Capture information for not only microstructure but also macrostructure.	<ul style="list-style-type: none">• Complexity increases.
FPLBP	<ul style="list-style-type: none">• Rotation invariant for texture descriptor.• Capture information for not only microstructure but also macrostructure.	<ul style="list-style-type: none">• More complex.

Table 2.
Comparative analysis of spatial domain feature representations.

12. Result analysis for face recognition

Face recognition has been explored over last many years, hence there exists a large number of researches in this domain. In this section, we present existing face recognition results and analysis based on different spatial domain representations. Deniz et al. [28] proposed face recognition using HOG features by extracting features from varying image patches which resulted in an improved accuracy. Recognition accuracy is evaluated on FERET database with best result of 95.4%. Other related researches are [29] which used EBGM-HOG and showed robustness to change in illumination, rotation and small displacements. Some existing works on face recognition using SIFT features are [30, 31]. These works have also used

variants of SIFT such as volume-SIFT (VSIFT), partial-descriptor-SIFT (PDSIFT), learning SIFT at specific locations to improve verification accuracy.

Face recognition using LPQ feature representation is inspired by [18, 19] which used LPQ as blur invariant descriptor. Damane et al. [32] presented face recognition using LPQ under varying conditions of light, blur, and illumination. Experiments are performed on extended YALE-B, CMU-PIE, and CAS-PEAL-R1 face databases and results showed that LPQ has more robustness to light and illumination variation. Chan et al. [33] presented multiscale LPQ for face recognition and evaluated results on FERET and BANCA face databases. Multiscale LPQ is obtained by applying varying filter size and combining LPQ images, which are then projected into LDA space. Best results of 99.2% for FB, 92% for DP1 and 88% for DP2 are achieved on FERET probe sets.

Face recognition using LBP feature representation is one of the most researched area [34–38]. Again, Tan et al. [24] evaluated face recognition under varying lighting condition using LTP feature representation on Extended Yale-B, and CMU PIE face databases. They showed that LTP is more discriminant and less sensitive to noise in uniform regions and improved results in case of flat images. Wolf et al. [26] proposed TPLBP and FPLBP features for face recognition. Accuracy results are validated on two well-known databases, labeled faces in the wild (LFW) and multi PIE. They showed that combining several descriptors from the same LBP boosts family recognition rate. This paper claimed that best accuracy of 80.75% for TPLBP and 75.57% for FPLBP are obtained with the combination of ITML with MultiOSS ID and pose variation. Ahmed et al. [25] proposed CLBP features for facial expression recognition. It is an extension of LBP features. Results are verified in Cohn-Kanade (CK) facial expression database. CLBP features are classified with the help of SVM classifier. They showed that classification rate can be effected by adjusting the number of regions into which expression images are partitioned. For this, they considered three cases by dividing images into 3×3 , 5×5 , and 7×6 patches. Best accuracy result for CLBP is 94.4% in case of image with 5×5 patch size.

13. Conclusion

This chapter presents well-known and some recently explored spatial feature representations for face recognition. These feature representations are scale, translation and rotation invariants for 2-D face images. This chapter covers HOG, SIFT and LBP feature representations and complete processing flow to generate feature vectors using these representations for face recognition. SIFT and HOG based on computing image gradients and local extrema are commonly used feature representations for face recognition. LBP performs texture based analysis to represent local facial appearance and an encoded facial image. Other relevant spatial domain representations, such as, LPQ and variants of LBP are explained and analyzed for face recognition. LPQ possesses blur invariant property and provides improved results for blurred facial image. Different variants of LBP, such as, LTP, CLBP, TPLBP and FPLBP are more robust to noise and lighting conditions. These representations characterize facial features more effectively and obtain discriminative feature vectors for face recognition.

Acknowledgements

The research work is supported by Science and Engineering Research Board (SERB), Department of Science and Technology (DST), Government of India for

the research grant. The sanctioned project title is “Design and development of an Automatic Kinship Verification system for Indian faces with possible integration of AADHAR Database.” with reference no. ECR/2016/001659.

Conflict of interest


The authors have no conflict of interest.

Author details

Toshanal Meenpal*, Aarti Goyal and Moumita Mukherjee
Electronics and Telecommunication Department, National Institute of Technology,
Raipur, Chhattisgarh, India

