**Introduction**

Today, almost every area of technical endeavor is impacted in some way by digital image processing. Image processing is used in medical field to analyze X-ray and MRI, in astronomy it is used to analyze images of space object. Image processing in microwave band have improved satellite imaging. Image processing is also used in the field of weather forecasting, gene study, biometrics, etc. In literature, kinship has been Kinship is verified using descriptor like DAISY (Digital Accessible Information System) and SIFT (Scale Invariant Feature Transform).

This report is based on the task of localizing features in human face images that leads to an attempt of kinship verification. We look into this challenge by applying image processing, geometric and bio informatics techniques. Human faces have some features that explicitly or implicitly indicate the family linkage. We approach this challenge by identifying the facial features/traits from images, enhancing their quality of representation and compare features for kinship verification activities.

**Chapter 1**

* 1. **Problem Statement**

Localization of face features from face images of children and their biological parents leading to verification of their kinship.

**1.2 Problem Definition**

Kinship verification is defined as the verification of the existence of genetic relationship (children – biological parent relation) between two persons by analyzing their face images and localizing the face features using digital image processing techniques [4].

**1.3 Objective**

Given a human face image of standard dimension our objective is to localize the face features using image processing and geometric techniques. A scheme is proposed for comparing the geometric representation of face features of two such images as an attempt to determine the existence of kinship between the corresponding persons.

**1.4 Tools and Platform**

Softwares used for the project are

1. Dev-C++ (C language coding)
2. Irfanview (image viewer)

Project work has been developed in

1. Windows 10 platform
2. i5 processor
3. 4 GB RAM

**1.5 Brief Discussion on Problem**

**1.5.1 Motivation**

Motivation for doing this project involves its use in real world for

A) Finding missing children.

B) Helping in adoption of children.

C) Stopping trafficking/smuggling of children.

**1.5.2 Literature Survey**

In stochastic combination method, a descriptor named DAISY (Digital Accessible Information System) is used for extraction of familial traits. The DAISY description is computed in a neighborhood around each center pixel with spatial Gaussian kernels arranged similar to the shape of a daisy. The traits are used for calculating kinship. Features which suggest kinship and features which contradict it are kept in 2 different classes. Bayes’ theorem is applied if probability of kinship is above a certain percentage then kinship exist else it does not exist [4].

In spring like connection model, a descriptor named SIFT (Scale Invariant Feature Transform) is used for extraction of familial traits. SIFT extract 22 features from each of the images and combine them into a feature vector. A spring like Connection Model Is formed by those features where each feature is presented and are kept in constant distance. Similarity in the model of 2 images suggests kinship [2].

The original LBP operator extract features of image by LBP codes which are formed by comparing neighbor pixels with central pixel (every pixel of image are used starting from the top left corner) in terms of intensity values. LBP mask is used on LBP code to get every output pixels which forms output image consist of features we want for kinship verification. If two different person’s features of output images match then kinship is verified [3]. Extended LBP uses circular mask (P, R) denotes a neighborhood of P sampling points on a circle of radius of R [3]. Improved LBP (ILBP) compares all the pixels (including the central pixel) with the mean intensity of all the pixels in the neighborhood [3]. Elongated LBP (ELBP) operator consists of several LBP codes at multiple layers, which encode the GD (gray-value differences) between the central pixel and its neighboring pixels. The shape of the neighborhood in ELBP is an ellipse where A and B denote the long axis and short axis, respectively, and m is the number of neighboring pixels [3]. Multi-block LBP (MB-LBP) compares average intensities of neighboring sub-regions instead of comparing neighboring pixels [3]. Virtual LBP (VLBP) combines motion and appearance information, and can thus be used to analyze image sequences or videos. It makes use of dynamic texture analysis of 2-D time series [3].Other methods based on LBP may be seen in [5, 7].

A new chain code for shapes composed of regular cells is deﬁned. This boundary chain code is based on the numbers of cell vertices which are in touch with the bounding contour of the shape. This boundary chain code is termed vertex chain code (VCC) [8].

Some important characteristics of the VCC are: (1) The VCC is invariant under translation and rotation, and optionally may be invariant under starting point and mirroring transformation. (2) Using the VCC it is possible to represent shapes composed of triangular, rectangular, and hexagonal cells. (3) The chain elements represent real values not symbols such as other chain codes, are part of the shape, indicate the number of cell vertices of the contour nodes, maybe operated for extracting interesting shape properties. (4) Using the VCC it is possible to obtain relations between the bounding contour and interior of the shape [8].

Needleman Wunsch algorithm is a method for finding similarities in the amino acid sequences of two proteins has been developed. From these findings it is possible to determine whether significant homology exists between the proteins [9].

* + 1. **Brief Summary**

The project is completed in five steps. First step is identification of face features where the length and width of the portions of the image containing eyes, nose and mouth are approximately measured and are extracted. In the second step LBP is used with proper threshold so that the object can be detected. Then in third step, the object is preprocessed to form a polygon from disjoint object parts. The fourth step deals with geometric representation of the object by extraction of its chain code. In the last step comparison of the geometric structures representing face features are done by comparing their chain code using Needleman-Wunsch algorithm.

**1.5.4 Assumptions**

Some assumptions are taken under consideration which includes

A) All images should be of equal height and width.

B) All images should be taken in standard lighting condition (brightness) and with standard equipment.

C) All images should be taken by keeping the model persons in same standard posture.

D) All images should be taken avoiding any sudden flash of light in image or blurriness.

E) For every image ratio of background data (ear, eyebrows, backdrops, Neck, Jaw, Forehead, Head & its hair) & facial/familial feature based data should be same.

F) Contrast of both images (in case of parent-child pairs) should be equal or nearly equal.

**Chapter 2**

**Concepts and Problem analysis**

Some preliminary concepts related to the proposed method are explained below along with a few experimental results.

**2.1 Local Binary Pattern (LBP)**

LBP was first introduced by Timo Ojala et al. in 1996.

The LBP mainly acts as unified approach to the existing statistical and structural models of texture

analysis. LBP stands for Local Binary Pattern. The LBP operator was originally designed for texture

description. The operator assigns a label to every pixel of an image by thresholding the 3\*3- neighbourhood of each pixel with the center pixel value and considering the result as a binary

number (Fig. 1) [3]. If neighboring pixel value <= (center pixel value + threshold value), then 1 is

assigned in LBP code, otherwise 0 is assigned. Threshold values differ from one image to another

image.

An image of m\*n size is padded with zero at its surrounding to turn it into an image of (m+2) \* (n+2)

size. Now starting from top left corner of the image we are running the LBP operator throughout the

image. By this an output image is created including the desired features. Based on the similarity on these features kinship measurement can be done. Weight matrix (3\*3) used on LBP code is shown in Fig. 2 where the central pixel’s intensity value is calculated as

(1\*1)+(1\*2)+(0\*4)+(1\*8)+(0\*16)+(0\*32)+(1\*64)+(1\*128)=203

203 will now replace 120 in given sample matrix in Fig. 1.

Start Here

|  |  |  |
| --- | --- | --- |
| 150 | 200 | 4 |
| 128 | 120 | 185 |
| 164 | 32 | 16 |

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| 150 | 200 | 4 |
| 128 | 203 | 185 |
| 164 | 32 | 16 |

End Here

Fig. 1: Producing LBP code with zero threshold value.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 4 |
| 128 | 0 | 8 |
| 64 | 32 | 16 |

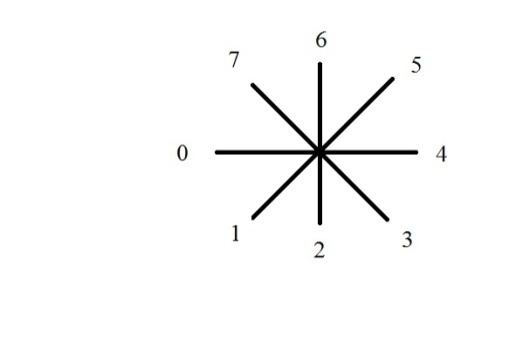
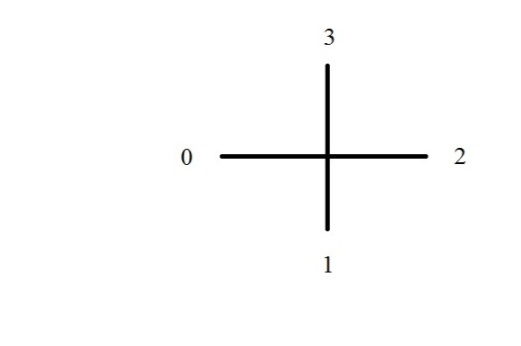
Fig. 2: Weight matrix.

