Table 4.4 Overall Performance of WHT

<table>
<thead>
<tr>
<th>Method</th>
<th>Full image transform</th>
<th>Row Feature Vector</th>
<th>Column Feature Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Local</td>
<td>ORL</td>
<td>Local</td>
</tr>
<tr>
<td><strong>Accuracy (90%)</strong></td>
<td>60%</td>
<td>66%</td>
<td>90%</td>
</tr>
<tr>
<td><strong>%Occlusion withstand</strong></td>
<td>20%x20%</td>
<td>20%x20%</td>
<td>50%x50%</td>
</tr>
<tr>
<td><strong>Speckle Noise</strong></td>
<td>20%</td>
<td>20%</td>
<td>50%</td>
</tr>
<tr>
<td><strong>Gaussian</strong></td>
<td>1%</td>
<td>1%</td>
<td>1%</td>
</tr>
<tr>
<td><strong>Salt &amp; Pepper</strong></td>
<td>20%</td>
<td>10%</td>
<td>30%</td>
</tr>
<tr>
<td><strong>FAR</strong></td>
<td>15%</td>
<td>13%</td>
<td>7%</td>
</tr>
<tr>
<td><strong>FRR</strong></td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

The Table 4.4 shows overall performance of Walsh Hadamard Transform (WHT) method. The ORL database due to its controlled nature performs better than local database. The Column feature vector method and Row feature vector methods give remarkably good performance compared to full transform. FAR for full image transform is 15% for local database and 13% for ORL database, RV gives 7% FAR but for CV method FAR is zero for both databases. FRR is zero for all the approaches.

### 4.6 Slant Transform (ST)

The concept of an orthogonal transformation containing slant basis vector was introduced by Enomoto and Shibata [11]. The slant transform has its first basis function as constant and second basis function as linear slant line. The Slant vector is a discrete saw tooth waveform decreasing in uniform steps over its length. It has been seen that Slant vectors are suitable for efficiently representing gradual brightness change in a face image line.
4.6.1 Slant Matrix Construction:

If $S(n)$ denotes the $N \times N$ Slant matrix ($N=2^n$), then

$$S(2) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$  \hspace{1cm} 4.10

The Slant matrix for $N=4$ can be written as

$$S(4) = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ a+b & a-b & -a+b & -a-b \\ 1 & -1 & -1 & 1 \\ a-b & -a-b & a+b & -a+b \end{bmatrix}$$  \hspace{1cm} 4.11

where $a$ and $b$ is real constants to be determined subject to the following conditions:

1) step size must be uniform

2) $S(4)$ must be orthogonal

The value of $a$ and $b$ are calculated as follows:

1) Step size between first two element of 2\textsuperscript{nd} row of $S(4)$ is

$$(a+b)-(a-b)=2b$$

2) The step size between the 2\textsuperscript{nd} and 3\textsuperscript{rd} element of 2\textsuperscript{nd} row of $S(4)$

$$(a_b)-(a+b)=2a-2b$$

3) The step size should be uniform so,

$$2b=2a-2b$$

$$a=2b$$

so now $S(4)$ can be written as,
The value of \( a \) and \( b \) can be calculated from orthogonality condition

\[
\left\{ \frac{1}{\sqrt{4}} \begin{bmatrix} 3b & b & -b & -3b \\ b & -b & 3b & 3b \\ b & b & -3b & b \\ -b & -b & -3b & b \end{bmatrix} \right\} \cdot \left\{ \frac{1}{\sqrt{4}} \begin{bmatrix} 3b & b & -b & -3b \end{bmatrix}^T \right\} = 1
\]

\[
\therefore \quad b = \frac{1}{\sqrt{5}}
\]

\[
\therefore \quad a = 2b
\]

\[
\therefore \quad a = \frac{2}{\sqrt{5}}
\]

So, \( S(4) \) is given as

\[
S(4) = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}
\]

