Chapter – 6 Palm and Nail Color Analysis

6.1 Introduction
To perform palm color analysis and nail color analysis the first step is to extract palm and nail from the palm image. This chapter explains the algorithms to extract palm from background of image, and to extract nails from rest of the palm. It also discusses palm and nail color analysis.

6.2 Palm Extraction
To analyze the palm for prediction of the disease, it is necessary to understand the color of palm first. Moreover, not only for palm color analysis, but also for nail color analysis and mount analysis, the region of palm must be separated from rest of the image. Section 6.2.1 explains the process of extraction of the region of palm from its background.

6.2.1 Separation of Palm from background
The logic for separation of palm from background is based on the Von Luschan's chromatic scale for different skin colors. This scale is used in dermatology. Von Luschan's chromatic scale is a method of classifying skin color. It is also called the von Luschan scale or von Luschan's scale. It is named after its inventor, Felix von Luschan. The equipment consists of 36 opaque glass tiles which were compared to the subject's skin, ideally in a place which would not be exposed to the sun (such as under the arm). The von Luschan scale was used to establish racial classifications of populations according to skin color; in this respect it is in contrast to the Fitzpatrick scale intended for the classification of the skin type of individuals introduced in 1975 by Harvard dermatologist Thomas B. Fitzpatrick to describe sun tanning behavior.

The von Luschan scale was used extensively throughout the first half of the 20th century in race studies and anthropometry. However, it was considered problematic, even by its practitioners, because it was very inconsistent. In many instances, different investigators would give different readings of the same person. The von Luschan scale was largely abandoned by the early 1950s, replaced instead by methods utilizing reflectance spectrophotometry [26]. Following figure shows Von Luschan's chromatic scale [26].
One can see in figure 6.1 36 different skin colors of human beings are found in the world. According figure 6.1 scale of skin tones which has been used now days in dermatology. There are six types of skin colors [26]:

<table>
<thead>
<tr>
<th>Fitzpatrick type</th>
<th>von Luschan scale</th>
<th>Also called</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1–5</td>
<td>Very light or white, &quot;Celtic&quot; type</td>
</tr>
<tr>
<td>II</td>
<td>6–10</td>
<td>Light or light-skinned European</td>
</tr>
<tr>
<td>III</td>
<td>11–15</td>
<td>Light intermediate, or dark-skinned European</td>
</tr>
<tr>
<td>IV</td>
<td>16–21</td>
<td>Dark intermediate or &quot;olive skin&quot;</td>
</tr>
<tr>
<td>V</td>
<td>22–28</td>
<td>Dark or &quot;brown&quot; type</td>
</tr>
<tr>
<td>VI</td>
<td>29–36</td>
<td>Very dark or &quot;black&quot; type</td>
</tr>
</tbody>
</table>

Table 6.1: correlation between Von Luschan scale and Fitzpatrick scale types

In the Indian subcontinent, the skin color of people normally falls between 15 to 28, as per Von Luschan scale.

To extract the palm from background of scanned image, a model is designed as below.
6.2.2 Extraction of the Palm from Background

To extract the area of palm from background of the image, a model is designed and developed as shown in figure 6.2.

![Figure 6.2: Model for Extraction of Palm from Background](image)

The model takes scanned image of human palms of front side as an input. Extraction of palm from background is a three steps process:

(i) Separation of palm area from rest of the image

(ii) Rotation of palms if required

(iii) Cropping of palm from rest of the area

For performing all the steps, different algorithms have been developed

(i) **Separation of palm area from rest of the image:**

An algorithm is designed and implemented to separate the palm area from rest of the image.

**Algorithm 6.1: Extraction of Palm from Background**

**Observation:** It is observed from the above scale of human skin color, that when the RGB components for human skin color are considered, red component value is greater than at least 10 than green component value, and red component value is greater than at least 20 than blue component.

**Input:** The processed image of front side of human palm (left or right) as mentioned in Chapter 5.
Chapter – 6 Palm and Nail Color Analysis

Step 1: for each pixel in the image, perform the following test

If [(red color component value > green color component value + 10) and (red color component value > blue color component value + 20)]

Accept the pixel as a pixel of palm

Else

Reject the pixel by considering as a background and change the color of the pixel to yellow (255, 255, 0).