**2.2 Chain Code**

The boundaries of any discrete shape composed of regular cells can be represented by chain codes. In general, a coding scheme for extracting chain code must satisfy three objectives: (1) it must faithfully preserve the information of interest; (2) it must permit compact storage and be convenient for display; and (3) it must facilitate any required processing. The three objectives are somewhat in conﬂict with each other, and any code necessarily involves a compromise among them.

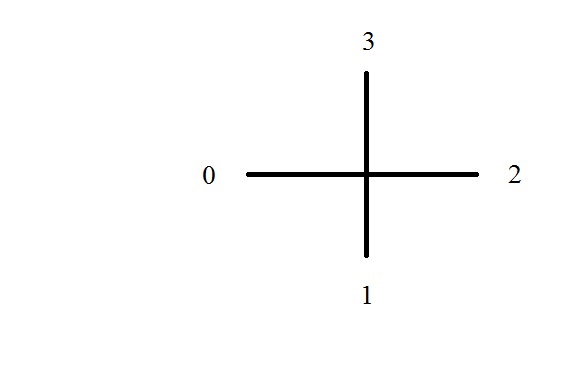
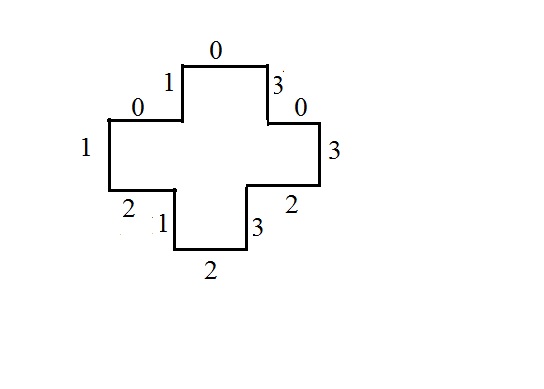
Chain code is a sequence of integers which represents the shape of the polygon. While tracing the boundary of a polygon, the direction of motion at every pixel is captured in the chain code representation of the polygon. Chain codes are extracted in two ways using 4-neighbourhood and using 8-neighbourhood shown in Fig. 3.

**Polygon:** A polygon is a closed 2-dimensional geometric object consisting of points (vertices) and line segments (edges) with no consecutive points collinear and line segments joining consecutive pairs of points. Examples of polygons are shown in Fig. 4(b) and 5(b).



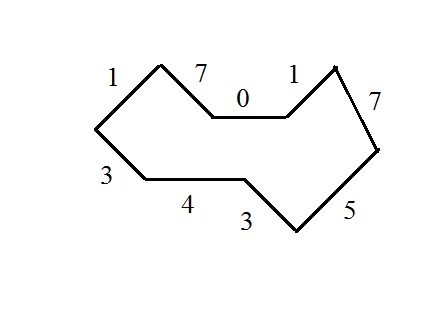
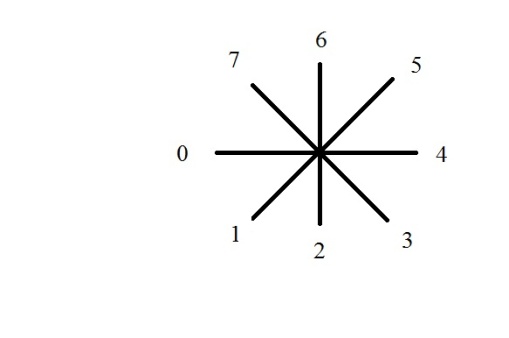
a) 4 neighbourhood directions b) 8 neighbourhood directions

Fig. 3: Chain code directions are presented.

a) 4 neighbourhood directions b) Structure of a polygon

Fig. 4: Application of 4 neighbourhood chain code. The 4-neighborhood chain code is 1 2 1 2 3 2 3 0 3 0 1 0.

a) 8 neighbourhood directions b) Structure of a polygon

Fig. 5: Application of 8 neighbourhood chain code. The 8-neighborhood chain code is 3 4 3 5 7 1 0 7 1.

**2.3 Needleman Wunsch Algorithm**

A global alignment algorithm based on dynamic programming which is used in bioinformatics to compare two different protein sequences. Score is calculated based on pre-defined match score, mismatch score and gap penalty and based on the scorethe extent of similarity between two sequences can be determined. Consider two alignments as Sequence 1: C G C A T and Sequence 2: C G A T. Length of sequence 1 is 5 and Length of sequence 2 is 4. Consider match score = 5, mismatch score = -2 and gap penalty = -5. First a scoring matrix of size ((length of sequence 1) + 1) \* ((length of sequence 2) + 1)) is formed.

First element of matrix cell is assigned with 0. First row and first columns’ values are assigned with (previous left or upper cell’s value + gap penalty). Initialization of scoring matrix is shown in Fig. 6.

For other cells, assigned value will be = maximum {(Previous diagonal cell’s value + match/mismatch score), (Previous left cell’s value + gap penalty), (Previous upper cell’s value + gap penalty)}.It is described in Fig. 7.Completion of scoring matrix is shown in Fig. 8.

Seq 1 : C G C A T

Seq 2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | -5 | -10 | -15 | -20 | -25 |
| -5 |  |  |  |  |  |
| -10 |  |  |  |  |  |
| -15 |  |  |  |  |  |
| -20 |  |  |  |  |  |

C

G

A

T

Fig. 6: Scoring in scoring matrix is initiated.

Seq 1 : C

|  |  |
| --- | --- |
| 0 + 5 = 5 | -5 + (-5) = -10 |
| -5 + (-5) = -10 | Maximum(5,-10,-10)  = 5 |

Seq 2:

C

Fig. 7: Scoring strategy for a cell is shown.

Seq 1 : C G C A T

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | -5 | -10 | -15 | -20 | -25 |
| -5 | 5 | 0 | -5 | -10 | -15 |
| -10 | 0 | 10 | 5 | 0 | -5 |
| -15 | -5 | 5 | 8 | 10 | 5 |
| -20 | -10 | 0 | 3 | 6 | 15 |

Seq 2:

C

G

A

T

Fig. 8: Scoring matrix completion is shown.

Seq 1 : C G C A T

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | -5 | -10 | -15 | -20 | -25 |
| -5 | 5 | 0 | -5 | -10 | -15 |
| -10 | 0 | 10 | 5 | 0 | -5 |
| -15 | -5 | 5 | 8 | 10 | 5 |
| -20 | -10 | 0 | 3 | 6 | 15 |

Seq 2:

C

G

A

T

Fig. 9: Trace back on scoring matrix.

In second step, trace back is done where the method of assigning value in the cells in north-west corner to south-east corner is shown. Trace back is presented in Fig. 9.

In third step, alignment is done based on trace back. If a horizontal flow of trace back occurs then gap in sequence 2 will appear. If a vertical flow of trace back occurs then gap in sequence 1 will appear. In given example alignment will be

Sequence 1: C G C A T

Sequence 2: C G \_ A T

Here 4 matches, 0 mismatch and 1 gap has occurred. Therefore, the score will be

[((Number of matches) \* (match score)) + ((Number of mismatches) \* (mismatch score)) + ((Number of gaps) \* (gap penalty))] = (4 \* 5) + (0 \* -2) + (1 \* -5) =15.

**Chapter 3**

**Design and Methodology**

The proposed method may be divided into five steps as explained below.

**3.1 Identification of face features**

The length and width of the portions of the image containing eyes, nose and mouth are approximately measured. Measurements of different facial portions are done by measuring the total number of rows and columns. On that basis portion of image containing eyes, nose, and mouth are extracted. On given Fig. 10 and 11 sizes are indicated like (row size \* column size). Figure 10 and 11 shows application of first step on a parent child image pair.

D:\Last Year Project\Presentation\F3\F3Mouth.jpgD:\Last Year Project\Presentation\F3\F3Nose.jpgD:\Last Year Project\Presentation\F3\F3Eye.jpg

b) Eyes (22 \* 64)

C) Nose (17 \* 64)

a) Parent’s image (64 \* 64)

d) Mouth (15 \* 64)

Fig. 10: Extraction of eyes (22 \* 64), nose (17 \* 64) and mouth (15 \* 64) from parent’s image.

