Chapter 6

SIFT (Scale Invariant Feature Transform) based Face Recognition

6.1 Introduction

The primary motivation for the work proposed in this chapter is the search for an appropriate methodology for robust face recognition that will provide satisfactory performance based on a single training image. This is a significant attempt as the availability of gallery images for different subjects in any face recognition applications is limited. The approaches proposed in the previous chapters provide promising results and have simple implementations. However their performance steadily degrades with the reduction in the number of gallery images. After reviewing the recent trend of feature extraction proposed by various researchers, it has been found that the SIFT (Scale Invariant Feature Transform) features are most suited to satisfy the objectives of the proposed work, mainly due to its inherent properties such as invariance to image scaling and rotation, partial occlusion and to a certain extent to the changes in illumination and 3D camera view point. [10, 16]. However it also has some deficiencies when applied to the problem of face recognition. Compared to general objects, there are less structures with high contrast or high-edge responses in facial images. Since key points along edges and low-contrast key points are removed by the original SIFT algorithm, the interest points representing distinctive facial features can also be removed. Therefore, it is very important to properly adjust
the thresholds while removing the unstable key points. Variable illumination still has a significant influence on the detection of key points since the key point detector intrinsic to the SIFT technique is not invariant to illumination [16]. Also intrinsic changes like facial expressions lead to degradation of face recognition performance using SIFT. In this chapter two different algorithms based on SIFT descriptors have been proposed. The first algorithm is designed to tackle the problem of illumination variance and the second algorithm is designed to take care of expression variations. It is important to note that both the algorithms are designed by using single image for training.

### 6.2 SIFT Descriptor

SIFT feature extraction [66] method consists of 4 main steps \( \text{viz;} \) (i) Scale Space Extrema Detection (ii) Removal of unreliable keypoints (iii) Orientation Assignment and (iv) Keypoint Descriptor Calculation. The details are stated in subsequent sub-sections below.

#### 6.2.1 Scale Space Extrema Detection

The Scale Space Extrema Detection of SIFT algorithm involves the identification of interest points or key points in the scale space by finding the image locations that represent maxima or minima of the difference-of-Gaussian function \( D(x,y,\sigma) \). The first stage of computation searches over all scales and image locations. At each candidate location, a detailed model is fit to determine location and scale. Keypoints are usually selected based on measures of their stability.
The scale space of an image is defined as a function \( L(x, y, \sigma) \), that is produced from the convolution of a variable-scale Gaussian \( G(x, y, \sigma) \), with the input image, \( I(x,y) \)

\[
L(x, y, \sigma) = G(x, y, \sigma) \ast I(x, y),
\]

Eq. (6.1)

with

\[
G(x, y, \sigma) = \frac{1}{2\pi \sigma^2} e^{-(x^2 + y^2)/2\sigma^2}
\]

Eq. (6.2)

where \( \sigma \) denotes the standard deviation of the Gaussian \( G(x, y, \sigma) \)

The difference-of-Gaussian function \( D(x, y, \sigma) \) computed from the difference of Gaussians of two scales that are separated by a factor \( k \) can be given as:

\[
D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \ast I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma).
\]

Eq. (6.3)

Local maxima and minima of \( D(x, y, \sigma) \) are computed based on the comparison of the sample point and its eight neighbours in the current image as well as the nine neighbours in the scale above and below. Normalization of the Laplacian with the factor \( \sigma^2 \) is required for true scale invariance. The maxima and minima of \( \sigma^2 \nabla^2 G \) produce the most stable image features compared to a range of other possible image functions, such as the gradient, Hessian, or Harris corner function. Following [66], it can be seen that \( D(x, y, \sigma) \) provides a close approximation to the scale-normalized Laplacian of Gaussian \( \sigma^2 \nabla^2 G \).

### 6.2.2 Removal of unreliable keypoints

The final keypoints are selected based on measures of their stability. During this stage, low contrast points and poorly localized points along edges are discarded to remove noises and
instability. For detection of unreliable key points, the value of $|D(x, y, \sigma)|$ at each candidate key point is computed. If the value is below some threshold, the key point is removed as the structure has low contrast. The Taylor expansion up to the quadratic terms of the scale-space function $D(x, y, \sigma)$ can be shifted to compute the origin at the sample point.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x$$

Eq. (6.4)

where $D$ and its derivatives are evaluated at the sample point and $x = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum $\hat{x}$, is determined by taking the derivative of this function with respect to $x$ and setting it to zero and is given as:

$$\hat{x} = -\frac{\partial^2 D}{\partial x^2}^{-1} \frac{\partial D}{\partial x}.$$ 

Eq. (6.5)

If the offset $\hat{x}$ is larger than 0.5 in any dimension, then it means that the extremum lies closer to a different sample point. In this case, the sample point is changed and the interpolation performed about that point. The final offset $\hat{x}$ is added to the location of its sample point to get the interpolated estimate for the location of the extremum. The function value at the extremum, $D(\hat{x})$, is useful for rejecting unstable extrema with low contrast. This can be obtained by substituting equation (6.5) into (6.4), giving