This is 4x4 normalized slant matrix. The \( S(4) \) can be express in terms of \( S(2) \) such that,
\[ S(4) = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ a_4 & b_4 & -a_4 & b_4 \\ 0 & 1 & 0 & -1 \\ -b_4 & a_4 & b_4 & a_4 \end{bmatrix} \begin{bmatrix} S_2 & 0_2 \\ 0_2 & S_2 \end{bmatrix} \]  
\[ \text{Here } a_4 = \frac{2}{\sqrt{5}} ; \quad b_4 = \frac{1}{\sqrt{5}} \text{ and } 0_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \]

In a similar manner, the relation between \( s(4) \) and \( S(8) \) can be given as

\[ S(8) = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ a_8 & b_8 & 0 & 0 & -a_8 & b_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -b_8 & a & 0 & 0 & b_8 & a_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_4 & 0_4 \\ 0_4 & S_4 \end{bmatrix} \]

A generalized way to obtained a Slant matrix of order \( N \) in terms of Slant matrix of order \( N/2 \) is shown below,
1. The Slant transform is real and orthonormal transform

\[ S = S^*, \quad S^{-1} = S^T \]

2. It is a fast algorithm reducing the complexity to

\[ O(N \log_2 N) \] for N×1 vector
3. It is very good in energy compaction for facial images. Very few coefficients are sufficient for recognizing the stored image of face in database.

4.6.2 Result Analysis and Discussion

The proposed algorithm of taking 1D Slant Transform row wise and column wise separately is tested for ORL and local database. Following graph shows that as percentage of energy in image for Row Feature Vector, Column Feature Vector and even for entire image is increased the percentage accuracy also increases.

Fig.4.37 Percentage accuracy in Full Transformed image, Row Feature vector and Column Feature vector for different percentage of energy in local database for ST.
Fig. 4.38 Percentage accuracy in Full Transformed image, Row Feature vector and Column Feature vector for different percentage of energy in ORL database for ST.

**Observation:** The graphs (Fig 4.37-Fig 38) show that as percentage image energy increases the Row feature Vectors and Column Feature Vectors gives better accuracy than the full transformed image in local as well as ORL database.

The robustness of algorithm is checked for the different percentage of occlusion and various types of noise like Speckle Noise, Gaussian noise, Salt and Pepper noise introduced on the test image. Following Fig. 4.39, Fig. 4.40 shows the results for occlusion applied on test images for Local and ORL database.
Fig. 4.39 Graph of accuracy for different percentage of occlusion introduced as %length X %width of image for Local database in ST at 90% energy.

Fig. 4.40 Graph of accuracy for different percentage of occlusion introduced as %length X %width of image for ORL database in ST at 90% energy.

Observation: The graphs (Fig 39-Fig 4.40) show that as occlusion on image increases the accuracy reduces. The full transform efficiency is very less compared to row and column feature vector method.
The algorithm is also tested for various types of noise like Speckle Noise, Gaussian noise, Salt and pepper Noise. Following Graphs shows the performance of algorithm for these noise.

**Speckle Noise:**

Fig 4.41 Performance of ST in presence of a Speckle Noise for local database at 90% energy.

Fig 4.42 Performance of ST in presence of a Speckle Noise for ORL database at 90% energy.
**Observation:** It has been observe clearly from graphs (Fig 4.41-Fig 4.42) that Full image transform withstands Speckle Noise more robustly than row and column feature method both databases. After 10% of noise addition the accuracy drops down to less than 50%.

**Gaussian Noise:**

![Gaussian Noise Graph](image1)

Fig 4.43 Performance of ST in presence of a Gaussian noise for local database at 90% energy.

![Gaussian Noise Graph](image2)

Fig 4.44 Performance of ST in presence of a Gaussian noise for ORL database at 90% energy.
**Observation:** The graphs give an idea about the variation of a Gaussian noise and its effect on accuracy for both databases. It is clear from the graphs that ORL database withstands Gaussian noise up to 5% but the accuracy at this level is below 50%. The local database can withstand the Gaussian noise up to 5% in column feature vector method. For other methods at this level of noise addition accuracy drops below 50%.