D:\Last Year Project\Presentation\S3\S3Eye.jpgD:\Last Year Project\Presentation\S3\S3Nose.jpgD:\Last Year Project\Presentation\S3\S3Mouth.jpg

b) Eyes (20 \* 64)

C) Nose (16 \* 64)

d) Mouth (14 \* 64)

a) Child’s image (64 \* 64)

Fig. 11: Extraction of eyes (20 \* 64), nose (16 \* 64) and mouth (14 \* 64) from child’s image

**3.2 Extraction of objects**

After taking out the portions we are performing Local Binary Pattern operation. Using LBP the objects of eyes, nose and mouth can be detected with respect to the background. Pixel intensity values of the object are higher than that of the background. Intensity values nearer to each other are considered as the portion of the object. By setting up different threshold values, 8 bit binary value LBP code is found. In uniform LBP pattern 8 bit binary number does not have interleaved 1’s and 0’s like 11001111 in other case interleaved 1s’ and 0s’ exist like 11010110 which is called non-uniform LBP pattern. We will take the uniform and non-uniform patterns of the LBP code into consideration to explain the utility of our modified LBP. LBP codes are two type uniform and non uniform. If the detected LBP pattern is uniform it means central pixel is part of object we need to extract. Hence those pixels are given intensity value of 255 or converted to white pixels. Other pixels which fail to represent uniform patterns are assigned with intensity value of 0 or converted to black pixels. In Fig. 12, use of modified LBP is shown on a sample 3 \* 3 matrix. First the 8 bit binary LBP code is generated and if the generated value is uniform then central pixel’s intensity value is assigned with value 255. Fig.13 and 14 shows application of second step on the parent child images pair.

|  |  |  |
| --- | --- | --- |
| 120 | 120 | 130 |
| 140 | 236 | 245 |
| 150 | 250 | 240 |

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 |  | 1 |
| 0 | 1 | 1 |

8 bit binary LBP code

00011100

Pattern is uniform.

|  |  |  |
| --- | --- | --- |
| 120 | 120 | 130 |
| 140 | 255 | 245 |
| 150 | 250 | 240 |

Fig. 12: Concept of modified LBP is shown.

D:\Last Year Project\Presentation\F3\F3ED(Eye)+50.jpgD:\Last Year Project\Presentation\F3\F3Eye.jpg

D:\Last Year Project\Presentation\F3\F3ED(Nose)+30.jpgD:\Last Year Project\Presentation\F3\F3Nose.jpg a) Eyes (22 \* 64) d) Objects of eyes (22 \* 64)

D:\Last Year Project\Presentation\F3\F3ED(Mouth)+30.jpgD:\Last Year Project\Presentation\F3\F3Mouth.jpg b) Nose (17 \* 64) e) Object of nose (17 \* 64**)**

c**)** Mouth (15 \* 64) f) Objects of mouth (15 \* 64)

Fig. 13: Extraction of objects of eyes (22 \* 64), nose (17 \* 64) and mouth (15 \* 64) of parent’s image.

D:\Last Year Project\Presentation\S3\S3Eye.jpgD:\Last Year Project\Presentation\S3\S3ED(Eye)+50.jpg

D:\Last Year Project\Presentation\S3\S3ED(Nose)+30.jpgD:\Last Year Project\Presentation\S3\S3Nose.jpga) Eyes (20 \* 64) d) Objects of eyes (20 \* 64)

D:\Last Year Project\Presentation\S3\S3Mouth.jpgD:\Last Year Project\Presentation\S3\S3ED(Mouth)+50.jpg b) Nose (16 \* 64) e) Object of nose (16 \* 64)

c) Mouth (14 \* 64) f) Objects of mouth (14 \* 64)

Fig. 14: Extraction of objects of eyes (20 \* 64), nose (16 \* 64) and mouth (14 \* 64) of child’s image.

**3.3 Preprocessing on objects**

Preprocessing is required if extracted objects are disjoint. For proper representation of an object as a polygon it is necessary for the parts of the object to be connected properly. A count matrix is a matrix representing the number of white or object pixel that a black or background pixel has in its 8-neighbourhood. In Fig. 17, if a black cell is surrounded by n white pixels then its count in count matrix will be n. X refers to the object pixels shown in count matrix in Fig. 17(b). Now set threshold as K (K must be value between 0 and 8). K is 4 in given example of Fig. 17(c). If K > = n then that particular background pixel is included in the object or converted to white pixel. In Fig. 17(d), black pixel is turned into white pixel. Figure 15 and 16 shows application of second step on the parent child image pair.

D:\Last Year Project\Presentation\F3\F3Eop50(4).jpgD:\Last Year Project\Presentation\F3\F3ED(Mouth)+30.jpgD:\Last Year Project\Presentation\F3\F3ED(Eye)+50.jpgD:\Last Year Project\Presentation\F3\F3ED(Nose)+30.jpg

**D:\Last Year Project\Presentation\F3\F3Nop30(3).jpgD:\Last Year Project\Presentation\F3\F3Nop30(3).jpg** a) Objects of eyes (22 \* 64) d) Preprocessed objects of eyes (22 \* 64)

D:\Last Year Project\Presentation\F3\F3Mop30(3).jpg b) Object of nose (17 \* 64) e) Preprocessed object of nose (17 \* 64)

c) Objects of mouth (15 \* 64) f) Preprocessed object of mouth (15 \* 64)

Fig. 15: Preprocessing of objects of eyes (22 \* 64), nose (17 \* 64) and mouth (15 \* 64) of parent’s images.

D:\Last Year Project\Presentation\S3\S3Nop30(1).jpgD:\Last Year Project\Presentation\S3\S3ED(Mouth)+50.jpgD:\Last Year Project\Presentation\S3\S3ED(Nose)+30.jpgD:\Last Year Project\Presentation\S3\S3Mop50(3).jpgD:\Last Year Project\Presentation\S3\S3Eop50(3).jpgD:\Last Year Project\Presentation\S3\S3ED(Eye)+50.jpg

a) Objects of eyes (22 \* 64) d) Preprocessed objects of eyes (20 \* 64)

b) Object of nose (17 \* 64) e) Preprocessed object of nose (16 \* 64)

c) Objects of mouth (15 \* 64) f) Preprocessed object of mouth (14 \* 64)

Fig.16: Preprocessing of objects of eyes (20 \* 64), nose (16 \* 64) and mouth (14 \* 64) of child’s images.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **1** | **1** | **0** |
| **1** | **X** | **2** | **1** |
| **2** | **3** | **X** | **1** |
| **2** | **X** | **4** | **2** |
| **3** | **X** | **X** | **1** |
| **2** | **X** | **3** | **1** |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

a) Sample image b) Count matrix

c) Threshold 4 d) Preprocessed image

Fig. 17: Concept of preprocessing is shown.

**3.4 Geometric representation of the objects**

Geometric representation of an object is done by extracting its chain code. We are using 8-neighbourhood chain code algorithm where lower number indicates higher priority in chain code. First far north-west pixel is selected. 8 neighbourhood chain code algorithm is used on it until chain code of that object is extracted. In Fig. 18, a polygon is shown that represents white pixel surrounded by black pixels. At first we observe the 4-neighboring pixels (north, south, east and west) of a white pixel. If all the four pixels are white, then the pixel in question is not a boundary pixel. All other pixels are selected as boundary pixels of the object. Now polygon’s far north-west boundary pixel is selected in Fig. 18(a). It is surrounded by two boundary pixels (north-east and south pixels). According to the Fig. 3(b), south pixel will have more priority than north-east pixel as lower number indicates higher priority in chain code. This way we will move on to the south pixel shown in Fig. 18(b). The pixel which is chosen first is marked so that same pixel will not be used again to represent chain code in near future. This process is repeated until we will have no more boundary pixel to visit. In this way chain code of that polygon will be extracted. Figure 18(d) represent completion of chain code extraction.

.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

a) First pixel selected from the shape #1 b) Transition to the next border pixel

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

d) In this way border pixels are visited c) Chain code extraction complete

Fig.18: Geometric representation (chain code extraction) of the shape #1. Chain code of shape #1 is 2 3 3 4 4 6 6 6 7 0 0 1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

a) Shape #2 b) Chain code extraction complete

Fig. 19: Geometric representation (chain code extraction) of the shape #2. Chain code of shape #2 is 2 2 2 2 4 4 4 4 6 6 6 6 0 0 0 0.