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x}.$$ 

Eq. (6.6)

For the present study, all extrema with a value of $|D(\hat{x})|$ less than 0.03 has been discarded assuming that the image pixel values are normalized between 0 and 1. Next, the ratio of principal curvatures of each candidate key point has been evaluated to search for poorly defined peaks in
the Difference-of-Gaussian function. For key points with high edge responses, the principal curvature across the edge will be much larger than the principal curvature along it. If the ratio is below some threshold \((r = 10)\), the key point has been retained, otherwise it is removed.

### 6.2.3 Orientation assignment

An orientation is assigned to each key point by building a histogram of gradient orientations \(\theta(x,y)\) weighted by the gradient magnitudes \(m(x, y)\) from the neighbourhood of the key point, where \(\theta\) and \(m\) are given as:

\[
\begin{align*}
m(x, y) &= \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \\
\theta(x, y) &= \tan^{-1}((L(x, y + 1) - L(x, y - 1))/(L(x + 1, y) - L(x - 1, y)))
\end{align*}
\]

where \(L\) is a Gaussian smoothed image with a closest scale to that of a key point.

By assigning a consistent orientation to each key point, the key point descriptor can be represented relative to this orientation and therefore invariance to image rotation can be achieved.

### 6.2.4 Key point Descriptor Calculation

The gradient magnitude and orientation at each image point of the \(16 \times 16\) keypoint neighbourhood is obtained as shown in Fig. 6.1(a). The neighbourhood is then weighted by a Gaussian window and accumulated into orientation histograms summarizing the contents over sub regions of the neighbourhood of size \(4\times4\) as shown in Fig.6.1 (b). The length of each arrow
in Fig.6.1 (b) corresponding to the sum of the gradient magnitudes near that direction within the region [66]. As each histogram contains 8 bins, therefore each keypoint descriptor features 4 x 4 x 8 = 128 elements.

![Fig. 6.1(a) Image Gradients](image1)

![Fig. 6.1(b) Key point Descriptor](image2)

**Fig.6.1. 2x2 Descriptor array computed from an 8x8 set of samples**

### 6.3 Adaptive N1 Means Normalization

The adaptive N1 Means normalization procedure proposed by Struc. et al [24] is a variation of the Non Local Means (NL Means) algorithm as proposed by Buades et al [21]. The NL Means algorithm is an image de-noising technique which is based on the fact that for every small window of the image several similar windows can be found in the image as well and all of these windows can be exploited to de-noise the image.

Let $I_n(x) \in \mathbb{R}^{a \times b}$ be noisy image, where a and b are image dimensions in pixels, and x represents an arbitrary pixel location $x = (x, y)$ within the image. The NL means algorithm then constructs the de-noised image $I_d(x)$ by computing each pixel value of $I_d(x)$ as a weighted average of pixels comprising $I_n(x)$, i.e.
\[ I_d(x) = \sum_{x \in I_n(x)} w(z, x) I_n(x), \]

Eq. (6.8)

where \( w(z, x) \) represents the weighting function that measures the similarity between the local neighbourhoods of the pixel at the spatial locations \( z \) and \( x \). The weighting function can be defined as follows.

\[ w(z, x) = \frac{1}{Z(z)} e^{-\frac{G_\sigma \| I_n(\Omega_x) - I_n(\Omega_z) \|^2}{2h^2}} \quad \text{and} \quad Z(z) = \sum_{x \in I_n(x)} e^{-\frac{G_\sigma \| I_n(\Omega_x) - I_n(\Omega_z) \|^2}{2h^2}}, \]

Eq. (6.9)

where,

- \( G_\sigma \) is a Gaussian kernel with the standard deviation \( \sigma \).
- \( \Omega_x \) and \( \Omega_z \) are the local neighbourhoods of the pixels at the locations \( x \) and \( z \), respectively.
- \( h \) is the decay parameter of the exponential function, and
- \( Z(z) \) is a normalizing factor.

The proposed algorithm can now be implemented to obtain a smoothed image with preserved edges by choosing suitable value of the decay parameter \( h \). Accordingly, the luminance and also the reflectance of an input image can be estimated by using the smoothed image obtained as discussed above.

In the adaptive N1 Means normalization procedure the decay parameter \( h \) is calculated as a function of local contrast. As the images are more smoothened in the homogeneous regions of
low contrast, more pixels are needed to estimate the de-noised pixel value. The reverse is the case for the regions of high contrast.