**Salt and Pepper Noise:**

![Graph showing performance of ST in presence of Salt and Pepper noise](image)

**Fig 4.45** Performance of ST in presence of a Salt and Pepper noise for local database at 90% energy.
Fig 4.46 Performance of ST in presence of a Salt and Pepper noise for ORL database at 90% energy.

**Observation:** The graph shows that as Salt and pepper noise in image increases the accuracy reduces. For 10% addition of noise in local database drops down the accuracy from more than 80% to less than 20%. ORL database gives better result compared to local database but after 20% noise addition it also becomes unreliable.

**Table 4.5** is overall performance table for the Slant Transform (ST). The data given for accuracy of algorithm is for 90% image energy. Under occlusion and noise addition conditions the cutoff point is 50% accuracy below which algorithm is not considered.
Table 4.5 Overall Performance of ST

<table>
<thead>
<tr>
<th>Method</th>
<th>Full image transform</th>
<th>Row feature vector</th>
<th>Column feature vector</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local</td>
<td>ORL</td>
<td>Local</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Accuracy (90%)</strong></td>
<td>60</td>
<td>60</td>
<td>80</td>
</tr>
<tr>
<td><strong>%Occlusion withstand</strong></td>
<td>0%X0%</td>
<td>0%X0%</td>
<td>50%X50%</td>
</tr>
<tr>
<td><strong>Speckle Noise</strong></td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td><strong>Gaussian</strong></td>
<td>0%</td>
<td>1%</td>
<td>1%</td>
</tr>
<tr>
<td><strong>Salt &amp; Pepper</strong></td>
<td>0%</td>
<td>10%</td>
<td>0%</td>
</tr>
<tr>
<td><strong>FAR</strong></td>
<td>70%</td>
<td>68%</td>
<td>10%</td>
</tr>
<tr>
<td><strong>FRR</strong></td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

The Table 4.4 shows overall performance of Slant Transform (ST) method. The ORL database due to its controlled nature performs better than local database. ST method is very sensitive to the noise and performs poorly in case of noise addition. The FAR of full image transform is very high compared to RV and CV approaches. FRR is zero for all methods.

4.7 Kekre’s Transform

Kekre’s Transform is newly introduced, real and an orthogonal transform. This is a non sinusoidal transform and generated from a Kekre’s matrix. The detail procedure for calculating the Kekre’s Matrix is shown in following section.
4.7.1 Kekre’s Matrix Construction

Let us generate the Kekre’s Matrix \([K]\) for size \(mxm\) where \(m\) can be any integer not necessarily the power of 2 as required for many other conventional transforms. This matrix has all 1’s on the main diagonal and upper triangle of the matrix. The sub diagonal just below the main diagonal has the value \((-m+i)\) where ‘\(m\)’ is the order of matrix and ‘\(i\)’ is the column number. Rest of the elements of lower triangle below the sub diagonal is all zeros. The general form of Kekre’s matrix \([K]\) can be written as

\[
[K] = \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 & 1 \\
-m+1 & 1 & 1 & \cdots & 1 & 1 \\
0 & -m+2 & 1 & \cdots & 1 & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & 1 \\
0 & 0 & 0 & \cdots & -1 & 1
\end{bmatrix}
\]

4.21

The properties of Kekre’s Transform are as follows:

1) The Kekre’s transform is real and orthogonal transform.

\[
[K] [K]^{T} = [\mu]
\]

4.22

Where \([K]^{T}\) is transpose of \([K]\) and \([\mu]\) is a diagonal matrix and its elements are given by
\( \mu_{ii} = m \) \hspace{1cm} 4.23

\( \mu_{ij} = (m-i) \cdot (m-i+1) \)

2) It has a fast algorithm as it contains \( m(m+1)/2 \) number of ones and \( (m-1)(m-2)/2 \) number of zeros leaving only \( (m-1) \) integer multiplications and only \( (m-1)(m/2) \) additions for transforming a column vector of dimension \( mx1 \). For a normal matrix transformation we require \( m^2 \) multiplications and \( m(m-1) \) additions.