**3.5 Comparison of the geometric structures representing face features**

The geometric structures representing face features are compared by Needleman Wunsch Algorithm. Length of chain code of shape #1 in Fig. 18 is 12 and length of chain code of shape #2 in Fig. 19 is 16. Scoring matrix of size (13 \* 17) is formed. Consider match score = 5, mismatch score = -2 and gap penalty = -5.

Step 1:

First element of matrix cell is assigned with 0. First row and first column’s values are assigned with (previous left or upper cell’s value + gap penalty). For Other cells, assigned value will be equal to maximum {(Previous diagonal cell’s value + match/mismatch score), (Previous left cell’s value + gap penalty), (Previous upper cell’s value + gap penalty)}.

Step 2:

Trace back is done where the method of assigning values from the north-west corner to south-east corner is shown in Fig. 20.

In following Fig. 20, step 1 and 2 are shown. Coloured cell represents trace back.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SEQ 2 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 6 | 6 | 6 | 6 | 0 | 0 | 0 | 0 |
| SEQ 1 | 0 | -2 | -4 | -6 | -8 | -10 | -12 | -14 | -16 | -18 | -20 | -22 | -24 | -26 | -28 | -30 | -32 |
| 2 | -2 | 5 | 3 | 1 | -1 | -3 | -5 | -7 | -9 | -11 | -13 | -15 | -17 | -19 | -21 | -23 | -25 |
| 3 | -4 | 3 | 4 | 2 | 0 | -2 | -4 | -6 | -8 | -10 | -12 | -14 | -16 | -18 | -20 | -22 | -24 |
| 3 | -6 | 1 | 2 | 3 | 1 | -1 | -3 | -5 | -7 | -9 | -11 | -13 | -15 | -17 | -19 | -21 | -23 |
| 4 | -8 | -1 | 0 | 1 | 2 | 6 | 4 | 2 | 0 | -2 | -4 | -6 | -8 | -10 | -12 | -14 | -16 |
| 4 | -10 | -3 | -2 | -1 | 0 | 7 | 11 | 9 | 7 | 5 | 3 | 1 | -1 | -3 | -5 | -7 | -9 |
| 6 | -12 | -5 | -4 | -3 | -2 | 5 | 9 | 10 | 8 | 12 | 10 | 8 | 6 | 4 | 2 | 0 | -2 |
| 6 | -14 | -7 | -6 | -5 | -4 | 3 | 7 | 8 | 9 | 13 | 17 | 15 | 13 | 11 | 9 | 7 | 5 |
| 6 | -16 | -9 | -8 | -7 | -6 | 1 | 5 | 6 | 7 | 14 | 18 | 22 | 20 | 18 | 16 | 14 | 12 |
| 7 | -18 | -11 | -10 | -9 | -8 | -1 | 3 | 4 | 5 | 6 | 7 | 20 | 21 | 19 | 17 | 15 | 13 |
| 0 | -20 | -13 | -12 | -11 | -10 | -3 | 1 | 2 | 3 | 4 | 5 | 18 | 19 | 26 | 24 | 22 | 20 |
| 0 | -22 | -15 | -14 | -13 | -12 | -5 | -1 | 0 | 1 | 2 | 3 | 16 | 17 | 24 | 31 | 29 | 27 |
| 1 | -24 | -17 | -16 | -15 | -14 | -7 | -3 | -2 | -1 | 6 | 10 | 14 | 15 | 22 | 29 | 30 | 28 |

Fig 20: Scoring matrix completion and trace back is shown.

Step 3:

Alignment is done based on trace back. If a horizontal flow of trace back occurs then gap in sequence 2 will appear. If a vertical flow of trace back occurs then gap in sequence 1 will appear. In given example alignment will be

Shape #1: \_ \_ \_ 2 3 3 4 4 \_ 6 6 6 7 0 0 1

Shape #2: 2 2 2 2 4 4 4 4 6 6 6 6 0 0 0 0

Here 8 matches, 4 mismatches and 4 gaps exist. Therefore, the score will be

[((Number of matches) \* (match score)) + ((Number of mismatches) \* (mismatch score)) + ((Number of gaps) \* (gap penalty))] = (8 \* 5) + (4 \* -1) + (4 \* -2) =28.

**Chapter 4**

**Sample Codes**

The coding is divided into five modules. In first module face features like nose, eyes and mouth containing image portion from images are extracted. Objects are extracted from images containing eyes, nose and mouth in second module of coding. Objects are preprocessed for clarity of representation using third module of code. Preprocessed object is represented in geometric form. In fourth module 8-neighbourhood chain code of preprocessed object is extracted. Comparison of two chain code by implementing Needleman Wunsch algorithm is done in the final module. A portion of code from fourth module, i.e., chain code extraction is shown below.

#include<stdio.h>

#include<conio.h>

#define max(a,b) (a>b?a:b)

void cc();

void fw();

void n7();

. . .

void maxima();

void sc7();

. . .

int rs=18,cs=64;

int M[512][512],a[512][512],s[10],sx,sy,m,x,y,c=0;

FILE \*fp1,\*fp2,\*fp3;

int main()

{

int i,j,k;

fp1=fopen("op25(3).txt","r");

fp2=fopen("N1.txt","w");

fp3=fopen("NN1.txt","w");

for(i=0;i<rs;i++)

{

for(j=0;j<cs;j++)

{

M[i][j]=0;

}

}

for(i=0;i<rs;i++)

{

for(j=0;j<cs;j++)

{

fscanf(fp1,"%d",&k);

a[i][j]=k;

}

}

fw();

cc();

fprintf(fp3,"%d",c);

fclose(fp1);

fclose(fp2);

fclose(fp3);

return 0;

}

void fw()

{

int i,j,s=0;

for(j=0;j<cs;j++)

{

for(i=0;i<rs;i++)

{

if(a[i][j]==255)

s++;

if(s==1)

{

sx=i;

sy=j;

}

if(a[i-1][j]==255 && a[i][j-1]==255 && a[i][j+1]==255 && a[i+1][j]==255)

{

M[i][j]=1;

}

}

}

x=sx;

y=sy;

printf("(%d,%d)",x,y);

M[x][y]=1;

printf("\nSEQ: ");

}

void cc()

{

int i;

while((sx!=x || sy!=y)||(c==0))

{

n7();

**.**

**.**

**.**

maxima();

if(m==7)

n0c();

**.**

**.**

**.**

if(m==10)

{

printf(" 0 ");

fprintf(fp2," 0 ");

sx=x;

sy=y;

}

if(m== -1)

{

l7();

**.**

**.**

**.**

m=max(s[0],max(s[1],max(s[2],max(s[3],max(s[4],max(s[5],max(s[6],s[7])))))));

if(m==7)

{

a[x-1][y-1]=255;

M[x-1][y-1]=1;

printf(" 7 ");

fprintf(fp2," 7 ");

x--;

y--;

}

**.**

**.**

**.**

if(m== -1)

{

printf("object can't have chaincode");

fprintf(fp2,"object can't have chaincode");

sx=x;

sy=y;

}

}

c++;

}

}

void l7()

{

if(a[x-1][y-1]==0)

{

s[0]=7;

}

else

s[0]= -1;

}

**.**

**.**

**.**

void n7()

{

if(a[x-1][y-1]==255)

{

if(M[x-1][y-1]==0)

{

s[0]=0;

}

else if(M[x-1][y-1]==1)

{

if( sx==x && sy==y-1)

s[0]=10;

else if(sx==x+1 && sy==y-1 )

s[0]=20;

else if(sx==x-1 && sy==y-1)

s[0]=30;

else

s[0]= -1;

}

}

else

s[0]= -1;

}

void n6()

{

if(a[x-1][y]==255)

{

if(M[x-1][y]==0)

{

s[1]=1;

}

else if(M[x-1][y]==1)

{

if( sx==x && sy==y-1)

s[1]=10;

else if(sx==x+1 && sy==y-1 )

s[1]=20;

else if(sx==x-1 && sy==y-1)

s[1]=30;

else

s[1]= -1;

}

}

else

s[1]= -1;

}

void n5()

{

if(a[x-1][y+1]==255)

{

if(M[x-1][y+1]==0)

{

s[2]=2;

}

else if(M[x-1][y+1]==1)

{

if( sx==x && sy==y-1)

s[2]=10;

else if(sx==x+1 && sy==y-1 )

s[2]=20;

else if(sx==x-1 && sy==y-1)

s[2]=30;

else

s[2]= -1;