The local contrast between neighbouring pixel locations \(a\) and \(b\) is defined as:

\[
\rho_{a,b} = \frac{|I_n(a) - I_n(b)|}{|I_n(a) + I_n(b)|} \quad \text{Eq. (6.10)}
\]

Assuming that \(a\) is an arbitrary pixel location and \(b\) is the location of its neighbouring pixel within the image \(I_n(x)\), four images in each of the possible four directions can be computed to encode the local contrast. The final contrast image \(I_c(x)\) is then be computed as the average of the four directional contrast images. To link the decay parameter \(h\) to the contrast image the logarithm of the inverse of the contrast image \(I_{ic}(x) = \log \left[ I / I_c(x) \right] \) has been computed, where \(I\) is an identity matrix and the operator ‘/’ stands for the element-wise division. Now by mapping linearly the values of the inverted contrast image \(I_{ic}(x)\) to the values of the decay parameter \(h\), the spatial location function can be given as:

\[
h(x) = \left( \frac{I_{ic}(x) - I_{ic\text{min}}}{I_{ic\text{max}} - I_{ic\text{min}}} \right) \cdot h_{\text{max}} + h_{\text{min}} \quad \text{Eq.(6.11)}
\]

where,

- \(I_{ic\text{max}}\) denote the maximum value of the inverted contrast image \(I_{ic}(x)\)
- \(I_{ic\text{min}}\) denote the minimum value of the inverted contrast image \(I_{ic}(x)\)
- \(h_{\text{max}}\) is the target maximum value of the decay parameter \(h\)
- \(h_{\text{min}}\) is the target minimum value of the decay parameter \(h\)

### 6.4 Angle Distance Similarity Measure

The angle distance similarity measure has been applied to the SIFT Descriptors for matching. Let \(x = \{ x_1, x_2, \ldots, x_N \} \) be the training image vector and \(y = \{ y_1, y_2, \ldots, y_N \} \) be the testing image vector. Then the Angle Distance Similarity measure is given as:
\[ d(x,y) = \cos^{-1}(x.y), \] where, the operator . (dot) implies a dot product, and the Inverse Cosine of dot products of unit vectors is a close approximation to the ratio of Euclidean distances for small angles.

### 6.5 Proposed Algorithm for illumination invariance

The methodology discussed above is summarized below with a block diagram shown in Fig.6.2:

- Normalizing the images using the Adaptive N1 Normalization Procedure
- Extracting the SIFT features of the normalized images to form the feature vector set
- Performing supervised classification of feature vectors for face recognition using the Angle Distance Similarity Measure

![Fig.6.2 Block Diagram of the SIFT based algorithm for illumination invariance](image_url)
6.5.1 Pseudo-code of the proposed approach

Step 1
- Select a particular face image database and separate the images into training and testing sets
- Load the training images of the database

Step 2
- For the given image compute the local contrast $\rho_{a,b}$ using Eq.6.10 in four neighbouring directions to obtain four contrast images
- Take the average of all the four contrast images to obtain the final contrast image
- Compute the logarithm of the inverse of the contrast image
- Compute the spatial location function $h(x)$ using Eq.6.11
- Compute the weighting function $w(z, x)$ using Eq.6.9
- Compute the normalized image $I(x, y)$ using Eq.6.8
- Compute and store SIFT features of the normalized image $I(x, y)$ using Eq.6.1 to Eq.6.7

Step 3
- Repeat step 2 for all the training images

Step 4
- Take as input the test image

Step 5
- Perform step 2 for the given test image and store the feature vectors

Step 6
- Compare feature vectors obtained from step 5 with the stored feature vectors obtained in step 3 using angle distance similarity measure to identify the face image
Step 7

- Compare with the actual class of the test image to determine whether decision obtained from the proposed algorithm is correct or not and store the outcome.

Step 8

- Repeat step 4-7 for all the test images and compute the percentage of correct recognition.

Step 9

- Repeat Step 1-8 for the next face image database.

6.5.2 Experimental results

The proposed algorithm has been implemented for the Yale B Database as discussed in Chapter 3. All the images of the database have been cropped to a uniform size of 192 x 168 to minimize the storage capacity and computation time. The adaptive N1 means normalization technique has been then applied to the images. The SIFT Descriptors of the normalized images have been computed for feature extraction and the angle distance similarity measure has been used for classification.

The sample images of Yale B database [Fig. 6.3(a)], their corresponding normalized images obtained after application of Adaptive N1 Means Normalization technique [Fig. 6.3(b)] and the keypoints extracted after computation of SIFT Descriptors [Fig. 6.3(c)] are shown below.
The first image of each class of the Yale B database has been used for training and the remaining images used for testing. The SIFT feature matching of some sample images have been shown in Fig.6.4. In the figure a test image has been compared with 10 different training images. In all the 10 figures of Fig.6.4 the face image on the left is the normalized test image and the images on the right are the normalized training images. Maximum match is obtained with the correct class image even with severe variation of illumination.
From the results obtained and summarized in Table 6.1, it may be inferred that amongst all illumination normalization techniques, the Adaptive N1 Means Normalization provides much better results for illumination invariant face recognition when it is combined with SIFT Descriptors for illumination invariant face recognition.