3) The transform of a vector \( f \) is given by

\[
F = [K] \star f
\]

4) For image \([I]\) the transform \([A]\) is calculated as

\[
[A] = [K][I][K]^T
\]

And inverse transform is given by

\[
[I] = [K]^T \begin{bmatrix} A \\ \mu \end{bmatrix} \begin{bmatrix} K & \mu \\ \mu & \mu \end{bmatrix}^{-1}
\]

Where \( \begin{bmatrix} A \\ \mu \end{bmatrix}_{ij} = a_{ij} \) and \( \mu_{ij} = \mu_{ii} \cdot \mu_{jj} \)
4.7.2 Proposed Algorithm

- Select the image.
- Select one column at a time from the image.
- Multiply by \([K]\) matrix to selected column and repeat the procedure for all columns giving intermediate matrix.
- Intermediate matrix is then multiplied with \([K]^T\) to generate full transformed image.
- Selection of number of coefficients for different percentage of image energy.
- Compare transformed coefficients with stored database coefficients using Euclidean distance as a similarity measure.
- If Euclidean distance is within the pre decided threshold value than recognize the image.
- Threshold =10 gives acceptable results.

In the recognition process the comparison is done in the transform domain only. This reduces computational complexity and time considerably to make it effective for real time applications.

4.7.3 Result Analysis and Discussion

In this particular section the performance of newly suggested Kekre’s transform is compared for full image transform, Row feature vector method and column feature vector method. The proposed algorithm of taking 1D transform row wise and column wise separately is tested for ORL and local database. Following graph shows that as percentage of energy in image for Row Feature Vector, Column Feature Vector and even for entire image is increased the percentage accuracy for face recognition also increases.
Fig. 4.47 Percentage accuracy in Full Transformed image, Row Feature vector and Column Feature vector for different percentage of energy in local database for Kekre’s Transform.

Fig. 4.48 Graph of percentage accuracy in Full Transformed image, Row Feature vector and Column Feature vector for different percentage of energy in ORL database for Kekre’s Transform.

**Observation:** The graphs (Fig. 4.47-Fig. 4.48) show that as percentage image energy increases the accuracy also improves. The
full transform performs better than row or column feature vector method.

The robustness of algorithm is checked for the different percentage of occlusion and various types of noise like Speckle Noise, Gaussian noise, Salt and Pepper noise introduced on the test image. Following Fig. 4.49, Fig. 4.50 shows the results for occlusion applied on test images for Local and ORL database.

![Graph of accuracy for different percentage of occlusion introduced as %length X %width in image for Local database in Kekre’s Transform at 90% energy.](image)

Fig.4.49 Graph of accuracy for different percentage of occlusion introduced as %length X %width in image for Local database in Kekre’s Transform at 90% energy.
Fig.4.50 Graph of accuracy for different percentage of occlusion introduced as %length X %width in image for ORL database in Kekre’s Transform at 90% energy.

**Observation:** The graphs show that as occlusion on image increases the accuracy reduces. Till 50% X 50% of occlusion the algorithm gives accuracy above 60% for both the databases in all three techniques. Full transform on image gives slightly better performance than row or column feature vector method.

The algorithm is also tested for various types of noise like Speckle Noise, Gaussian noise, Salt and pepper Noise. Following graphs (Fig 4.51-Fig 4.52) shows the performance of algorithm for these noises.
Speckle Noise:

Fig 4.51 Performance of Kekre’s transform in presence of a Speckle Noise for local database at 90% energy.