Code to implement the algorithm:

```csharp
// for each line
for (int y = 0; y < height; y++)
{
    // for each pixel
    for (int x = 0; x < width; x++)
    {
        { Inpoints.Add(new IntPoint(x, y));
        }
        else
        {
            System.Drawing.Color.Yellow;
        }
    }
}
```

The pixels which are not belonging to the palm region are colored as yellow.

Another important reason behind converting background color as yellow is that, when the user scans the palm, all the background color pixels are not having same color value. For example, in the figure shown in results, it seems that the background of input image is black. But when one analyzes the color values of each pixel, they are not same. There may be color variation between neighboring pixels also. So if one converts the color of background pixels into a uniform color to all, then the further processing becomes easier.

The reason behind choosing yellow color as new color to the background is that, the yellow color with RGB value (255, 255, 0) is never found in human palm. So, there will be no conflict between palm color and background color.
Chapter – 6 Palm and Nail Color Analysis

Result of the algorithm:

Figure 6.3 shows input and output images of palm. In the output image, the background color is converted to yellow.

<table>
<thead>
<tr>
<th>Input Image</th>
<th>Output Image</th>
</tr>
</thead>
</table>

![Input Image](image1.jpg) ![Output Image](image2.jpg)

**Figure 6.3: Result of the palm extraction algorithm**

At first glance, in the input figure, it may look like that background is black, but after observing carefully; one can see that the background color is not same at all places. Moreover, since surface of scanner may have some stains on it, it has also created noise in this input image. The algorithm removes all these noises and provides uniform background.

As shown in figure 6.3, the algorithm 6.1 can handle stains on the surface of the flat bed scanner.

While scanning palm, suppose there is any other object on the surface of the scanner, the algorithm does not consider the object as a region of palm. Figure 6.4 shows the result of the algorithm while putting marker pen on the scanner, along with the palm. The algorithm is not considering the marker as a palm, so it is removed from output image. Of course, it is not desired that someone may scan other object with palm, but yet, the algorithm is capable to manage this case.
There are three things to notice in figure 6.4:

(i) Removal of marker pen: the area of market pen is almost 90% removed from the scanned image by the algorithm.

(ii) Removal of cuff of the shirt: the region of cuff of shirt is also removed from the input image and only skin area is kept for further processing.

(iii) Removal of the Ring: from ring finger the user has scanned palm by wearing the ring in ring finger. The algorithm has successfully removed the area of palm covered by ring.

One more common mistake that any user can do while scanning the palm. The user may wear wrist watch in hand and scan the palm. The algorithm 6.1 successfully removes the area of hand covered by wrist watch. Figure 6.5 shows the result of the same case.
### Input Image

![Input Image]

### Output Image

![Output Image]

<table>
<thead>
<tr>
<th>Input Image</th>
<th>Output Image</th>
</tr>
</thead>
</table>

Figure 6.5: Result of Noise Removal by Algorithm 6.1

Figure 6.4 and 6.5 demonstrates the noise removal capacity of the algorithm 6.1: extraction palm from background.

**(ii) Rotation of palm:**

Image rotation for nail color is optional in this model. If the prototype finds that the image of palm is scanned horizontal instead of vertical, this step is performed. Image rotation technique is explained in the section 5.2.

**(iii) Cropping of palm from rest of the area:**

Since the analysis is to be done for only palm region, to reduce the pixel to be processed and to simplify the processing, the resulting images after applying algorithm 6.1 are cropped using cropping algorithm. The goal of cropping is to get the four points shown in figure 6.6 named topmost point, bottommost point, leftmost point, and rightmost point. Using these four points, image is cropped. To understand the cropping algorithm properly, consider the image displayed in following figure 6.6 for reference.
Fig 6.6: Points and Directions of interest for CropAlgorithm

Cropping algorithm is as follows:

**Algorithm 6.2: Cropping of Palm from rest of the area**

**Input:** Output image of algorithm 6.1.

**Steps:**

1. Define \( \text{max}_x = \text{max}_y = (0,0) \) and \( \text{min}_x = \text{min}_y = (\text{image}\_\text{width}, \text{image}\_\text{height}) \) as initial values for rightmost point, bottommost point, leftmost point, and topmost point respectively.