Fig. 6.4 SIFT Feature Matching of the Yale Database
<table>
<thead>
<tr>
<th>Sl.No</th>
<th>Normalization Method used along with SIFT</th>
<th>Recognition %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Single Scale Retinex[116]</td>
<td>83</td>
</tr>
<tr>
<td>2</td>
<td>Multiscale Retinex[117]</td>
<td>85</td>
</tr>
<tr>
<td>3</td>
<td>Adaptive Single Scale Retinex[34]</td>
<td>87</td>
</tr>
<tr>
<td>4</td>
<td>Homomorphic Filtering[63]</td>
<td>82</td>
</tr>
<tr>
<td>5</td>
<td>Single Scale Self Quotient Image[73]</td>
<td>87</td>
</tr>
<tr>
<td>6</td>
<td>Multi Scale Self Quotient[73]</td>
<td>86</td>
</tr>
<tr>
<td>7</td>
<td>DCT Normalization[53]</td>
<td>82</td>
</tr>
<tr>
<td>8</td>
<td>Wavelet Normalization[62]</td>
<td>80</td>
</tr>
<tr>
<td>9</td>
<td>Wavelet Denoising[25]</td>
<td>81</td>
</tr>
<tr>
<td>10</td>
<td>N1 Means Normalization[24]</td>
<td>85</td>
</tr>
<tr>
<td>11</td>
<td>Adaptive N1 Means Normalization</td>
<td>94</td>
</tr>
</tbody>
</table>

Table 6.1. Results obtained and comparison with other normalization techniques

6.6 Proposed Algorithm for expression invariance

The steps of the proposed algorithm for expression invariance are summarized below along with a block diagram shown in Fig.6.5.

- Computing the 2D Discrete Wavelet Transform of the input images using coiflet functions.
- Combining the subbands viz; the approximation subband and three detail subbands of horizontal detail, vertical detail and diagonal detail.
- Extracting the SIFT descriptors of the fused wavelet coefficients to form the final feature vector set.
- Performing supervised classification of feature vectors for face recognition.
6.6.1 Pseudo-code of the proposed approach

Step 1
- Select a particular face image database and separate the images into training and testing sets
- Load the training images of the database

Step 2
- Perform the first level wavelet decomposition of the image, \( I(x,y) \) using Eq.3.8
- Take the linear combinations of the four sets of coefficients obtained in the above step
- Compute and store the SIFT features of the wavelet transformed image to form the final feature vector set of the given image using Eq.6.1 to Eq.6.7

Step 3
- Repeat step 2 for all the training images

Step 4
- Take as input the test image

Step 5
- Perform step 2 for the given test image and store the feature vectors
Step 6
- Compare feature vectors obtained from step 5 with the stored feature vectors obtained in step 2 using angle distance similarity measure to identify the face image

Step 7
- Compare with the actual class of the test image to determine whether decision obtained from the proposed algorithm is correct or not and store the outcome

Step 8
- Repeat step 4-7 for all the test images and compute the percentage of correct recognition.

Step 9
- Repeat Step 1-8 for the next face image database

6.6.2 Experimental results

The proposed algorithm has been implemented for the Essex Grimace Database consisting of images with variable expression, as discussed in Chapter 3. After application of DWT to the images, the size of the fused wavelet transform coefficient obtained is found to be 102 x 92. A sample image from the Essex Grimace Database, the corresponding Wavelet Transformed image and SIFT Descriptors extracted from the wavelet transformed image has been shown in Fig.6.6.
Keypoints extracted from the wavelet transformed images are represented by a feature vector of size 1x128. However, the feature set size will be different for each image as the number of keypoints extracted from each image is different. The average number of keypoints from each image is found to be approximately 50 and hence the average size of the feature descriptors is 50x128. The first image of each class of the database has been used for training and the remaining images used for testing. The SIFT feature matching of some sample images have been shown in Fig.6.7, wherein a test image has been compared with 10 different training images and maximum match is obtained with the correct class image despite variations in expression.
To ensure the contribution of DWT and SIFT in the face recognition performance some experiments have also been conducted on the Essex Grimace Database which has been described hereunder. The results of the analysis/experiments are summarized in Table 6.2.