Fig 4.52 Performance of Kekre’s Transform in presence of a Speckle Noise for ORL database at 90% energy.
**Observation:** The full image transform gives better accuracy for face recognition under the noisy condition also. After 30% of noise addition the results become unreliable in full image transform domain. The Row and Column feature vector method cannot withstand the noise noticeably.

**Gaussian Noise:**

![Gaussian Noise Chart](image)

Fig 4.53 Performance of Kekre’s Transform in presence of a Gaussian noise for local database at 90% energy.
Fig 4.54 Performance of Kekre’s Transform in presence of a Gaussian noise for ORL database at 90% energy.

**Observation:** The graphs give an idea about the variation of a Gaussian noise and its effect on accuracy for both databases. It is clear from the graphs that ORL database withstands Gaussian noise up to 5% which is better than local database where only 1% noise addition is advisable after that the accuracy falls down to 0%. The row and column feature vectors cannot withstand the noise addition.
Salt and Pepper Noise:

Fig 4.55 Performance of Kekre’s Transform in presence of a Salt and Pepper noise for local database at 90% energy.

Fig 4.56 Performance of Kekre’s Transform in presence of a Salt and Pepper noise for ORL database at 90% energy.
Observation: The graphs (Fig 4.55-Fig 4.56) shows that as Salt and pepper noise in image increases the accuracy reduces and after 50% of noise addition the algorithm becomes unreliable. The full transform of image supports noise addition more robustly than individual row and column feature vector method.

Table 4.6 is a overall performance table for the Kekre’s Transform. The data given for accuracy of algorithm is for 90% image energy. Under occlusion and noise addition conditions accuracy below 50% is not considered.

### Table 4.6 Overall Performance of Kekre’s Transform

<table>
<thead>
<tr>
<th>Method</th>
<th>Full image transform</th>
<th>Row Feature Vector</th>
<th>Column Feature Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Local</td>
<td>ORL</td>
<td>Local</td>
</tr>
<tr>
<td>Accuracy (90%)</td>
<td>90%</td>
<td>93%</td>
<td>73%</td>
</tr>
<tr>
<td>%Occlusion withstand</td>
<td>50%X50%</td>
<td>50%X50%</td>
<td>50%X50%</td>
</tr>
<tr>
<td>Speckle Noise</td>
<td>30%</td>
<td>20%</td>
<td>0%</td>
</tr>
<tr>
<td>Gaussian</td>
<td>1%</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td>50%</td>
<td>50%</td>
<td>0%</td>
</tr>
<tr>
<td>FAR</td>
<td>0%</td>
<td>0%</td>
<td>10%</td>
</tr>
<tr>
<td>FRR</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

The Table 4.6 shows overall performance of Kekre’s Transform method. The ORL database due to its controlled nature performs better than local database. When transform is applied on full image it gives better result overall compare to Row feature vector and Column feature vector techniques. RV and CV methods fails for the salt and Pepper noise addition but Full transform can give accuracy more than 50% for noise addition till 50%. FAR is higher for RV and CV methods compared to Full image transform. FRR is zero for all approaches.
4.8 The Haar Transform

The Haar Transform is derived from the Haar Matrix. The Haar Transform in general be expressed as

\[ \text{Transformed Image} = H \ast I \ast H' \]

4.27

Where \( I \) is a \( N \times N \) image, \( H \) is a \( N \times N \) Haar transformation matrix. The matrix \( H \) contains the Haar Basis functions which are defined over a continuous closed interval \([0,1]\). The derivation of Haar basis function in general is defined as

\[
h_{00} \left( x \right) = \frac{1}{\sqrt{N}} \quad x \in [0,1]
\]

and

\[
h_{pq} \left( x \right) = \begin{cases} 
\frac{1}{\sqrt{N}} & 2^{p/2} \leq x \leq \frac{q-0.5}{2^p} \\
-\frac{1}{\sqrt{N}} & \frac{q-0.5}{2^p} \leq x \leq \frac{q}{2^p} \\
0 & \text{otherwise}
\end{cases}
\]