2. For each row of pixels follow the direction shown in figure. The first point of the palm region encountered would be the topmost point (the point with minimum value of y-coordinate). Save this point as “\( \text{min}_y \)”. Proceeding in the same fashion, the last point (the point with highest value of y-coordinate) of palm region in vertical direction is also obtained. Save this point as “\( \text{max}_y \)”.  
3. For each column of pixels follow the direction shown in figure. The first point of the palm region encountered would be the leftmost point (the point with minimum value of x-coordinate). Save this point as “\( \text{min}_x \)”. Proceeding in the same fashion,
Chapter – 6 Palm and Nail Color Analysis

the last point (the point with highest value of x-coordinate) of palm region in vertical direction is also obtained. Save this point as “max_x”.

4. Get four end points of palm region leftmost point = x-coordinate of min_x; rightmost point = difference between x-coordinates of max_x and min_x; topmost point = y-coordinate of min_y; bottommost point = difference between y-coordinates of max_y and min_y.

5. Crop the rectangle using these four points, which will touch the four boundaries of palm.

The code to implement algorithm 6.2 (cropping algorithm) is as below:

```csharp
IntPoint maxy = new IntPoint(1, 1); //declaration of bottommost point
IntPoint miny = new IntPoint(1000, 1000); //declaration of topmost point
IntPoint maxx = new IntPoint(1, 1); //declaration of rightmost point
IntPoint minx = new IntPoint(1000, 1000); //declaration of leftmost point
foreach (IntPoint i in imgpoints)
{
    //getting maximum and minimum coordinate values
    if (i.Y > maxy.Y)
    {
        maxy = i;
    }
    if (i.Y < miny.Y)
    {
        miny = i;
    }
    if (i.X > maxx.X)
    {
        maxx = i;
    }
    if (i.X < minx.X)
    {
        minx = i;
    }
}
//-- Cropping the palm image only
Bitmap cropped = cfil.Apply(img);
cropped.Save(folder + "\\" + lbluids.Text + "L3croppedimg.jpg");
```

The result of algorithm 6.2 is shown in figure 6.7.
Algorithm 6.1 and algorithm 6.2 are executed on all four images of scanned palm. Thus, after applying algorithm 6.1 and algorithm 6.2 on the front palm images for left and right palms, the prototype performs palm color analysis.

### 6.2.3 Palm color analysis

To analyze the color of palm, the prototype analyzes color of each pixel on the region of palm. In the cropped image of palms, there are only two sets of pixels:

(i) Yellow pixels with RGB component value (255, 255, 0) and
(ii) non-yellow pixels which are the pixels of palm.

The prototype analyzes only non-yellow pixels. Following algorithm is designed and implemented for palm color analysis.
Algorithm 6.3: Palm Color Analysis

**Input:** Output image of algorithm 6.2.

**Steps:**

1. Get the color of pixel in terms of RGB component values.

2. Compare the value of each component with the value stored in knowledge base. While comparing value of each component consider the deviation for each component as stored in knowledge base as mentioned in section 4.3 of chapter 4.

3. If the match is found, consider this pixel as the pixel of mentioned color and increase the counter for this color by one (the counter for each color is initialized by 0). The value of counter for each color will give number of pixels with respective color.

4. Count the percentage of pixels with each color. If the percentage of pixels found with given color is less than 5%, the case is not considered. If the percentage of pixels found with given color is between 5% to 10%, then the level of disease is 0, which shows probability of disease. If the percentage of pixels found with given color is between 11% to 30, then the level of disease is 1, which shows initial stage of disease related to respective color. If the percentage of pixels found with given color is more than 30%, then there are strong chances of materialization of the disease associated with respective color.
Chapter – 6 Palm and Nail Color Analysis

Following is the sample code to identify disease associated with white color of palm.