}

}

else

s[2]= -1;

}

void n4()

{

if(a[x][y+1]==255)

{

if(M[x][y+1]==0)

{

s[3]=3;

}

else if(M[x][y+1]==1)

{

if( sx==x && sy==y-1)

s[3]=10;

else if(sx==x+1 && sy==y-1 )

s[3]=20;

else if(sx==x-1 && sy==y-1)

s[3]=30;

else

s[3]= -1;

}

}

else

s[3]= -1;

}

void n3()

{

if(a[x+1][y+1]==255)

{

if(M[x+1][y+1]==0)

{

s[4]=4;

}

else if(M[x+1][y+1]==1)

{

if( sx==x && sy==y-1)

s[4]=10;

else if(sx==x+1 && sy==y-1 )

s[4]=20;

else if(sx==x-1 && sy==y-1)

s[4]=30;

else

s[4]= -1;

}

}

else

s[4]= -1;

}

void n2()

{

if(a[x+1][y]==255)

{

if(M[x+1][y]==0)

{

s[5]=5;

}

else if(M[x+1][y]==1)

{

if( sx==x && sy==y-1)

s[5]=10;

else if(sx==x+1 && sy==y-1 )

s[5]=20;

else if(sx==x-1 && sy==y-1)

s[5]=30;

else

s[5]= -1;

}

}

else

s[5]= -1;

}

void n1()

{

if(a[x+1][y-1]==255)

{

if(M[x+1][y-1]==0)

{

s[6]=6;

}

else if(M[x+1][y-1]==1)

{

if( sx==x && sy==y-1)

s[6]=10;

else if(sx==x+1 && sy==y-1 )

s[6]=20;

else if(sx==x-1 && sy==y-1)

s[6]=30;

else

s[6]= -1;

}

}

else

s[6]= -1;

}

void n0()

{

if(a[x][y-1]==255)

{

if(M[x][y-1]==0)

{

s[7]=7;

}

else if(M[x][y-1]==1)

{

if( sx==x && sy==y-1)

s[7]=10;

else if(sx==x+1 && sy==y-1 )

s[7]=20;

else if(sx==x-1 && sy==y-1)

s[7]=30;

else

s[7]= -1;

}

}

else

s[7]= -1;

}

void n7c()

{

printf(" 7 ");

fprintf(fp2," 7 ");

M[x-1][y-1]=1;

x--;

y--;

}

.

**.**

void maxima()

{

int i,j,z=0,k=0,max;

int sc[10];

for(i=0;i<8;i++)

{

if(s[i]==10 || s[i]==20 || s[i]==30 || s[i]== -1)

{

k++;

}

}

if(k==0)

{

max= -100;

for(i=0;i<8;i++)

{

if(max<s[i])

{

max=s[i];

}

}

m=max;

}

else if(k==8)

{

max= -100;

for(i=0;i<8;i++)

{

if(max<s[i])

{

max=s[i];

}

}

m=max;

}

else if(k>0 && k<8)

{

z=0;

for(i=0;i<8;i++)

{

if(s[i]!=10 && s[i]!=20 && s[i]!=30 && s[i]!= -1)

{

sc[z]=s[i];

z++;

}

}

max= -100;

for(i=0;i<z;i++)

{

if(max<sc[i])

{

max=sc[i];

}

}

m=max;

}

else

m=max(s[0],max(s[1],max(s[2],max(s[3],max(s[4],max(s[5],max(s[6],s[7])))))));

}

**Chapter 5**

**5.1 Testing**

The method of localization of shape features has been tested on various databases. In Fig. 21, application of LBP on Database 1 at the standard straight posture is found to extract objects from face images and preprocess them upto a high degree of accuracy. Also, the accuracy of the results for face images (Database 2) at different postures are appreciable (Fig. 21). In Fig. 22, the accuracy of the detected and preprocessed objects from face images of parent child pairs (Database 3) are also reasonable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Database 1 : Straight Faces** | | | | |
| D:\Last Year Project\Final sets\Straight\Pic\F1.jpg  Straight Face 1 | D:\Last Year Project\Final sets\Straight\Pic\M3.jpg  Straight Face 2 | D:\Last Year Project\Final sets\Straight\Pic\D3.jpg  Straight Face3 | D:\Last Year Project\Final sets\Straight\Pic\D1.jpg  Straight Face 4 | S1  Straight Face 5 |
| D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\F1\ED(Eye)+30.jpg  D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\F1\ED(Nose)+30.jpg  D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\F1\ED(Mouth)+20.jpg  Object Detection | D:\Last Year Project\Final sets\Straight\M3(C)\ED(Eye)+30.jpg  D:\Last Year Project\Final sets\Straight\M3(C)\ED(Nose)+25.jpg  D:\Last Year Project\Final sets\Straight\M3(C)\ED(Mouth)+25.jpg  Object Detection | D:\Last Year Project\Final sets\Straight\D3(C)\ED(Eye)+25.jpg  D:\Last Year Project\Final sets\Straight\D3(C)\ED(Nose)+25.jpg  D:\Last Year Project\Final sets\Straight\D3(C)\ED(Mouth)+25.jpg  Object Detection | D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\D1\ED(Eye)+30.jpg  D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\D1\ED(Nose)+20.jpg  D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\D1\ED(Mouth)+20.jpg  Object Detection | D:\Last Year Project\Final sets\Straight\S1(C)\ED(Eye)+35.jpg  D:\Last Year Project\Final sets\Straight\S1(C)\ED(Nose)+35.jpg  D:\Last Year Project\Final sets\Straight\S1(C)\ED(Mouth)+35.jpg  Object Detection |
| **D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\F1\op30E(3).jpg**  **D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\F1\op30N(1).jpg**  **D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\F1\op20M(1).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Straight\M3(C)\op30(1).jpg**  **D:\Last Year Project\Final sets\Straight\M3(C)\op25n(1).jpg**  **D:\Last Year Project\Final sets\Straight\M3(C)\op25(1).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Straight\D3(C)\op25(1).jpg**  **D:\Last Year Project\Final sets\Straight\D3(C)\op25n(1).jpg**  **D:\Last Year Project\Final sets\Straight\D3(C)\op25m(1).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\D1\op30E(1).jpg**  **D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\D1\op20N(3).jpg**  **D:\Last Year Project\Final sets\Straight\F1 Vs D1\PP\D1\op20M(3).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Straight\S1(C)\op35(3).jpg**  **D:\Last Year Project\Final sets\Straight\S1(C)\op35n(3).jpg**  **D:\Last Year Project\Final sets\Straight\S1(C)\op35m(3).jpg**  Preprocessed object |
| **Database 2: Postures** | | | | |
| **D:\Last Year Project\Final sets\posture\1.FPE\1.1.jpg**  Posture 1 | **D:\Last Year Project\Final sets\posture\1.FPE\2.2.jpg**    Posture 2 | **D:\Last Year Project\Final sets\posture\1.FPE\3.3.jpg**  Posture 3 | **D:\Last Year Project\Final sets\posture\1.FPE\4.4.jpg**  Posture 4 | **D:\Last Year Project\Final sets\posture\1.FPE\5.5.jpg**  Posture 5 |
| **D:\Last Year Project\Final sets\posture\1\ED(Eye)+35.jpg**  **D:\Last Year Project\Final sets\posture\1\ED(Nose)+20.jpg**  **D:\Last Year Project\Final sets\posture\1\ED(Mouth)+35.jpg**  Object Detection | **D:\Last Year Project\Final sets\posture\2\ED(Eye)+35.jpg**  **D:\Last Year Project\Final sets\posture\2\ED(Nose)+25.jpg**  **D:\Last Year Project\Final sets\posture\2\ED(Mouth)+35.jpg**  Object Detection | **D:\Last Year Project\Final sets\posture\3\ED(Eye)+25.jpg**  **D:\Last Year Project\Final sets\posture\3\ED(Nose)+25.jpg**  **D:\Last Year Project\Final sets\posture\3\ED(Mouth)+35.jpg**  Object Detection | **D:\Last Year Project\Final sets\posture\4\ED(Eye)+25.jpgD:\Last Year Project\Final sets\posture\4\ED(Nose)+25.jpg**  **D:\Last Year Project\Final sets\posture\4\ED(Mouth)+25.jpg**  Object Detection | **D:\Last Year Project\Final sets\posture\5\ED(Eye)+35.jpg**  **D:\Last Year Project\Final sets\posture\5\ED(Nose)+25.jpg**  **D:\Last Year Project\Final sets\posture\5\ED(Mouth)+35.jpg**  Object Detection |
| **D:\Last Year Project\Final sets\posture\1\op35(E)(3).jpg**  **D:\Last Year Project\Final sets\posture\1\op35(N)(3).jpg**  **D:\Last Year Project\Final sets\posture\1\op35(M2)(3).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\posture\2\op35(E)(2).jpg**  **D:\Last Year Project\Final sets\posture\2\op35(N)(2).jpg**  **D:\Last Year Project\Final sets\posture\2\op35(M)(3).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\posture\3\op35(E)(3).jpg**  **D:\Last Year Project\Final sets\posture\3\op35(N2)(3).jpg**  **D:\Last Year Project\Final sets\posture\3\op35(M)(1).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\posture\4\op25(E)(3).jpg**  **D:\Last Year Project\Final sets\posture\4\op25(N)(3).jpg**  **D:\Last Year Project\Final sets\posture\4\op25(M)(3).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\posture\5\op25(E)(3).jpg**  **D:\Last Year Project\Final sets\posture\5\op25(N2)(1).jpg**  **D:\Last Year Project\Final sets\posture\5\op25(M)(3).jpg**  Preprocessed object |

