Project Report:

Hand Gesture Recognition

Daksh Gupta (A20351161)
Saurabh Jadhav (A20359388)

18 November 2015

PROBLEM STATEMENT

The problem we will be addressing is the recognition of hand gestures. The data retrieved after recognition can be used for any application to which they are applicable.

PROPOSED SOLUTION

We will take a color input image of a palm from a source directory and pass it to the program as an external image. This image will be processed to extract the hand gesture. After the gesture is extracted we will display the gesture class on the screen

Here is an overview of how the image will be processed to extract the gesture

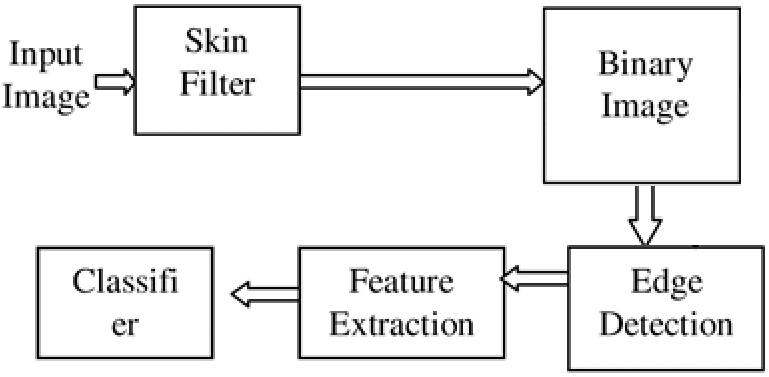


Fig. 1. Block diagram of the proposed approach

For skin filtering we will use the approach used in [1] and convert RGB image into a HSV image. The R, G, B values are divided by 255 to change the range from (0, 255) to (0, 1):

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

Hue calculation:

$$H=\begin{cases} 60° * \frac{G'-B'}{\Delta} mod\ 6, & C_{max} = R' \\ 60° * \frac{B'-R'}{\Delta} mod\ 6, & C_{max} = G' \\ 60° * \frac{R-G'}{\Delta} mod\ 6, & C_{max} = B' \end{cases}$$

Saturation Calculation:

$$S=\begin{cases} 0, & C_{max} = 0 \\ \frac{\Delta}{C_{max}}, & C_{max} \neq 0 \end{cases}$$

To generate a binary image we will use a binary threshold which modifies the image as follows:

$$imb(x, y) = \begin{cases} 1 & if\ im(x, y) > Thresh \\ 0 & otherwise \end{cases}$$

Where,

im is the original image
imb is the binary image generated
Thresh is the threshold used to generate the image.

Canny Edge detection will be used for detecting edges in the binary images obtained.

We will be using Euclidean Distance to find out the matches between test data and sample data. The Euclidean Distance will be calculated on each pixel in the two image. The minimum distance sample image will then determine the gesture of the test image.

IMPLEMENTATION DETAILS

OpenCV version: 2.4

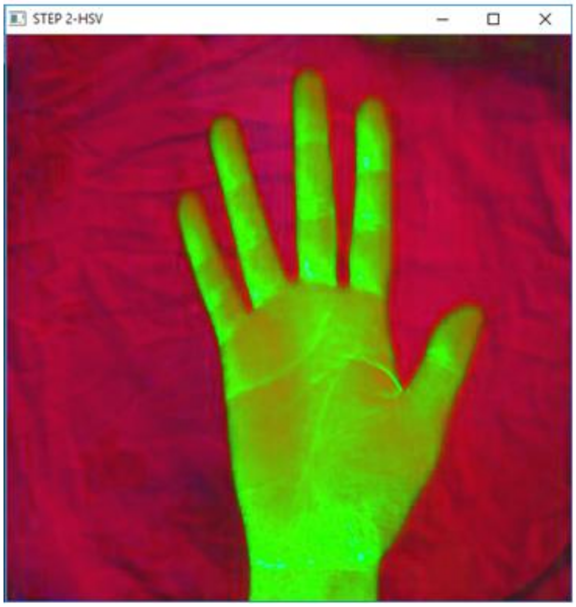
Python version: 2.7

We have used a source folder which will be passed as an argument to the program. Following are the steps performed

Converting to HSV color space

Detecting a hand in RGB color space is very difficult. RGB color space in sensitive to lighting conditions and hence is ill advised to use it. In HSV space we separate the hue, saturation and variance information.