// PROCESSING RIGHT PALM

System.Drawing.Bitmap oimg = new System.Drawing.Bitmap("D:\PhD Project\DSS\UploadedPalmImages\" + widthcropped);

for (int y = 0; y < oimg.Height; y++)
{
    for (int x = 0; x < oimg.Width; x++)
    {
        {
            RightPalm.Add(new IntPoint(x, y));
        }
    }
}

foreach (IntPoint RP in RightPalm)
{
    int current_red = oimg.GetPixel(RP.X, RP.Y).R;
    int current_green = oimg.GetPixel(RP.X, RP.Y).G;
    int current_blue = oimg.GetPixel(RP.X, RP.Y).B;
    // checking for white nails
    if (current_red <= Whitew && current_red >= Whitew)
    {
        int WhiteNailcount++;
        oimg.SetPixel(RP.X, RP.Y, Color.Green);
    }
}

// CALCULATION OF TOTAL PIXELS WITH DIFFERENT COLORS IN BOTH THE PALMS

int totalwhite, totblue, totyellow, totblueyolo, totred, totredyolo; // variables to hold total colored pixels for different colors
int total_pix both_palm;
float whiteper, blueper, yellowper, blueyoloper, redper, redyoloper; // variables to hold percentage of each colored pixels
totalwhite = WhiteNailcount + 1; WhiteNailcount;

DSS for Healthcare on the basis of Medical Palmistry through Digital Image Processing and Analysis   Page 98
Chapter – 6 Palm and Nail Color Analysis

```csharp
        total_pix_both_palm = RightPalm.Count + LeftPalm.Count;
        whiteper = (100 * totwhite) / total_pix_both_palm;
        int normalcount = 0; // counter to identify healthy palm
        // PREDICTION FOR WHITE PALM
        if (whiteper > 5 && whiteper < 10)
        {
            Wtext.Width = 400;
            Wtext.Height = 100;
            Wtext.Font.Bold = true;
            Wtext.Text = "the system has found that " + Convert.ToString(whiteper) + " percent area of your palms is coloured white." + WL0;
            Wtext.TextMode = TextMode.Multiline;
            pnlPrediction.Controls.Add(Wtext);
            normalcount++;
        }
else if (whiteper >= 10 && whiteper <= 30)
{
            Wtext.Width = 400;
            Wtext.Height = 90;
            Wtext.Font.Bold = true;
            Wtext.Text = "the system has found that " + Convert.ToString(whiteper) + " percent area of your palms is coloured white." + WL1;
            Wtext.TextMode = TextMode.Multiline;
            pnlPrediction.Controls.Add(Wtext);
            normalcount++;
        }
else if (whiteper > 30)
{
            Wtext.Width = 400;
            Wtext.Height = 100;
            Wtext.Font.Bold = true;
            Wtext.Text = "the system has found that " + Convert.ToString(whiteper) + " percent area of your palms is coloured white." + WL2;
            Wtext.TextMode = TextMode.Multiline;
            pnlPrediction.Controls.Add(Wtext);
            normalcount++;
        }
```

This is a sample partial code. The algorithm is implemented for the palms with colors white, blue, light yellow, dark yellow, light red, and dark red. In the above code the predictions are fetched from knowledgebase. For example, “WL0” is a string which stores the prediction fetched from knowledge base for white color and level 0 for the palm. Section 6.2.4 demonstrates the results obtained by executing the above code.

### 6.2.4 Results

Figure 6.8 shows the output of the process of palm color analysis. There are three major components of output. (i) The input and output images, (ii) the statistical analysis, (iii) textual Prediction.
Figure 6.8 (a) and (b): Result of Palm Color Analysis

(i) The input and output images: the output of palm color analysis is shown in pictorial format for both the palms. The images of left and right palms are colored according to the color of pixels at respective places. The displayed color in output image and original color of respective pixels is mentioned in neighboring table on the same page. This component gives visual effect of any abnormality found in palm. Simply, if in output image any region of palm is found colored differently, user should refer the neighboring table to get the meaning.
(ii) The statistical analysis: the table displayed in the right side shows statistical description of palm color analysis. The table explains meaning of each color painted in output image by giving actual color name of that region. The table also shows number of pixels found with each color. In the end, the table shows percentage of pixels found with each color. Based on these values only, the predictions are made by the prototype.