**Experiment 1 (Only DWT)**

- Computing DWT of the training images taking a single image from each class for training and the rest for testing.
- Storing the DWT coefficients as final feature vector set
• Performing face recognition using the Angle Distance Similarity Measure

Experiment 2 (Only SIFT)

• Extracting the SIFT Descriptors of the Essex Grimace database images taking a single image for training from each class.
• Storing the SIFT Descriptors as final feature vector set
• Performing face recognition using the Angle Distance Similarity Measure

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Method</th>
<th>Recognition%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Proposed DWT SIFT approach</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>Experiment 1-Only DWT approach</td>
<td>88</td>
</tr>
<tr>
<td>3</td>
<td>Experiment 2-Only SIFT</td>
<td>90</td>
</tr>
<tr>
<td>6</td>
<td>Face Recognition using Curvelet based PCA[22]</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>(Uses 8 images for training)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Face Recognition by Curvelet based Feature extraction[41]</td>
<td>98.1</td>
</tr>
<tr>
<td></td>
<td>(Uses 12 images for training)</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2. Results obtained and comparison

6.7 Discussion

In this chapter, an attempt has been made to apply the SIFT descriptors in face recognition techniques. Two different algorithms have been developed. The first algorithm has been developed to tackle the problem of illumination variations in face images especially when
availability of gallery images per person is limited. Though SIFT descriptors have various essential properties like invariance to image scaling, rotation and partial occlusion the SIFT features are not completely illumination invariant. Accordingly a second algorithm viz; Adaptive N1 Means Normalization has also been used prior to calculation of SIFT descriptors for illumination normalization. For extracting expression invariant descriptors using SIFT first the wavelet coefficients of the images have been computed. Wavelet coefficients provide excellent constancy when the subject undergoes rotation. The coefficients pack maximum information from the images and discard the redundancies. Hench the SIFT descriptors extracted from the wavelet decomposed images provide excellent performance. The proposed approach has been compared with individual performance of DWT as feature extractor and individual performance of SIFT as feature extractor for the same database and same single training image per person. The results of the proposed approach clearly outperform the results of experiment 1 and experiment 2. The results have also been compared with a couple of algorithms based on Curvelets [22, 41]. In both the papers the authors have used Essex Grimace Database to test their algorithms. Though results provided by them are satisfactory but the number of training images required is quite large compared to the proposed approach in this chapter.
Chapter 7

Local Binary Patterns based

Face Recognition

7.1 Introduction

Local Binary Patterns (LBP) for its excellent capability of description of local texture have been widely used in texture classification and various other applications including face detection and recognition, background removal, region of interest description etc. [90]. However due its inherent weaknesses of lack of noise sensitivity and rotation invariance, several variants of LBP have been proposed to address the rotation invariance issues. It has been seen that the uniform local binary patterns usually denoted as $LBP_{u2}$ are statistically more robust and has produced better recognition results than conventional LBP in several applications [90]. The original rotation invariant LBP operator as denoted by $LBP_{rin2}$ has also been widely used [90]. Although these approaches have produced satisfactory results in rotation invariant texture classification they have some shortcomings. As each local descriptor such as the filter bank response is transformed to canonical representation independently, the relative distribution of different orientations is lost. Further, as the transformation needs to be performed for each texture patch, it must be computationally simple to keep the overall computational cost low. Ahonen et al [26] has addressed this issue by using the LBP HF (LBP high pass filter) descriptor first by computing a non-invariant LBP histogram over the whole region and then constructing
rotationally invariant features from the histogram. This resulted in the rotation invariance being
attained globally and the features being invariant to rotations of the whole input signal while
retaining the information about relative distribution of different orientations of uniform local
binary patterns.

In this chapter, a rotation invariant LBP feature descriptor has been proposed for rotation
invariant feature extraction by using DWT (Discrete Wavelet Transform) instead of using FT
(Fourier Transform). The advantage of using DWT lies in its ability of revealing the aspects of
trends, breakdown points, and discontinuities in higher derivatives and self-similarity. More
often DWT needs only a few coefficients in comparison to Fourier transform for approximation
of any given function and is able to separate the fine details in a signal.

LBP is primarily used for texture classification. However, as the edges also play an important
role in face recognition [8], in the proposed approach the edge features have also been
concatenated along with the texture features to improve the results of face recognition.

### 7.2 Local Binary Patterns

The Local Binary Patterns operator is computed in a local circular region by taking the
difference of the centre pixel with respect to its neighbours by using the following equations.

\[
LBP_{R,P} = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p
\]

Eq. (7.1)
where,

$g_c$ and $g_p$ denote the gray values of the central pixel and its neighbor

$p$ is the index of the neighbor

$R$ is the radius of the circular neighborhood and

$P$ is the number of the neighbors.

Now for the central pixel with coordinate $(x, y)$, the coordinates of uniformly spaced circular neighborhood can be obtained as $(x + R \cos(2\pi p/P), y - R \sin(2\pi p/P))$ for $p = 0, 1, 2, \ldots, P-1$. In case of non-integer values of the neighboring coordinate a bilinear interpolation is used for estimation of pixel value. Fig. 7.2 shows some examples of the LBP operator, where the notation $(P, R)$ denotes a neighborhood of $P$ sampling points on a circle of radius of $R$.