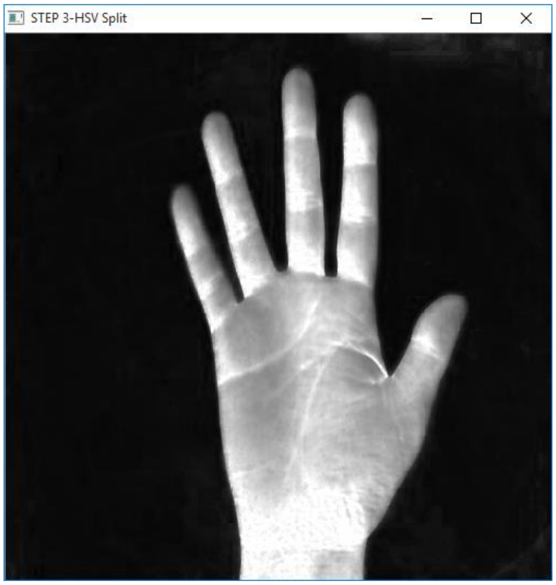
Conversion to HSV color space was achieved using the OpenCV function cvtColor with the constant COLOR_BGR2HSV



Splitting the HSV color Space

As the background of the image remains white, the saturation channel of the HSV color space gives the maximum information about palm. For this we split the image into 3 channels namely,

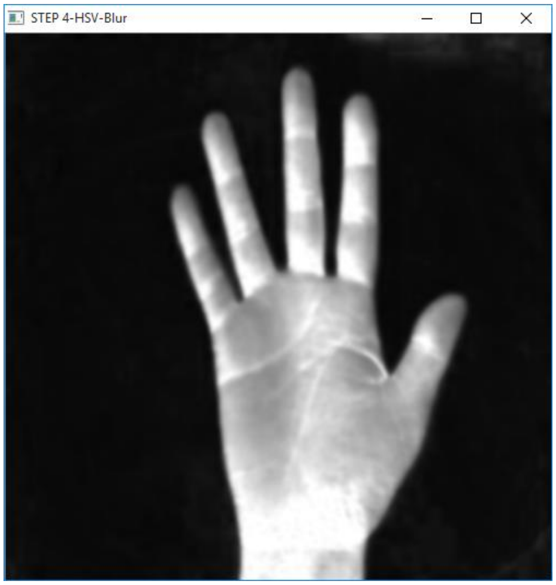
Hue, Saturation and variance. Saturation channel will be further used for processing



Applying Gaussian blur:

This step is useful to remove any noise from the image.

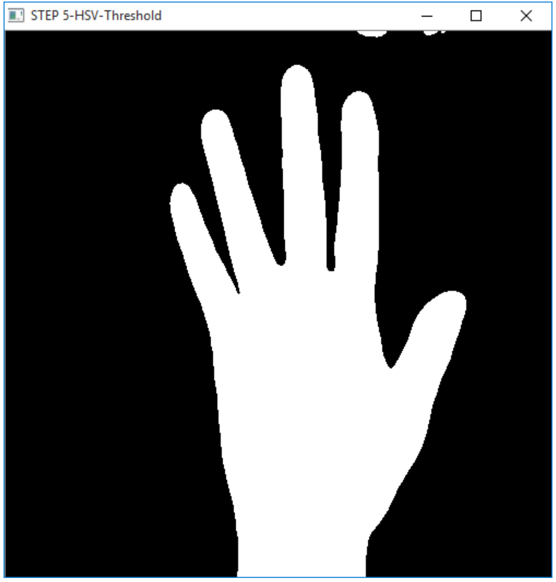
Inbuilt function *GaussianBlur* is used to perform this task



Normalizing the image

After reducing the noise we need to extract objects from the image which meet the certain threshold. For this we use the inbuilt OpenCV function *threshold* which gives control over upper and lower limit.