Fig. 21: Results for preprocessing on objects detected from face images at standard straight positions and at different postures.

Fig. 22: Results for preprocessing on objects detected from face images at parent and their respective child images.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Database 3: Parent child pairs** | | | | |
| D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\Original Picture\md_070_2.jpg  Child Face 1 | D:\Last Year Project\Final sets\Father son\S3.jpg  Child Face 2 (S3) | D:\Last Year Project\Final sets\Father son\S2.jpg  Child Face 3 (S2) | D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\Original Picture\md_088_2.jpg  Child Face 4 | D:\Last Year Project\Final sets\Father son\S6.jpg  Child Face 5 |
| D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D1(C)\ED(Eye)+30.jpg  D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D1(C)\ED(Nose)+25.jpg  D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D1(C)\ED(Mouth)+25.jpg  Object Detection | D:\Last Year Project\Final sets\Father son\S3(C)\ED(Eye)+50.jpg  D:\Last Year Project\Final sets\Father son\S3(C)\ED(Nose)+30.jpg  D:\Last Year Project\Final sets\Father son\S3(C)\ED(Mouth)+50.jpg  Object Detection | D:\Last Year Project\Final sets\Father son\S2(C)\ED(Eye)+40.jpgD:\Last Year Project\Final sets\Father son\S2(C)\op30(1).jpg  D:\Last Year Project\Final sets\Father son\S2(C)\ED(Nose)+35.jpg  Object Detection | D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D2(C)\ED(Eye)+25.jpg  D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D2(C)\ED(Nose)+10.jpg  D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D2(C)\ED(Mouth)+15.jpg  Object Detection | D:\Last Year Project\Final sets\Father son\S6(C)\ED(Eye)+30.jpg  D:\Last Year Project\Final sets\Father son\S6(C)\ED(Nose)+30.jpg  D:\Last Year Project\Final sets\Father son\S6(C)\ED(Mouth)+25.jpg  Object Detection |
| **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D1(C)\op30(3).jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D1(C)\op25n(3).jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D1(C)\op25(3).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Father son\S3(C)\op50(3).jpg**  **D:\Last Year Project\Final sets\Father son\S3(C)\op30(1).jpg**  **D:\Last Year Project\Final sets\Father son\S3(C)\op50(1).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Father son\S2(C)\op40(1).jpg**  **D:\Last Year Project\Final sets\Father son\S2(C)\op35(1).jpg**  **D:\Last Year Project\Final sets\Father son\S2(C)\op30m(1).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D2(C)\op25(3).jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D2(C)\op10(3).jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\D2(C)\op15(3).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Father son\S6(C)\op30(3).jpg**  **D:\Last Year Project\Final sets\Father son\S6(C)\op30n(3).jpg**  **D:\Last Year Project\Final sets\Father son\S6(C)\op25(3).jpg**  Preprocessed object |
| **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\Original Picture\md_070_1.jpg**  Parent Face 1 | **D:\Last Year Project\Final sets\Father son\F3.jpg**  Parent Face 1(F3) | **D:\Last Year Project\Final sets\Father son\F2.jpg**  Parent Face 1(F2) | **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\Original Picture\md_088_1.jpg**  Parent Face 1 | **D:\Last Year Project\Final sets\Father son\F6.jpg**  Parent Face 1 |
| **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M1(C)\ED(Eye)+30.jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M1(C)\ED(Nose)+25.jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M1(C)\ED(Mouth)+25.jpg**  Object Detection | **D:\Last Year Project\Final sets\Father son\F3(C)\ED(Mouth)+30.jpgD:\Last Year Project\Final sets\Father son\F3(C)\ED(Nose)+30.jpgD:\Last Year Project\Final sets\Father son\F3(C)\ED(Eye)+50.jpg**  Object Detection | **D:\Last Year Project\Final sets\Father son\F2(C)\ED(Eye)+25.jpg**  **D:\Last Year Project\Final sets\Father son\F2(C)\ED(Nose)+40.jpg**  **D:\Last Year Project\Final sets\Father son\F2(C)\ED(Mouth)+30.jpg**  Object Detection | **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M2(C)\ED(Eye)+30.jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M2(C)\ED(Nose)+12.jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M2(C)\ED(Mouth)+20.jpg**  Object Detection | **D:\Last Year Project\Final sets\Father son\F6(C)\ED(Eye)+30.jpg**  **D:\Last Year Project\Final sets\Father son\F6(C)\ED(Nose)+40.jpg**  **D:\Last Year Project\Final sets\Father son\F6(C)\ED(Mouth)+30.jpg**  Object Detection |
| **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M1(C)\op30(3).jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M1(C)\op25n(3).jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M1(C)\op25(3).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Father son\F3(C)\op50(4).jpg**  **D:\Last Year Project\Final sets\Father son\F3(C)\op30(1).jpg**  **D:\Last Year Project\Final sets\Father son\F3(C)\op30(3).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Father son\F2(C)\op25(3).jpg**  **D:\Last Year Project\Final sets\Father son\F2(C)\op40(1).jpg**  **D:\Last Year Project\Final sets\Father son\F2(C)\op30(1).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M2(C)\op30(3).jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M2(C)\op12(1).jpg**  **D:\Last Year Project\Final sets\Mother Daughter\Test 2(Mother-Daughter)\M2(C)\op20(1).jpg**  Preprocessed object | **D:\Last Year Project\Final sets\Father son\F6(C)\op30(5).jpg**  **D:\Last Year Project\Final sets\Father son\F6(C)\op40(3).jpg**  **D:\Last Year Project\Final sets\Father son\F6(C)\op30(3).jpg**  Preprocessed object |

**5.2 Result**

Geometric representation of the object is done by extracting chain code of that object. Figure 23 shows application of chain code on parent child pair like F3 and S3. Figure 24 shows application of chain code on parent child pair like F2 and S2.Figure 25 shows application of chain code on parent child pair like F3 and S2.