*Address all correspondence to: tmeenpal.etc@nitrr.ac.in

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Viola P, Jones MJ. Robust real-time face detection. *International Journal of Computer Vision*. 2004;**57**(2):137-154
- [2] Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*. 2015. pp. 91-99
- [3] King DE. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*. 2009;**10**(Jul): 1755-1758
- [4] Dharavath K, Talukdar FA, Laskar RH. Improving face recognition rate with image preprocessing. *Indian Journal of Science and Technology*. 2014;**7**(8):1170-1175
- [5] Gross R, Brajovic V. An image preprocessing algorithm for illumination invariant face recognition. In: *International Conference on Audio- and Video-Based Biometric Person Authentication*. Berlin, Heidelberg: Springer; 2003. pp. 10-18
- [6] Chandrashekar G, Sahin F. A survey on feature selection methods. *Computers and Electrical Engineering*. 2014;**40**(1, 1):16-28
- [7] Available from: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- [8] Gao W, Cao B, Shan S, Chen X, Zhou D, Zhang X, et al. The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*. 2008;**38**(1): 149-161
- [9] Sim T, Baker S, Bsat M. The CMU pose, illumination, and expression (PIE) database. In: *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*. IEEE; 2002. pp. 53-58
- [10] Phillips PJ, Moon H, Rauss P, Rizvi SA. The FERET evaluation methodology for face-recognition algorithms. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE; 1997. pp. 137-143
- [11] Hwang BW, Roh MC, Lee SW. Performance evaluation of face recognition algorithms on Asian face database. In: *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004. *Proceedings*. IEEE; 2004. pp. 278-283
- [12] Available from: http://vision.ucsd.edu/datasets/yale_face_dataset_original/yalefaces.zip
- [13] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: *International Conference on computer vision & Pattern Recognition (CVPR'05)*. Vol. 1. IEEE Computer Society; 2005. pp. 886-893
- [14] Shu C, Ding X, Fang C. Histogram of the oriented gradient for face recognition. *Tsinghua Science and Technology*. 2011;**16**(2):216-224
- [15] Dadi HS, Pillutla GK. Improved face recognition rate using HOG features and SVM classifier. *IOSR Journal of Electronics and Communication Engineering*. 2016;**11**(04):34-44
- [16] Lowe DG. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. 2004;**60**(2):91-110
- [17] Bicego M, Lagorio A, Grosso E, Tistarelli M. On the use of SIFT features for face authentication. In: *2006 Conference on Computer Vision and*

Pattern Recognition Workshop (CVPRW'06). IEEE; 2006. pp. 35-35

[18] Ojansivu V, Heikkilä J. Blur insensitive texture classification using local phase quantization. In: International Conference on Image and Signal Processing. Berlin, Heidelberg: Springer; 2008. pp. 236-243

[19] Rahtu E, Heikkilä J, Ojansivu V, Ahonen T. Local phase quantization for blur-insensitive image analysis. *Image and Vision Computing*. 2012;**30**(8): 501-512

[20] Ahonen T, Rahtu E, Ojansivu V, Heikkilä J. Recognition of blurred faces using local phase quantization. In: 2008 19th International Conference on Pattern Recognition. IEEE; 2008. pp. 1-4

[21] Nguyen HT. Contributions to facial feature extraction for face recognition [Doctoral dissertation]. Université de Grenoble, Sep 2014

[22] Ojala T, Pietikäinen M, Harwood D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*. 1996;**29**(1, 1):51-59

[23] Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2006;**1**(12):2037-2041

[24] Tan X, Triggs W. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*. 2010;**19**(6):1635-1650

[25] Ahmed F, Hossain E, Bari AS, Hossen MS. Compound local binary pattern (clbp) for rotation invariant texture classification. *International Journal of Computers and Applications*. 2011;**33**(6):5-10

[26] Wolf L, Hassner T, Taigman Y. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2011;**33**(10):1978-1990

[27] Guo Z, Zhang L, Zhang D. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*. 2010;**19**(6): 1657-1663

[28] Déniz O, Bueno G, Salido J, De la Torre F. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*. 2011; **32**(12):1598-1603

[29] Albiol A, Monzo D, Martin A, Sastre J, Albiol A. Face recognition using HOG-EBGM. *Pattern Recognition Letters*. 2008;**29**(10):1537-1543

[30] Križaj J, Štruc V, Pavešić N. Adaptation of SIFT features for robust face recognition. In: International Conference Image Analysis and Recognition. Berlin, Heidelberg: Springer; 2010. pp. 394-404

[31] Sadeghipour E, Sahragard N. Face recognition based on improved SIFT algorithm. *International Journal of Advanced Computer Science and Applications*. 2016;**7**(1):547-551

[32] Damane Local Phase-Context for Face Recognition under Varying Conditions. *Procedia Computer Science*. 2014;**39**:12-19

[33] Chan CH, Kittler J, Poh N, Ahonen T, Pietikäinen M. (Multiscale) local phase quantisation histogram discriminant analysis with score normalisation for robust face recognition. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. IEEE; 2009. pp. 633-640

[34] Huang D, Shan C, Ardebilian M, Chen L. Facial image analysis based on local binary patterns: A survey. *IEEE Transactions on Image Processing*. 2011; **41**:1-14

[35] Shan C, Gong S, McOwan PW. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*. 2009; **27**(6):803-816

[36] Zhang G, Huang X, Li SZ, Wang Y, Wu X. Boosting local binary pattern (LBP)-based face recognition. In: *Advances in Biometric Person Authentication*. Berlin, Heidelberg: Springer; 2004. pp. 179-186

[37] Dadiz BG, Ruiz CR. Detecting depression in videos using uniformed local binary pattern on facial features. In: *Computational Science and Technology*. Singapore: Springer; 2019. pp. 413-422

[38] Liu L, Fieguth P, Zhao G, Pietikäinen M, Hu D. Extended local binary patterns for face recognition. *Information Sciences*. 2016; **358**:56-72