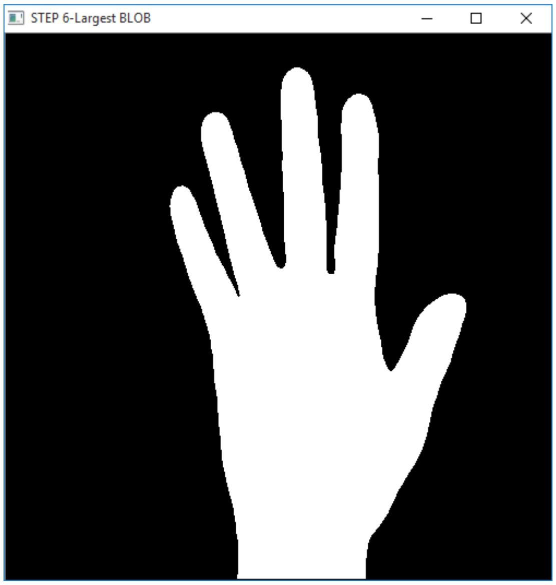
The threshold value is configurable and is stored in *prj.config.* The threshold value is currently set to 50. It can vary from 25 to 250. This value is dependent on the room conditions and camera. Adaptive thresholding would not work for our case since it calculated the threshold based on the neighborhood of the pixel.

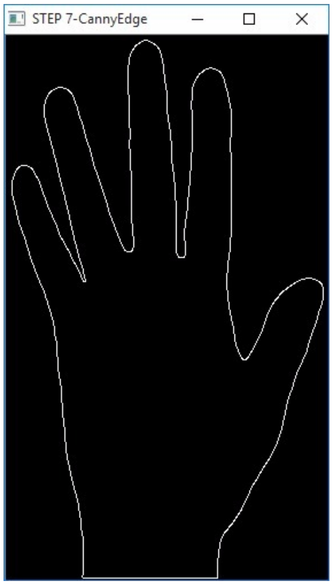


Finding the largest blob and edges:

After normalizing the image, there can be various objects in the image which are noise for gesture recognition. To remove such objects it is assumed that the largest object will be the required palm

To find the largest object, we used the function *findContours*. Using *contourArea* we find out the area of each contour and then select the largest amongst them.



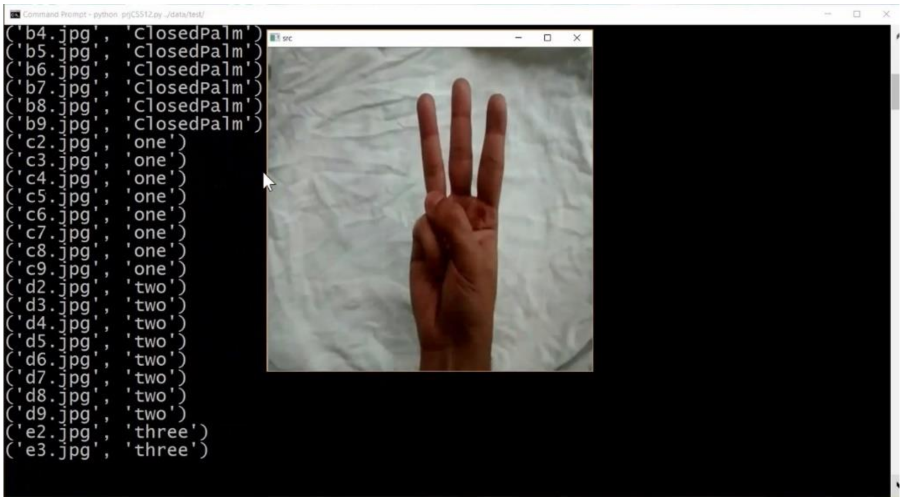


Finding Gestures

We have used Euclidean distance to recognize the gestures.