(iii) Textual Prediction: the output screen also has a section named “Conclusion of Analysis”, which displays the prediction made by the prototype in textual format. The predictions contain the name of the disease if found, its stage, and if necessary it gives advice.

6.3 Nail Extraction

Nail color indicates health status of a person. The prototype analyzes the color of nail and using knowledge base, results are displayed. To analyze the nails, the prototype uses images of back sides of both the palms. Since, nails of both the thumbs are not clearly visible in the scanned images; the prototype does not analyze nail of thumb. Before starting analysis of nails, the prototype has to distinguish between nail region and rest of the palm accurately. This separation of nails from palm region is explained in section 6.3.1.

6.3.1 Separation of Nails from Palm

To separate the nails from rest of the palm, an algorithm is design and implemented successfully. To start the process, both the human palms are scanned i.e. left and right, from front and back side, using scanner. Scanned images of back side of both the palms are taken and after execution of algorithm 6.1 and 6.2 these images are taken as input for this process.
Algorithm 6.4: Separation of Nails from Palm [16]
Input: Back side images of right and left palms.

Assumption
By performing experiments on scanned back side palm images it is observed that, the pixels at the boundary of nail, i.e. from where nail ends and finger skin begin, are darker than skin color and nail color. So in this algorithm, it is assumed that pixels at the boundary of nail are darkest in the image, i.e. their RGB values are least among all pixels on that scan line.

Data variables used in algorithm:

1. TOP = the top most point of the image that is the first pixel from top side whose color value is not yellow. For this point (x,y), the value of y-coordinate will always be 0. See figure 6.9.
2. LEFT = the left most point of the image that is the first pixel from left side whose color value is not yellow. For this point (x,y), the value of x-coordinate will always be 0. See figure 6.9.
3. RIGHT = the right most point of the image that is the last pixel from left side whose color value is not yellow. For this point (x,y), the value of x-coordinate will be maximum of x-axis. See figure 6.9.
4. SECOND = the top most point of second finger from left. For right palm, it would be for index finger, and for left palm it would be for ring finger.
5. FOURTH = the top most point of fourth finger from left. For right palm, it would be ring finger, and for left palm it would be for index finger.
6. SKINCOLOR = average RGB color value of back side of human palm
Chapter – 6 Palm and Nail Color Analysis

Steps

1. Set \( y = 0 \). Scan the first row of the cropped image. Save the first pixel \((x, y)\) whose color value is not yellow as TOP. Draw a rectangle \( R_1 \) of size \( m \) by \( n \) pixels downwards keeping TOP as a center of upper boundary of rectangle. Here, \( m \) is width and \( n \) is height of rectangle. This will give us area which contains nail of tallest finger of palm.

2. Set \( x = 0 \). Scan the image row by row. Save the \( y \)-coordinate value of the first pixel whose color is not yellow as LEFT. Draw a rectangle \( R_2 \) of size \( m \) by \( n \) pixels rightwards keeping LEFT as a center of left boundary of rectangle. This will give us area which contains nail of leftmost finger of palm.

3. Set \( x = \) image width. Scan the image row by row. Save the \( y \)-coordinate value of the first pixel whose color is not yellow as RIGHT. Draw a rectangle \( R_3 \) of size \( m \) by \( n \) pixels leftwards keeping RIGHT as a center of the right boundary of rectangle. This will give us area which contains nail of rightmost finger of palm.

4. Set \( x = y = 0 \). Scan the image row by row in the rectangular region formed by four lines; the horizontal line (bottom side) that passes through coordinates \((x_1, y_1)\), where \( x_1 = 0 \), \( y_1 = \text{LEFT}.y - n/2 \) and the vertical line (right side) that passes through coordinates \((x_2,y_2)\), where \( x_2 = \text{TOP}.x - m/2 \), and \( y_2 = 0 \). The top and left sides of this rectangular region would be upper and left boundaries of cropped image respectively. Save the first pixel \((x_i,y_i)\) whose color value is not yellow as SECOND. Draw a rectangle \( R_4 \) of size \( m \) by \( n \) pixels downwards keeping SECOND as a center of upper boundary of rectangle. This rectangle will give us area which contains nail of second finger from left.

