****

**PROJECT REPORT OF**

**COMPUTER APPLICATION IN ENGINEERING DESIGN LAB**

|  |  |
| --- | --- |
| **NAME:** | **Rabia**  **Zainab** |
| **ID:** | **FA17-BECE-2001**  **FA17-BECE-0020** |
| **TEACHER NAME:** | **Rafia Shaikh** |
| **SECTION:** | **AM(BE-CSE)** |
| **PROJECT TOPIC:** | **Hand Detection Using Image Processing** |

**INTRODUCTION:**

An RGB image, sometimes referred to as a true color image, is stored in MATLAB as an m-by-n-by-3 data array that defines red, green, and blue color components for each individual pixel. ... The precision with which a real-life image can be replicated has led to the commonly used term true color image.

Conversions to:

Skin Segmentation.

Gray Scale.

Binary Image.

**PROJECT DESCRIPTION:**

This project work focuses on the problem of gesture recognition in real time that sign language used by the community of deaf people. Research problem identified is based on Digital Image Processing using Color Segmentation, Skin Detection, Image Segmentation, Image Filtering, and Template Matching techniques. Our project is basically converting the image into skin segmentation through(Ycrcb) , into Gray scale and in binary image.

Skin Detection: There are several techniques used for color space transformation for skin detection. Some potential color spaces that are considerable for skin detection process are:

• CIEXYZ

• YCbCr

• YIQ

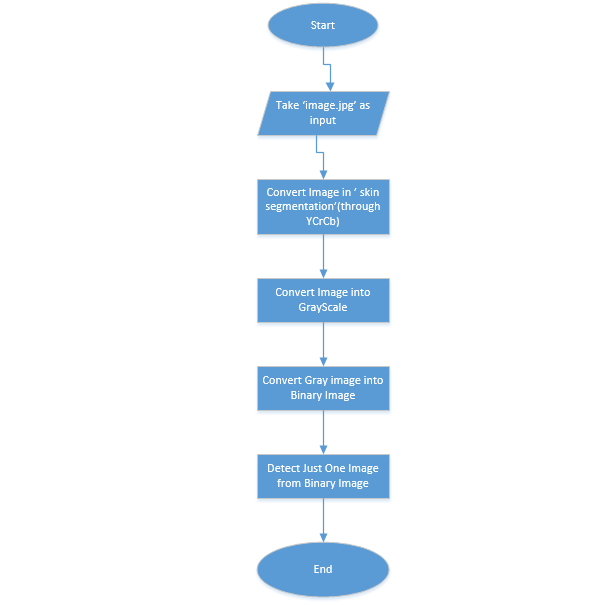
• YUV

Conversion to Gray Scale: In MATLAB, there is a function called rgb2gray() is available to convert RGB image to grayscale image. Here we will convert an RGB image to grayscale image without using rgb2gray() function.

Conversion to Binary Image: BW converts the grayscale image to binary image BW, by replacing all pixels in the input image with luminance greater than level with the value 1 (white) and replacing all other pixels with the value 0 (black).

A performance metric that the other color spaces have used is scatter matrices for the computation of skin and non-skin classes. Another drawback is to comparison through histogram of the skin and non-skin pixel after transformation of color space. The YCbCr, color space performs very well in 3 out of 4 performance metrics. Thus, it was decided to use YCbCr color space in skin detection algorithm.

**FLOW CHART:**



**PROJECT CODE:**

folder=('C:\Users\FA19-BEEE-0026\Downloads');

baseFileName=('image.jpg');

fullFileName=fullfile(folder,baseFileName);

format long g;

format compact;

fontSize = 20;

%IMAGE SEGMENTATION

img=imread(fullFileName);

img=rgb2ycbcr(img);

for i=1:size(img,1)

for j= 1:size(img,2)

cb = img(i,j,2);

cr = img(i,j,3);

if(~(cr > 132 && cr < 173 && cb > 76 && cb < 126))

img(i,j,1)=235;

img(i,j,2)=128;

img(i,j,3)=128;

end

end

end

img=ycbcr2rgb(img);

subplot(2,2,1);

image1=imshow(img);

axis on;

title('Skin Segmentation', 'FontSize', fontSize);

set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);

%SEGMENTED IMAGE TO GRAYIMAGE

grayImage=rgb2gray(img);

subplot(2,2,2);

image2=imshow(grayImage);

axis on;

title('Original Grayscale Image', 'FontSize', fontSize);

set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);

%GRAY TO BINARY IMAGE

binaryImage = grayImage < 245;

subplot(2, 2, 3);

axis on;

image3=imshow(binaryImage, []);

title('Binary Image', 'FontSize', fontSize);

% Label the image

labeledImage = bwlabel(binaryImage); % label the connected components in an image and assigning each one a unique label

measurements = regionprops(labeledImage, 'BoundingBox', 'Area');

for k = 1 : length(measurements)

thisBB = measurements(k).BoundingBox;

rectangle('Position', [thisBB(1),thisBB(2),thisBB(3),thisBB(4)],...

'EdgeColor','r','LineWidth',2 )

end

% Let's extract the second biggest blob - that will be the hand.

allAreas = [measurements.Area];

[sortedAreas, sortingIndexes] = sort(allAreas, 'descend');

handIndex = sortingIndexes(2); % The hand is the second biggest, face is biggest.

% Use ismember() to extact the hand from the labeled image.

handImage = ismember(labeledImage, handIndex);

% Now binarize

handImage = handImage > 0;

% Display the image.

subplot(2, 2, 4);

image4=imshow(handImage, []);

title('Hand Image', 'FontSize', fontSize);

**PROJECT OUTPUT:**