4.28

where \( 0 \leq p \leq \log_2 N \) and \( 1 \leq q \leq 2^p \).
Haar matrix for size 4x4 and 8x8 is shown here,

\[
H_4 = \frac{1}{\sqrt{4}} \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & -1 & -1 \\
\sqrt{2} & -\sqrt{2} & 0 & 0 \\
0 & 0 & \sqrt{2} & -\sqrt{2}
\end{bmatrix}
\]  

4.29(a)

\[
H_8 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
\sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\
2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & -2
\end{bmatrix}
\]  

4.29(b)

The important characteristics of Haar basis is that it varies both in width and position. This gives Haar transform a dual scale position nature. So Haar transform can be used in development of Wavelet Transform for which the base is a mutiresolution analysis.

**4.8.1 Wavelet Transform**

The beginnings of Wavelet transform as a specialized field can be traced to the work of Grossman and Morlet [114]. Major contributors in development of field of Wavelet transform are Meyer [115] and Daubechies[90].

A wave is an oscillating function of time or space that is periodic. The wave is an infinite length continuous function in time or space. In contrast wavelets are localized waves. A wavelet is a waveform of an effectively limited duration that has an average value of zero.
The function $\Psi(x)$ is said to be wavelet if it posses the following properties.

1. The function integrates to zero, or equivalently its Fourier transform denoted as $\Psi(\omega)$ is zero at the origin.

$$\int_{-\infty}^{\infty} \Psi(x)dx = 0 \quad 4.30$$

The above equation suggest that the function is either oscillatory or has a wave appearance.

2. It is square integrable or equivalently has finite energy.

$$\int_{-\infty}^{\infty} |\Psi(x)|^2dx < \infty \quad 4.31$$

Here it implies that most of energy in $\Psi(x)$ is confined to a finite interval so $\Psi(x)$ has good space localization. So ideally function has a zero value outside finite interval.

3. The Fourier transform must satisfy the admissibility condition given by

$$C_\Psi = \int_{-\infty}^{\infty} \frac{\Psi(\omega)^2}{|\omega|}d\omega < \infty \quad 4.32$$

This equation is useful in formulating the inverse wavelet transform. $\Psi(\omega)$ must have sufficient decay in frequency. So wavelet has a band pass characteristic. Thus a wavelet is a small wave that exhibits good time frequency localization.
A family of wavelets can be generated by dilating and translating the mother wavelet \( \Psi(x) \) which is given by

\[
\Psi_{(a,b)}(x) = \frac{1}{\sqrt{a}} \Psi\left(\frac{x-b}{a}\right)
\]

Here, ‘a’ is a scale parameter and b is the shift parameter.

The wavelet transform provides time frequency representation of a signal. It is designed to overcome the shortcomings of the Short Time Fourier Transform (STFT). In STFT the resolution is constant for all the frequencies. In Wavelet transform multi resolution technique is used by which different frequencies are analyzed with different resolutions.

**4.8.2 Discrete Wavelet Transform (DWT)**

The Discrete Wavelet Transform is obtained by filtering the signal through a series of digital filters at different scales. The scaling operation is done by changing the resolution of signal by process of subsampling.

The DWT can be computed by decomposing the input sequence into low pass and high pass sub bands each consisting of half the number of samples in the original sequence. Sub band coding is a procedure in which input signal is subdivided into several frequency bands. Sub band coding can be implemented through a filter bank.

Relationship Between wavelet and filter:
If the space \( V_0 \) and \( W_0 \) are subspace of \( V_1 \) then scaling function \( \Phi(x) \) and wavelet function \( \Psi(x) \) can be expressed in terms of the basis function of \( V_1 \) as ,
\[ \Phi(x) = 2 \sum_k h_k \Phi(2x-k) \]
\[ \Psi(x) = 2 \sum_k g_k \Phi(2x-k) \]

Here \( h_k \) and \( g_k \) are the filter coefficients that uniquely define the scaling function and wavelet function respectively.