5. Set \( x = y = 0 \). Scan the image row by row in the rectangular region formed by four lines; the horizontal line (bottom side) that passes through coordinates \((x_3, y_3)\), where \( x_3 = 0 \), \( y_3 = \text{RIGHT}.y - n/2 \), the vertical line (left side) that passes through coordinates \((x_4,y_4)\), where \( x_4 = \text{TOP}.x + m/2 \), \( y_4 = 0 \). The top and right sides of this rectangular region would be upper and right boundaries of cropped image respectively. Save the first pixel \((x_k,y_k)\) whose color value is not yellow as FOURTH. Draw a rectangle \( R_5 \) of size \( m \) by \( n \) pixels downwards keeping FOURTH as a center of upper boundary of rectangle. This rectangle will give us area which contains nail of fourth finger from left.

6. For each \( R_i \), where \( i = 1 \) to \( 5 \), repeat step 7.
Chapter – 6 Palm and Nail Color Analysis

7. Scan pixels within $R_i$ row by row and repeat following steps for ‘n’ scan lines, where $n =$ height of rectangle in terms of number of pixels. Initialize line_number by 0:
   i. If line_number == n, or if pixel color is yellow, or if pixel color is SKINCOLOR, do nothing.
   ii. Find the first pixel where RGB value suddenly decreases towards black store this pixel as $b_1$ (boundary of nail). If no such $b_1$ is found then stop.
   iii. On the same scan line, find the next pixel where RGB value suddenly increases towards white. From this pixel, select all pixels on this scan line and get their RGB color value, until another pixel $b_2$ is found, where RGB color value decreases towards black (opposite boundary of nail). line_number = line_number + 1. Go to step (i).

For implementation of this algorithm, the area highlighted by rectangle is divided in to four quadrants and each quadrant is processed separately.

**Result of Algorithm:**

The algorithm 6.3 is implemented successfully on different users, and the results are achieved. The results are shown in following figures. Figure 6.10 (a) shows scanned back side of left palm and figure 6.10 (b) shows the extracted nails from that image, which are highlighted by rectangle.

![Figure 6.10: Region of Nails Highlighted by Rectangles](image-url)
Figure 6.11 (a) shows back side of scanned right palm which is not purely horizontal or vertical. Even though, the algorithm 6.3 identifies the regions around nails. The resultant image is shown in figure 6.11 (b).

<table>
<thead>
<tr>
<th>(a) Scanned Back side of Right Palm with Random orientation</th>
<th>(b) Highlighted Region Around Nails by Rectangle</th>
</tr>
</thead>
</table>

Figure 6.11 (a) & (b): Identification of Nail Region in the Image with Random Orientation

The extracted nails can be highlighted with any color. Figure 6.12 (a) shows the scanned back side of right palm and figure 6.12 (b) shows the extracted area of nails, highlighted with blue color.

Analysis of nail color is explained in section 6.3.2.
6.3.2 Nail color analysis
The process of nail color analysis is very similar to the process of palm color analysis, explained in section 6.2.3 at algorithm 6.3. The only change is in the knowledge base. The knowledge bases are different for nails and palm colors. The algorithm is designed and implemented to make analysis of nails, except nail of thumb.

6.3.3 Results
The prototype works successfully to do nail color analysis. The results are shown in figure 6.13. The result of nail color analysis is displayed on a page in three parts.

(i) Pictorial Output: on the left side, input and output images of back sides of both the palms are displayed. In the output image, the area of nail is colored with specific colors, here with blue color. The meaning of each color is shown in neighboring table.

(ii) Statistical Output: The table on right side shows statistical analysis of the color of pixels with each color found in the area of nails.
(iii) Textual Output: In the bottom of right side of the page, the conclusion of analysis is displayed in textual format. This helps user to understand the results of analysis.

![Figure 6.13 (a) and (b): Result of Nail Color Analysis](image)

### 6.4 Summary

This chapter explains the two major functionalities of this prototype; palm color analysis and nail color analysis. The chapter also focuses on four algorithms used to extract region of palm from background, cropping of palm region, extraction of nails from palm image,
and algorithm for nail color analysis. The sample code to execute these algorithms and results are also explained in this chapter.