For this we find the distance between test image and sample images. Sample images are stored in folder /data/samples and are labeled according to the gestures. The folder contains sample images for each gesture that is to be recognized. The minimum distance between test image and sample images gives the required gesture.

The recog function is the Euclidean Distance classifier for the hand images.



```
('b4.jpg', 'ClosedPalm')
('b5.jpg', 'ClosedPalm')
('b6.jpg', 'ClosedPalm')
('b7.jpg', 'ClosedPalm')
('b8.jpg', 'ClosedPalm')
('b9.jpg', 'ClosedPalm')
('c2.jpg', 'one')
('c3.jpg', 'one')
('c4.jpg', 'one')
('c5.jpg', 'one')
('c6.jpg', 'one')
('c7.jpg', 'one')
('c8.jpg', 'one')
('c9.jpg', 'one')
('d2.jpg', 'two')
('d3.jpg', 'two')
('d4.jpg', 'two')
('d5.jpg', 'two')
('d6.jpg', 'two')
('d7.jpg', 'two')
('d8.jpg', 'two')
('d9.jpg', 'two')
('e2.jpg', 'three')
('e3.jpg', 'three')
```

ASSUMPTIONS

The background of the image is white

Image should contain only wrist and fingers

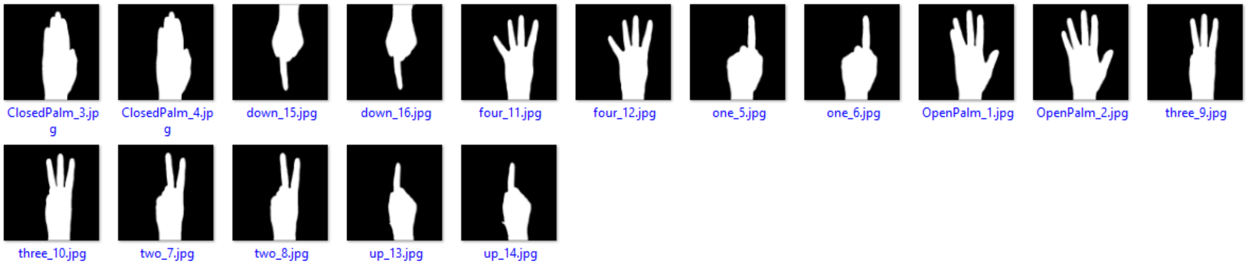
Only one hand is detected

GENERATING SAMPLE AND TEST DATA

The /data/images/ contains raw files captured from a camera

The sample data is generated by the program *createSample.py* with the "../data/images/" folder as the argument

This data can be used for finding the Euclidean distance with respect to test image and the gesture is then recognized



RESULTS

We used both edge images and binary images for the classification of gestures. We dropped the edge image because the results obtained were far better with binary images. Very minor changes to the code can bring back the edge image implementation and also be used to generate edge images of hands as sample data.

The sample images in /data/samples contains two images per gesture

For each gesture we tested eight images stored in folder /data/test/

Below is the result for edge images:

Gestures	Success Ratio (%)
Open Palm	0
Closed Palm	0
one	37.5
two	12.5
three	0
four	0
up	37.5
down	0

Below is the result for binary images:

Gestures	Success Ratio (%)
Open Palm	37.5
Closed Palm	100
one	100
two	100
three	100
four	100
up	100
down	100

CONCLUSIONS

We were able to successfully recognize hand gestures. The results obtained had a good success ratio for almost all the gestures. HSV can be used for skin filtering, however, the filtering is dependent on the background of the hand. Better techniques can be implemented to improve the result that we have obtained in this project.

BIBLIOGRAPHY

[1] Singha, J., Das, K.: Hand Gesture Recognition Based on Karhunen-Loeve Transform. Mobile & Embedded Technology International Conference 2013