![Fig.7.1. The circular (8, 1), (16, 2) and (8, 2) neighborhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel](image)

The most fundamental structure from the LBP has been extracted by using the concept of uniform pattern or a local binary pattern in which the binary code contains at most two
transitions from 0 to 1 or vice versa. In uniform LBP mapping there is a separate output label for each uniform pattern and all the non-uniform patterns are assigned to a single label. Thus, the number of different output labels for mapping for patterns of \( P \) bits is \( P (P - 1) + 3 \). For instance, the uniform mapping for neighborhoods of 8 sampling points as shown in Fig.7.2 will produce 59 output labels. Similarly, for neighborhoods of 16 sampling points it would be 243 labels.

There are mainly two reasons for omitting the non-uniform patterns. Firstly, most of the local binary patterns in natural images are uniform. Hadid et al. [67] has shown that 90.6% of the patterns in the (8,1) neighborhood and 85.2% of the patterns in the (8,2) neighborhood are uniform in experiments with facial images. The second reason for considering uniform patterns is the statistical robustness. Use of uniform patterns instead of all the possible patterns has most often lead to better recognition in many applications, as they are more stable and less prone to noise. Accordingly, the number of possible LBP labels are significantly lowered by considering only uniform patterns leading to the reliable estimation of their distribution with fewer samples.

The original rotation invariant LBP (LBP\textsuperscript{riu}) is achieved by circularly rotating each bit pattern to the minimum value. For instance, the bit sequences 1000011, 1110000 and 0011100 arising out of different rotations of the same local pattern correspond to the normalized sequence 0000111. Ahonen et al [26] has proposed a class of features that are invariant to rotation of the input image in the sense that such features computed along the input histogram rows are invariant to cyclic shifts. The Discrete Fourier Transform as given in equation (3) below has been used to construct these features where \( H(n,u) \) is the DFT of \( n \)th row of the histogram \( h_I (U_P (n, r)) \), where the histogram value \( h_I \) at bin \( U_P (n, r) \) is the number of occurrences of uniform pattern \( U_P (n, r) \) in image \( I \), \( n \) is the number of 1 bits in the pattern and \( r \) is the rotation of the pattern.
Fig. 7.2. The 58 different uniform patterns in (8,R) neighborhood

\[ H(n, u) = \sum_{r=0}^{P-1} h_I(U_P(n, r)) e^{-i2\pi ur/P}. \]  
Eq. (7.3)
7.3 LBPDWT Features

In the proposed work a modified LBPDWT feature extraction method has been proposed by using the Wavelet Transforms in place of Fourier Transform used by Timo Ahonen et al [26]. The following steps are followed.

- Calculate the Uniform LBP Histograms $h_I(U_P(n,r))$ of the image, where the histogram value $h_I$ at bin $U_P(n, r)$ is the frequency of uniform pattern $U_P(n, r)$ in image I.
- Compute the Discrete Wavelet Transform of the $n^{th}$ row of the histogram $h_I(U_P(n,r))$ to get the approximation and detail coefficients.

\[
\begin{align*}
W_\varphi(j_0, k) &= \frac{1}{\sqrt{M}} \sum_x f(x)\varphi_{j_0, k}(x) \\
W_\psi(j, k) &= \frac{1}{\sqrt{M}} \sum_x f(x)\psi_{j, k}(x)
\end{align*}
\]

where,

$W_\varphi(j_0, k)$ are the approximation coefficients and

$W_\psi(j, k)$ are the detail coefficients

$f(x)$ denote the $n^{th}$ row of the histogram $h_I(U_P(n,r))$, $x=1,2,\ldots,M$

$\varphi(x)$ denotes the scaling function and

$\psi(x)$ denotes the wavelet function.

$j$ determines the width of $\varphi_{j,k}(x)$ along the x axis and

$k$ determines the position of $\varphi_{j,k}(x)$

DWT coefficients are actually computed using the fast discrete wavelet transform algorithm as shown below:
\[ W_\phi (j_0, k) = h_\phi (-n)^* f(x) \]
\[ W_\psi (j_0, k) = h_\psi (-n)^* f(x) \]  
\[ \text{Eq. (7.5)} \]

where,

\( h_\phi (n) \) and \( h_\psi (n) \) are the scaling and wavelet vectors respectively.

\( W_\phi (j_0, k) \) and \( W_\psi (j, k) \) are the approximation and detail coefficients obtained by convolving \( h_1 ( U_P (n,r) ) \) with \( h_\phi (-n) \) and \( h_\psi (-n) \) respectively and subsampling the convolved results by a factor of 2.

The filter coefficients of the low pass and high pass filters for the coiflet basis function have been shown in Fig.7.3

- Obtain the feature vector set for the given image by linear summation of the approximation and detail coefficients for each row of the histograms.

\[ \text{i.e } W_1 (j, k) = W_\psi (j, k) + W_\phi (j_0, k) \]  
\[ \text{Eq.(7.6)} \]
Fig. 7.3 Filter coefficients of coiflet basis filters

### 7.4 Proposed Algorithm

The proposed LBP based face recognition algorithm has been described below along with a block diagram of the same in Fig. 7.4.

