|  |  |
| --- | --- |
| Name | Hassan Shahzad |
| Reg | 211798 |
| Class | BSCS 7C |

## In your report, briefly explain what you have done in each of the previous signature verification labs. Give a generalized overview in the beginning, list down the features used, and summarize your methodology. a. e.g. how you divided the image into segments, and what did you use (summation, variance, standard deviation, etc.) to get a single value from the differences calculated for each of the 64 segments.

Ans) Following are the features used in the signature verification:

1. Centroid
2. Black to white transitions
3. Aspect ratio
4. Number of black pixels
5. Normalized size
6. Angle of inclination of centroid
7. Normalized sum of angles of inclinations of black pixels

First of all, I loaded an image from the directory, convert it into a binary image and identify the boundary of the signature. Then all the feature extraction methods were applied on that portion of the image to calculate the given features. After these features had been calculated the centroid of this image is used to divide it into fours sub images. The program keeps on divided the image until 64 segments of the image are formed. The given features are calculated and stored for each image.

## 2. For each of the features used, make a heading of that feature, and under that heading:

## a. Define the feature.

## b. How did you calculate it? Show your algorithm in pseudocode or mathematical notation.

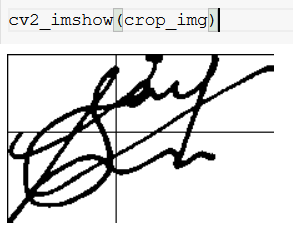
## c. Include visualizations you plotted for that feature if any.

### Centroid

Centroid is the point where center of mass of the signature image is located. It can be computed using the algorithm given below.

#### Algorithm

for x in range(width):  
 for y in range(height):  
 if img.getpixel((x,y)) == 0:  
 cx += x  
 cy += y  
 n += 1  
cx = int(cx/n)  
cy = int(cy/n)





### Black to white transitions

Find out how many times in the image if its pixels are traversed the transition from black color to white happens.

#### Algorithm

prev = img.getpixel((x1, y1))  
 n = 0  
 for p in range(x1, x2):  
 for q in range(y1, y2):  
 curr = img.getpixel((p, q))  
 if ((curr == 255) and (prev == 0)):  
 n += 1  
 prev = curr  
 return n

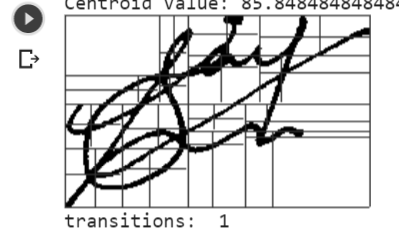
### Aspect ratio

Calculates the aspect ratio of the cell by dividing height by width

#### Algorithm

Width, height = Image.size

Return width/height



### Number of black pixels

Calculates the number of black pixels in the signature cell

#### Algorithm

Count = 0

For each pixel in the image:

If pixel value = 0

Count += 1

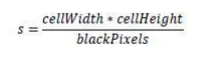
Return count

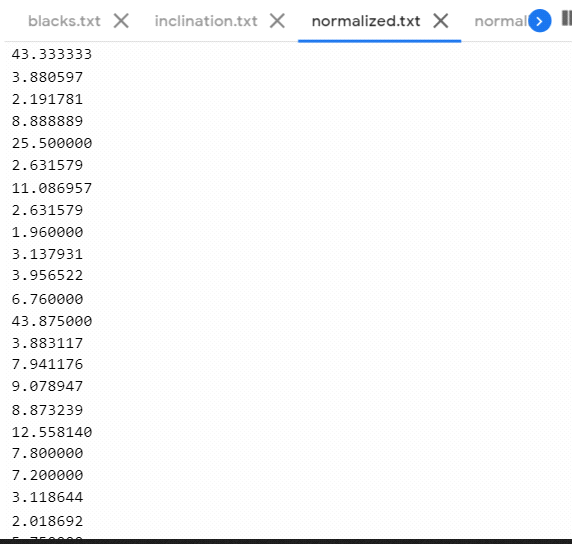
### 

### Normalized size

Size of each cell normalized by the number of black pixels in the cell

#### Algorithm

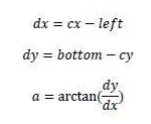


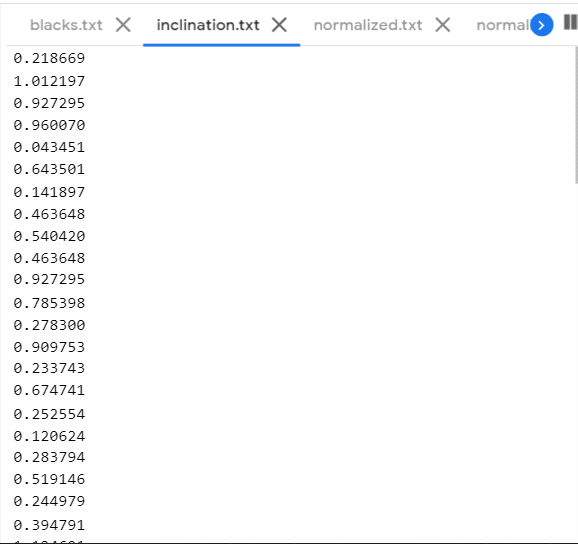


### Angle of inclination of centroid

### Calculates the angle of inclination of the centroid by computing angle of the line from centroid to the bottom-left corner of the cell.

#### Algorithm

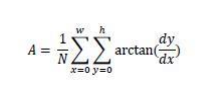


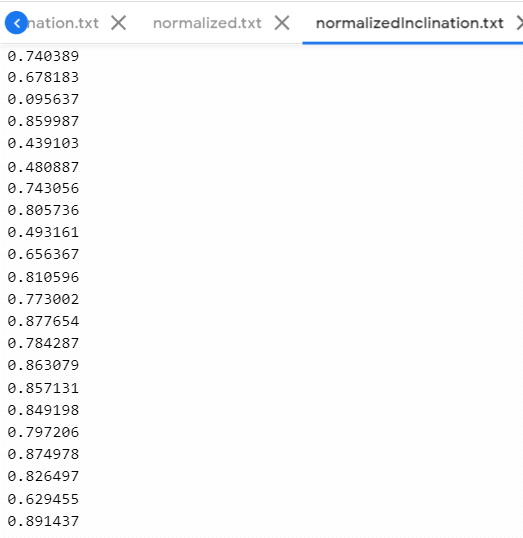


### Normalized sum of angles of inclinations of black pixels

It is the normalized sum of all the angels of inclination of all the black pixels in the cell

#### Algorithm





## 3. Identify top 3 features.

1. Normalized size
2. Aspect ratio
3. Centroid

## 4. Explain what you have learned from our series of labs on signature verification.

Feature extraction from the images, making data set.