|  |  |  |  |
| --- | --- | --- | --- |
| Object | Image | Marked Image | Chain Code |
| Left Eye | Father | D:\Last Year Project\Presentation\F3\F3Eop50(4)mrle.jpg | 2 2 2 1 3 5 6 5 3 3 4 4 4 5 4 4 6 6 6 0 0 0 0 0 0 0 0 0 0 |
| Son | D:\Last Year Project\Presentation\S3\S3Eop50(3)mrle.jpg | 2 3 3 5 4 4 5 3 4 3 4 5 4 5 4 3 6 7 5 7 7 0 0 0 0 0 0 1 0 0 0 3 1 0 0 0 |
| Right Eye | Father | D:\Last Year Project\Presentation\F3\F3Eop50(4)mrre.jpg | 2 2 5 3 3 2 4 4 4 4 5 4 4 3 5 7 7 7 7 0 7 0 1 0 0 0 1 |
| Son | D:\Last Year Project\Presentation\S3\S3Eop50(3)mrre.jpg | 2 2 4 3 3 4 3 5 4 6 5 4 3 4 6 7 7 0 0 0 0 0 7 0 1 0 |
| Nose | Father | D:\Last Year Project\Presentation\F3\F3Nop30(3)mrn.jpg | 2 2 2 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 7 7 0 7 0 0 0 0 7 7 6 6 7 6 7 1 1 1 7 6 6 0 1 2 2 2 2 2 1 |
| Son | D:\Last Year Project\Presentation\S3\S3Nop30(1)mrn.jpg | 2 2 2 2 3 4 3 4 4 4 4 4 4 4 4 4 5 4 4 4 4 4 5 6 6 7 7 0 0 0 1 0 0 7 7 6 6 6 6 6 6 6 0 0 2 1 2 1 0 0 1 1 2 1 |
| Mouth | Father | D:\Last Year Project\Presentation\F3\F3Mop30(3)mrm.jpg | 2 2 3 4 3 3 3 4 3 4 4 3 4 4 4 4 4 4 4 5 4 5 5 6 7 5 4 4 4 4 4 4 4 4 3 5 5 7 6 0 7 0 7 0 0 1 0 0 0 7 6 0 0 1 1 0 0 7 0 7 0 0 1 0 0 0 1 1 0 0 1 |
| Son | D:\Last Year Project\Presentation\S3\S3Mop50(3)mrm.jpg | 2 3 4 3 3 3 4 3 4 3 4 3 4 4 4 4 5 4 4 5 4 6 7 0 0 0 0 0 0 7 5 4 4 4 4 4 4 4 4 4 4 4 5 4 4 3 5 4 6 6 0 7 0 0 1 1 7 0 0 0 1 0 7 0 7 0 0 0 7 0 0 1 0 0 0 0 0 1 1 0 |

Fig. 23: Chain code extraction on preprocessed images of F2 and S2.

|  |  |  |  |
| --- | --- | --- | --- |
| Object | Image | Marked Image | Chain Code |
| Left Eye | Father | D:\Last Year Project\Final sets\Father son\F2(C)\op25(3)mrre.jpg | 2 3 3 2 2 4 5 4 5 4 4 3 4 5 5 6 5 4 3 2 2 2 2 4 5 6 6 6 7 6 0 0 7 1 7 1 0 0 2 2 1 7 1 7 6 5 7 1 0 0 1 |
| Son | D:\Last Year Project\Final sets\Father son\S2(C)\op40(1)mrle.jpg | 2 2 4 3 2 3 2 3 4 3 4 4 5 4 3 4 4 4 4 5 6 6 5 0 5 3 4 6 6 6 0 0 2 7 0 0 0 7 7 0 0 7 0 0 0 0 0 2 1 0 0 |
| Right Eye | Father | D:\Last Year Project\Final sets\Father son\F2(C)\op25(3)mrle.jpg | 2 2 3 3 3 4 4 4 3 4 4 5 5 3 3 5 7 6 6 5 7 6 0 7 0 0 0 0 0 0 0 0 0 1 1 |
| Son | D:\Last Year Project\Final sets\Father son\S2(C)\op40(1)mrre.jpg | 2 2 2 3 3 3 3 2 4 4 4 4 4 4 5 3 4 4 6 5 4 4 4 5 5 4 6 6 6 6 6 0 0 7 0 0 0 2 1 0 7 6 0 0 0 0 0 0 2 1 0 1 7 0 0 |
| Nose | Father | D:\Last Year Project\Final sets\Father son\F2(C)\op40(1)mrm.jpg | 2 2 4 3 2 4 4 4 3 4 4 4 4 4 4 3 2 4 4 5 4 5 4 3 4 4 6 6 6 5 4 4 6 6 0 7 0 5 4 4 4 6 6 0 0 0 0 0 2 1 0 0 0 2 2 1 1 0 0 7 6 6 0 0 0 0 7 0 0 0 0 2 1 0 0 |
| Son | D:\Last Year Project\Final sets\Father son\S2(C)\op35(1)mrn.jpg | 2 2 3 3 4 4 3 2 4 4 4 4 4 4 4 4 6 5 4 4 3 4 4 4 6 6 7 5 4 6 6 7 6 0 0 1 0 0 2 1 0 1 2 2 1 7 6 6 6 0 0 1 0 7 6 5 5 6 6 6 0 0 0 1 2 2 2 2 1 0 0 |
| Mouth | Father | D:\Last Year Project\Final sets\Father son\F2(C)\op30(1)mrm.jpg | 2 1 3 4 4 3 1 2 2 2 3 3 4 3 4 3 5 3 4 4 4 5 5 4 4 5 4 5 4 6 7 7 6 7 7 0 1 3 3 1 0 1 0 0 0 0 0 0 0 0 0 7 5 4 4 4 4 4 4 4 5 7 0 7 0 0 0 0 0 0 0 0 7 7 |
| Son | D:\Last Year Project\Final sets\Father son\S2(C)\op30m(1)mrm.jpg | 2 2 3 3 2 2 3 3 4 3 4 4 4 4 3 4 4 4 4 4 5 4 4 5 5 4 5 6 6 5 4 4 4 5 6 6 7 7 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 7 0 0 0 1 0 0 0 0 0 |

Fig. 24: Chain code extraction on preprocessed images of F3 and S3.

|  |  |  |  |
| --- | --- | --- | --- |
| Object | Image | Marked Image | Chain Code |
| Left Eye | Father | D:\Last Year Project\Presentation\F3\F3Eop50(4)mrle.jpg | 2 2 2 1 3 5 6 5 3 3 4 4 4 5 4 4 6 6 6 0 0 0 0 0 0 0 0 0 0 |
| Son | D:\Last Year Project\Final sets\Father son\S2(C)\op40(1)mrle.jpg | 2 2 4 3 2 3 2 3 4 3 4 4 5 4 3 4 4 4 4 5 6 6 5 0 5 3 4 6 6 6 0 0 2 7 0 0 0 7 7 0 0 7 0 0 0 0 0 2 1 0 0 |
| Right Eye | Father | D:\Last Year Project\Presentation\F3\F3Eop50(4)mrre.jpg | 2 2 5 3 3 2 4 4 4 4 5 4 4 3 5 7 7 7 7 0 7 0 1 0 0 0 1 |
| Son | D:\Last Year Project\Final sets\Father son\S2(C)\op40(1)mrre.jpg | 2 2 2 3 3 3 3 2 4 4 4 4 4 4 5 3 4 4 6 5 4 4 4 5 5 4 6 6 6 6 6 0 0 7 0 0 0 2 1 0 7 6 0 0 0 0 0 0 2 1 0 1 7 0 0 |
| Nose | Father | D:\Last Year Project\Presentation\F3\F3Nop30(3)mrn.jpg | 2 2 2 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 7 7 0 7 0 0 0 0 7 7 6 6 7 6 7 1 1 1 7 6 6 0 1 2 2 2 2 2 1 |
| Son | D:\Last Year Project\Final sets\Father son\S2(C)\op35(1)mrn.jpg | 2 2 3 3 4 4 3 2 4 4 4 4 4 4 4 4 6 5 4 4 3 4 4 4 6 6 7 5 4 6 6 7 6 0 0 1 0 0 2 1 0 1 2 2 1 7 6 6 6 0 0 1 0 7 6 5 5 6 6 6 0 0 0 1 2 2 2 2 1 0 0 |
| Mouth | Father | D:\Last Year Project\Presentation\F3\F3Mop30(3)mrm.jpg | 2 2 3 4 3 3 3 4 3 4 4 3 4 4 4 4 4 4 4 5 4 5 5 6 7 5 4 4 4 4 4 4 4 4 3 5 5 7 6 0 7 0 7 0 0 1 0 0 0 7 6 0 0 1 1 0 0 7 0 7 0 0 1 0 0 0 1 1 0 0 1 |
| Son | D:\Last Year Project\Final sets\Father son\S2(C)\op30m(1)mrm.jpg | 2 2 3 3 2 2 3 3 4 3 4 4 4 4 3 4 4 4 4 4 5 4 4 5 5 4 5 6 6 5 4 4 4 5 6 6 7 7 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 7 0 0 0 1 0 0 0 0 0 |

Fig. 25: Chain code extraction on preprocessed images F3 and S2.

Needleman Wunsch algorithm is applied on parent child pairs for some respected match score, mismatch score and gap penalty. Total score is just an accumulation of Needleman-Wunsch score of left eye, right eye, nose and mouth.

Table 1 represent result acquired after applying Needleman Wunsch algorithm on chain code of F2 and S2‘s objects. Table 2 represent result acquired after applying Needleman Wunsch algorithm on chain code of F2 and S3‘s objects. Table 3 represent result acquired after applying Needleman Wunsch algorithm on chain code of F3 and S3‘s objects. Table 4 represent result acquired after Applying Needleman Wunsch algorithm on chain code of F3 and S2‘s objects.