- Partitioning the image into non-overlapping blocks of equal size
- Computing the LBP histogram (H1) for each block
- Computing the Edge features of the blocks using Canny Edge Detection algorithm
- Computing the LBP histogram (H2) for each edge block
- Concatenate H1 for all blocks of the image and then compute the DWT to get the LBPDWT features (F1) of the given image
• Concatenate H2 for all the blocks of the image and then compute the DWT to get the LBPDWT features (F2) of the edge image

• Concatenate F1 and F2 to get the final feature vector set of the image

• Classification done using the Euclidean Distance Measure.

Fig. 7.4 Block diagram of the proposed LBPDWT based approach
7.5 Pseudo-code of the proposed approach

Step 1

- Select a particular face image database and separate the images into training and testing sets
- Load the training images of the database

Step 2

- Partition the given image into non overlapping blocks of equal size
- Compute the LBP features of the input block using Eq.7.1 and Eq.7.2
- Compute the histograms H1 of the features obtained
- Concatenate H1 for all image blocks and then compute the DWT of the concatenated vector using Eq.7.5 and Eq.7.6 to obtain the LBPDWT (F1) features of the image
- Compute the edge features using canny edge detection algorithm for each block
- Compute the LBP features of the edge detected block using Eq.7.1 and Eq.7.2
- Compute the histograms H2 of the features obtained
- Concatenate H2 for all image blocks and then compute the DWT of the concatenated vector using Eq.7.5 and Eq.7.6 to obtain the LBPDWT (F2) features of the edge image
- Concatenate F1 and F2 to obtain the final feature vector set

Step 3

- Repeat step2 for all the training images and store the feature vectors

Step 4

- Take as input the test image

Step 5

- Perform step 2 for the given test image and store the feature vectors
Step 6
- Compare feature vectors obtained from step 5 with the stored feature vectors obtained in step 3 using angle distance similarity measure to identify the face image.

Step 7
- Compare with the actual class of the test image to determine whether decision obtained from the proposed algorithm is correct or not and store the outcome.

Step 8
- Repeat step 4-7 for all the test images and compute the percentage of correct recognition.

Step 10
- Repeat Step 1-8 for the next face image database.

7.6 Experimental results

Several experiments have been carried out using the ORL, Yale and Essex Grimace Databases and the results of LBPDWT features are compared with those of LBP^u^, LBP^riu^ and LBPHF features. To verify whether the LBPDWT features are able to retain the discriminative nature of the original LBP histograms, the experiments have also been done with the KTH-TIPS2 database [142] of texture images used for material categorization.

In the first experiment the LBP^u^ (Uniform LBP), LBP^riu^ (Uniform Rotation Invariant LBP), LBP^ri^ (Rotation Invariant LBP), LBPHF (DFT based LBP) and LBPDWT (Wavelet based LBP) features, of the images as a whole have been computed. For each database one image has been used for training and remaining for testing. The number of the neighbours, P = 8 and, the radius of the circular neighbourhood, R = 1 have been considered. Chi Square Distance metric
Classification has been used for the LBP\textsuperscript{u2} and LBP\textsuperscript{riu2} features due to the sparse nature of the histograms. Whereas, the Euclidean distance measure has been used for LBPHF and LBPDWT features.

<table>
<thead>
<tr>
<th>Database Method</th>
<th>Recognition %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ORL</td>
</tr>
<tr>
<td>LBP\textsuperscript{u2}</td>
<td>82</td>
</tr>
<tr>
<td>LBP\textsuperscript{ri}</td>
<td>62</td>
</tr>
<tr>
<td>LBP\textsuperscript{riu2}</td>
<td>57</td>
</tr>
<tr>
<td>LBPHF</td>
<td>76</td>
</tr>
<tr>
<td>LBPDWT</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 7.1 Results of recognition and material categorization for images taken as a whole (P=8)

The results obtained in Table 7.1 above show that the Face Recognition using LBP provides better results when the images are partitioned into blocks. The LBP histograms contain information about distribution of local micro patterns such as edges, spots and flat areas over the whole image. For efficient facial representation feature extracted should also retain spatial information. Hence dividing the image into blocks provides better recognition results.

The images of the ORL Database are resized to 112x112 and are divided into blocks of equal size. For this experiment P = 8 and R = 1 have been taken. Optimum block size of 28 x 28 has been selected based on the results in Table 7.2. The blocks obtained represent different parts of the face image like eyes nose mouth etc. which are essential for face recognition. So the size of the blocks should be sufficient to include these features and will depend on the size of the face image. So the optimum block size for the face images has been determined experimentally.
<table>
<thead>
<tr>
<th>Block Size</th>
<th>Recognition %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>28x28</td>
</tr>
<tr>
<td>LBPHF</td>
<td>84</td>
</tr>
<tr>
<td>LBPDWT</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 7.2 Results for optimum block size selection for ORL Database with P=8 and R=1

Accordingly, the tests have been carried out for various values of P and R for block size 28 x 28. The values of P = 16 and R = 1 have been selected based on the results obtained in Table 7.3 Increasing the density of the circular neighbourhood will better capture the finer details of the image blocks.