In DWT an image signal can be analyzed by passing it through an analysis filter bank followed by decimation operation.

Fig 4.57 Different filter banks for down sampling the original image

As shown in above diagram the analysis filter bank consists of a low pass and high pass filter at each decomposition stage. When a
signal passes through these filters, it splits into two bands. The low pass filter extracts the coarse information of signal. The high pass filter gives the detailed information of the signal. After filtering the signal’s sampling frequency is too high, so half the frequencies are discarded by the down sampling operation.

![Original image](image1.png) ![First level decomposition](image2.png) ![Second level decomposition](image3.png)

(a) Original image First level decomposition using high pass and low pass filter bank.  
(b) Second level decomposition

A two dimensional transform is accomplished by performing two separate one dimensional transforms. First the image is filtered along the row and decimated by two and then followed by filtering the sub image along the column and decimated by two. This operation divides the image into four bands namely, LL, LH, HL and HH respectively as shown in Fig 4.58(a). Further decomposition can be achieved by acting upon the LL sub band successively and resultant image is split into multiple bands as shown in Fig 4.58 (b)

### 4.8.3 Proposed Algorithm:

1) The Discrete Wavelet Transform (DWT) is applied on the database till the third level of decomposition.
2) For each level of decomposition the LL coefficients of previous stage is used as input
3) Store all three level’s decomposition coefficients separately.
4) Read the test image
5) Decompose it till third level
6) On each decomposition level compare the test image’s approximation coefficients with the stored coefficients of database using the Euclidean distance as a similarity measure.
7) Calculate the accuracy at each level.
8) Test the robustness of algorithm for different percentage of occlusion and various types noise introduced on the test image

4.8.4 Result Analysis and Discussion

Here DWT on three different decomposition level is considered. The approximate coefficients of training set on each level of decomposition are stored separately and the test image is compared on each decomposition level for recognition. Following graph (Fig 4.59) shows the accuracy for three decomposition levels.
Fig 4.59 Graph of percentage accuracy in both the databases with respect to 1st, 2nd and 3rd decomposition level

**Observation:** As the decomposition levels of DWT in image increases the efficiency of face recognition reduces due to availability of lesser coefficients per decomposition level.

Following graphs (Fig 4.60-Fig 4.62) shows the performance of DWT under occlusion conditions applied on each decomposition level for both databases.
Fig 4.60 Graph of percentage accuracy in both the databases with respect to 1\textsuperscript{st} level decomposition in case of occlusion introduced as %length x %width of image for both databases.

Fig 4.61 Graph of percentage accuracy in both the databases with respect to 2\textsuperscript{nd} level decomposition in case of occlusion introduced as %length x %width of image for both databases.
Fig 4.62 Graph of percentage accuracy in both the databases with respect to 3rd level decomposition in case of occluded images.

**Observation:** As the decomposition level of DWT is increased the accuracy drops down from near 90% to 70% in both database local and ORL.

The robustness of algorithm is checked for the different percentage of occlusion and various types of noise like Speckle Noise, Gaussian noise, Salt and Pepper noise introduced on the test image.

Following graphs shows the results for different types of noise applied on test images for Local and ORL database.
Speckle Noise:

Fig 4.63 Performance of DWT in presence of a Speckle Noise for local database.

Fig 4.64 Performance of DWT in presence of a Speckle Noise for ORL database.

Observation: It has been observe clearly from graphs(Fig 4.63-Fig 4.64) that after first level decomposition the algorithm withstands the Speckle Noise addition up to 50%. Second level decomposition
withstands the noise up to 30% and third decomposition level cannot withstand the noise at all for both databases. ORL database shows better performance under noise addition conditions compared to local database.