Table 1: Result acquired after Needleman Wunsch algorithm on chain code of F2 and S2’s objects.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Match Score | Mismatch Score | Gap penalty | Score | | | | |
| Left eye | Right eye | Nose | Mouth | Total |
| 10 | -1 | -10 | 112 | 84 | 274 | 205 | 675 |
| 10 | -2 | -10 | 84 | 78 | 244 | 186 | 592 |
| 10 | 5 | -10 | 315 | 120 | 490 | 385 | 1310 |
| 10 | 7 | -10 | 393 | 132 | 562 | 447 | 1534 |
| 10 | 9 | -10 | 471 | 144 | 634 | 509 | 1758 |
| 10 | -5 | -10 | 15 | 60 | 190 | 150 | 415 |
| 5 | -1 | -5 | 42 | 39 | 122 | 93 | 296 |
| 5 | -2 | -5 | 19 | 33 | 104 | 81 | 237 |
| 5 | 0 | -5 | 70 | 45 | 155 | 115 | 385 |
| 1 | 0 | -1 | 14 | 9 | 31 | 23 | 77 |

Table 2: Result acquired after Needleman Wunsch algorithm on chain code of F2 and S3’s objects.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Match Score | Mismatch Score | Gap penalty | Score | | | | |
| Left eye | Right eye | Nose | Mouth | Total |
| 10 | -1 | -10 | 67 | 93 | 132 | 144 | 436 |
| 10 | -2 | -10 | 54 | 86 | 114 | 98 | 352 |
| 10 | 5 | -10 | 145 | 135 | 240 | 430 | 950 |
| 10 | 7 | -10 | 171 | 149 | 276 | 530 | 1126 |
| 10 | 9 | -10 | 197 | 163 | 312 | 630 | 1302 |
| 10 | -5 | -10 | 15 | 65 | 60 | 20 | 160 |
| 5 | -1 | -5 | 27 | 43 | 57 | 49 | 176 |
| 5 | -2 | -5 | 14 | 36 | 39 | 16 | 105 |
| 5 | 0 | -5 | 40 | 50 | 75 | 95 | 260 |
| 1 | 0 | -1 | 8 | 10 | 15 | 19 | 52 |

Table 3: Result acquired after Needleman Wunsch algorithm on chain code of F3 and S3’s objects.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Match Score | Mismatch Score | Gap penalty | Score | | | | |
| Left eye | Right eye | Nose | Mouth | Total |
| 10 | -1 | -10 | 94 | 110 | 293 | 142 | 639 |
| 10 | -2 | -10 | 88 | 100 | 276 | 104 | 568 |
| 10 | 5 | -10 | 160 | 185 | 395 | 475 | 1215 |
| 10 | 7 | -10 | 184 | 211 | 429 | 530 | 1354 |
| 10 | 9 | -10 | 208 | 237 | 463 | 590 | 1498 |
| 10 | -5 | -10 | 70 | 70 | 225 | -40 | 325 |
| 5 | -1 | -5 | 44 | 50 | 138 | 52 | 284 |
| 5 | -2 | -5 | 38 | 40 | 121 | -23 | 176 |
| 5 | 0 | -5 | 50 | 60 | 155 | 180 | 445 |
| 1 | 0 | -1 | 10 | 12 | 31 | 36 | 89 |

Table 4: Result acquired after Needleman Wunsch algorithm on chain code of F3 and S2’s objects.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Match Score | Mismatch Score | Gap penalty | Score | | | | |
| Left eye | Right eye | Nose | Mouth | Total |
| 10 | -1 | -10 | 15 | -76 | 140 | 353 | 432 |
| 10 | -2 | -10 | 10 | -82 | 130 | 336 | 394 |
| 10 | 5 | -10 | 45 | -40 | 230 | 460 | 695 |
| 10 | 7 | -10 | 55 | -28 | 262 | 504 | 793 |
| 10 | 9 | -10 | 65 | -16 | 294 | 548 | 891 |
| 10 | -5 | -10 | -5 | -100 | 100 | 285 | 280 |
| 5 | -1 | -5 | 5 | -41 | 65 | 168 | 197 |
| 5 | -2 | -5 | 0 | -47 | 55 | 151 | 159 |
| 5 | 0 | -5 | 10 | -35 | 75 | 185 | 235 |
| 1 | 0 | -1 | 2 | -7 | 15 | 37 | 47 |

**5.3 Discussion on Results**

Face features are detected quite accurately using LBP. Figure 21,22,23,24 and 25 shows face features.

Use of threshold in LBP provides the opportunity for variation in accuracy according to the images’

requirements. The face features are enhanced by preprocessing so that they are distinct from their surroundings. The face features are represented geometrically by using the concept of chain code. The chain codes are used as input strings to the Needleman-Wunsch algorithm. The simplicity and accuracy of chain code is the reason for selecting the method for geometric representation. Needleman-Wunsch score in case of matching and mismatching image pair varies with the variation in input string lengths, arrangement of characters in input strings and variation in match score, mismatch score and gap penalty. Variations are shown in Table 1, 2, 3 and 4. Variation in input string lengths causing abrupt changes in score for which this characteristic is not robust (strong) and is yet to be tested on more image pairs.

**Chapter 6**

* 1. **Scope for future improvement**

The current work may be enhanced in future in the following directions.

1. Localization of features and kinship verification should be possible from face images with different facial expression, multiple postures and age differences.
2. Histogram equalization and histogram specification will be used on images so we can take into account the illumination of original images.
3. Refine the process of extraction of facial features using intensity profile and scan line approach.
4. More refinement is required on the method of selecting match, mismatch, and gap penalty scores for accuracy of Needleman-Wunsch score and hence that of kinship verification.

**6.2 Conclusion**

LBP is a texture descriptor which stores present local structures. It is more tolerant for monotonic illumination changes in image and it is simple to compute. LBP has been successfully used for many different image analysis tasks, such as facial image analysis, biomedical image analysis, aerial image analysis and image and video retrieval.

Here we are using LBP to verify kinship among individuals. The specialty of the work is that the approach may be improved by performing Histogram equalization and specification on the image in order to handle illumination conditions. The standard LBP based approach is enhanced with a flavor of geometric techniques and bio informatics based methods. Kinship verification models should be able to perform on faces with different facial expression, multiple postures and age differences. In order to develop more accurate kinship verification models, we plan on increasing the number of images in our database and conduct a performance evaluation, both on kin relation verification ability and key inherited facial feature identification.

**Annexure**

**References**

**[1].** Timo Ahonen, Abdenour Hadid, and Matti Pietikaeinen. Face Description with Local Binary Patterns to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol 28. pages 2037--2041. 2006.

**[2].** Ruogu Fang, Kevin Tang, and Noah Snavely and Tsuhan Chen. Towards Computational Models

Of Kinship Verification. In proceeding of 17th IEEE Conf. Image Process. Hong Kong. pages

1577-1580.2010.

**[3].** Di Huang, Caifeng Shan, and Mohsen Ardabilian. Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. IEEE Transaction on Systems, Man, and Cybernetics—PART C: Applications and Reviews, vol. 41, NO. 6. November 2011.

**[4].** Guodong Guo and Xiaolong Wang. Kinship Measurement on Salient Facial Features.IEEE Trans. Instrumentation and Measurement . vol 61. pages 2322--2325.2012.

**[5].** Abdelhamid Abdesselam. Improving Local Binary Patterns Techniques by Using Edge Information. Lecture Notes on Software Engineering. Vol. 1, No. 4, November 2013.

**[6].** Du Hua, Gu Guangda, and Su Cheng. Feature points extraction from Faces. Image and Vision Computing NZ. Palmerston North. pages 154--158. 2003.

**[7].** Amit Satpathy and Xudong Jiang. LBP-Based Edge-Texture Features for Object Recognition. IEEE Transaction on Image Processing. vol. 23, No. 5, May 2014.

**[8].** Ernesto Bribiesca. A new Chain code. Pattern Recognition. Vol. 32. Issue 2. Pages 235- 251.1999.

**[9].** Saul B. Needleman and Christian D. Wunsch. A General Method Applicable to the Search for

Similarities in the Amino Acid Sequence of Two Proteins. Journal of Molecular Biology. Vol. 48.

Pages 443-453.1970.