<table>
<thead>
<tr>
<th>P R values</th>
<th>Recognition %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P = 16, R = 1</td>
</tr>
<tr>
<td>LBPHF</td>
<td>87</td>
</tr>
<tr>
<td>LBPDWT</td>
<td>88</td>
</tr>
</tbody>
</table>

Table 7.3 Results for selection of P and R values for a block size of 28x28 for ORL Database

As the edges play an important role in face image recognition, the LBP features of the Canny edge detected images of the ORL database have been computed and the recognition percentage have been obtained by diving the images into blocks of size 28x28. The edges of the images have been computed using the Canny Edge Detection algorithm and the LBP features have been
computed for P = 16 and R = 1 for the ORL database. The results obtained have been shown in Table 7.4. However it is clear from the results that considering only edges of the image will not provide sufficiently good results because the texture information has not been considered.

<table>
<thead>
<tr>
<th>SL.No.</th>
<th>Method</th>
<th>Recognition %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LBP^{u2}</td>
<td>79</td>
</tr>
<tr>
<td>2</td>
<td>LBP^{r}</td>
<td>86</td>
</tr>
<tr>
<td>3</td>
<td>LBP^{r,2}</td>
<td>79</td>
</tr>
<tr>
<td>4</td>
<td>LBPHF</td>
<td>86</td>
</tr>
<tr>
<td>5</td>
<td>LBPDWT</td>
<td>87</td>
</tr>
</tbody>
</table>

Table 7.4 Results for LBP Features computed on the Edge Detected Image for ORL Database

To ensure that the proposed face recognition algorithm considers both edge features as well as texture features for determination of a face image’s identity, the block wise concatenation of LBP features and LBP features of Canny edge detected images have been performed. Images of all the three databases have been resized to 112 x 112 and are divided into blocks of size 28 x 28. The LBP features have been computed for the blocks as well as the Canny edge detected blocks and the resultant histograms have been concatenated to obtain the feature vector set of the images. Taking one image for training and the remaining images for testing, the recognition percentages have been computed as shown in Table 7.5.
<table>
<thead>
<tr>
<th>Method</th>
<th>ORL</th>
<th>Yale</th>
<th>Essex Grimace</th>
<th>Feature Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBP&lt;sub&gt;u2&lt;/sub&gt;</td>
<td>95</td>
<td>89</td>
<td>99</td>
<td>243<em>16</em>2=7776</td>
</tr>
<tr>
<td>LBP&lt;sup&gt;i&lt;/sup&gt;</td>
<td>88</td>
<td>85</td>
<td>98</td>
<td>4116<em>16</em>2=131712</td>
</tr>
<tr>
<td>LBP&lt;sup&gt;rln2&lt;/sup&gt;</td>
<td>91</td>
<td>85</td>
<td>98</td>
<td>18<em>16</em>2=576</td>
</tr>
<tr>
<td>LBPHF</td>
<td>90</td>
<td>87</td>
<td>99</td>
<td>243<em>16</em>2=7776</td>
</tr>
<tr>
<td>LBPDWT</td>
<td>96</td>
<td>90</td>
<td>100</td>
<td>138<em>16</em>2=4416</td>
</tr>
</tbody>
</table>

**Table 7.5 Recognition Results for the proposed algorithm**

The proposed LBPDWT based algorithm has also been tested on images with partial occlusion and has provided satisfactory results. The gallery images used for the test have been shown in Fig.7.5. Three images from the gallery have been modified as shown in Fig.7.6 and have been used for testing. All the test images have been correctly classified by the proposed algorithm and the corresponding gallery images have been returned by the algorithm as shown in Fig.7.7 below.

![Fig.7.5 Training images used for occlusion test](image-url)
From the results obtained above, it can be seen that performance of the proposed LBPDWT feature extraction algorithm is the best in terms of recognition percentage. It can be seen from Table 7.5 that the LBP uniform pattern features also perform almost equally well in terms of recognition percentage. However, the size of the feature vectors obtained is significantly reduced when the LBPDWT features have been used. The $LBP_{riu2}$ feature set has the smallest feature vector size; however, the recognition percentage it provides is not as good as the proposed approach. So the proposed LBPDWT feature size and recognition percentage is optimum when compared to the other LBP features. The algorithm also shows satisfactory performance for partially occluded images. There is further scope to improve on the proposed work by reducing the size of the feature vector using some techniques of dimensionality reduction.