**Gaussian Noise:**

![Performance of DWT in presence of a Gaussian Noise for local database](image1)

![Performance of DWT in presence of a Gaussian Noise for ORL database.](image2)

Fig 4.65 Performance of DWT in presence of a Gaussian Noise for local database

Fig 4.66 Performance of DWT in presence of a Gaussian Noise for ORL database.
**Observation:** The graphs (Fig 4.65-Fig 4.66) give an idea about the variation of a Gaussian noise and its effect on accuracy for both databases. The local database can tolerate noise addition upto 10% for the local database after first level decomposition. But second level decomposition cannot withstand noise addition after 5% and third level decomposition is not at all robust against the noise addition.

**Salt and Pepper Noise:**

Fig 4.67 Performance of DWT in presence of a Salt and Pepper noise for local database.
Fig 4.68 Performance of DWT in presence of a Salt and Pepper noise for ORL database.

**Observation:** The graphs (Fig 4.67-Fig 4.68) shows that as Salt and pepper noise in image increases the accuracy reduces. The first level decomposition is giving good results for noise addition upto 40% whereas second and third level decomposition fails after 20% of noise addition.

Table 4.7 is an overall performance table for the DWT. The data given for accuracy of algorithm is for 90% image energy. Under occlusion and noise addition conditions the cutoff point is 50% accuracy below which algorithm is not reliable
Table 4.7 Overall Performance of DWT

<table>
<thead>
<tr>
<th>Decomposition level</th>
<th>Database</th>
<th>W1</th>
<th>W2</th>
<th>W3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local</td>
<td>92</td>
<td>88</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>ORL</td>
<td>93</td>
<td>90</td>
<td>65</td>
</tr>
<tr>
<td>Accuracy (90%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Local</td>
<td>92</td>
<td>88</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>ORL</td>
<td>93</td>
<td>90</td>
<td>65</td>
</tr>
<tr>
<td>%Occlusion withstand</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Local</td>
<td>92</td>
<td>88</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>ORL</td>
<td>93</td>
<td>90</td>
<td>65</td>
</tr>
<tr>
<td>Speckle Noise</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Local</td>
<td>92</td>
<td>88</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>ORL</td>
<td>93</td>
<td>90</td>
<td>65</td>
</tr>
<tr>
<td>Gaussian</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Local</td>
<td>92</td>
<td>88</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>ORL</td>
<td>93</td>
<td>90</td>
<td>65</td>
</tr>
<tr>
<td>Salt &amp; Pepper</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Local</td>
<td>92</td>
<td>88</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>ORL</td>
<td>93</td>
<td>90</td>
<td>65</td>
</tr>
<tr>
<td>FAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Local</td>
<td>92</td>
<td>88</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>ORL</td>
<td>93</td>
<td>90</td>
<td>65</td>
</tr>
<tr>
<td>FRR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Local</td>
<td>92</td>
<td>88</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>ORL</td>
<td>93</td>
<td>90</td>
<td>65</td>
</tr>
</tbody>
</table>

The Table 4.7 shows overall performance of Discrete Wavelet Transform method. The image of size 64x64 is decomposed into three decomposition levels 32x32, 16X16 and 8x8 respectively. As decomposition level increases number of approximate and detailed coefficients reduces which intern reduces the face recognition efficiency. FAR is 50% for all decomposition level which is remarkably high. FRR is zero in all decomposition level.

4.9 Summary

In this chapter the face recognition based on various sinusoidal and nonsinusoidal transforms are studied for standard as well as locally created unconstrained database. The concept of Row and Column Feature Vector (RV/CV) is also introduced which in general has given better performance compared to full image transform.

Among all the transforms when applied on full image the newly proposed Kekre’s transform gives better face recognition accuracy. Among the RV and CV methods WHT (RV) gives better performance among all.
Wavelet Transform after first level of decomposition gives good accuracy but it’s FAR is very high and as decomposition level increase the accuracy reduces.

In all the algorithms as energy increases the accuracy also increases and as occlusion or noise addition increases accuracy reduces.