Author's Name : Shriprasad Bhamare

email ID : sbhamar1@binghamton.edu

Purpose of Project

In the theory part,we got to know 4-,8-,m- adjacency and the shortest path from p to q based on 4-,8-,m- adjacency.In the programming part, we get familiarize with histograms and histogram creation. We can be able to give different effects to image like by using grey scale transformation function we can be able to determine the negative image of the given image and compare the histograms of the original and negative image. By using the histogram equalization algorithm ,we can be able see more clear picture by enhancing the image quality.In the region dection code,we get familiarize with the binary image conversion of original image and then detection of regions based upon the area of that region(number of pixels).This project gave us deep understanding of histograms,negative image,threshold value ,binary image conversion and region detection.

Method

Program1:

Grey-scale transformation

Subtract each intensity level of an image from 255. The negative transformation function is

s=L-1-r

Where ***s*** is the pixel after transformation, ***r*** is the pixel before transformation and ***L*** is the maximum intensity level.

Histogram equalization algorithm

Compute a scaling factor, $\alpha = 255$ / *number of pixels*
Calculate histogram of the image
Create a table *T*
for all remaining grey levels, *i*, **do**
  $T[i] = T[i-1] + \alpha * histogram[i]$
for all pixel coordinates, *x* and *y*, do
  $g(x, y) = T[f(x, y)]$

Plotting of histogram for respective Images

program2:

Optimal Threshold algorithm

Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels ≤ T

Compute the average grey levels of pixels in G_1 to give m_1 and G_2 to give m_2.

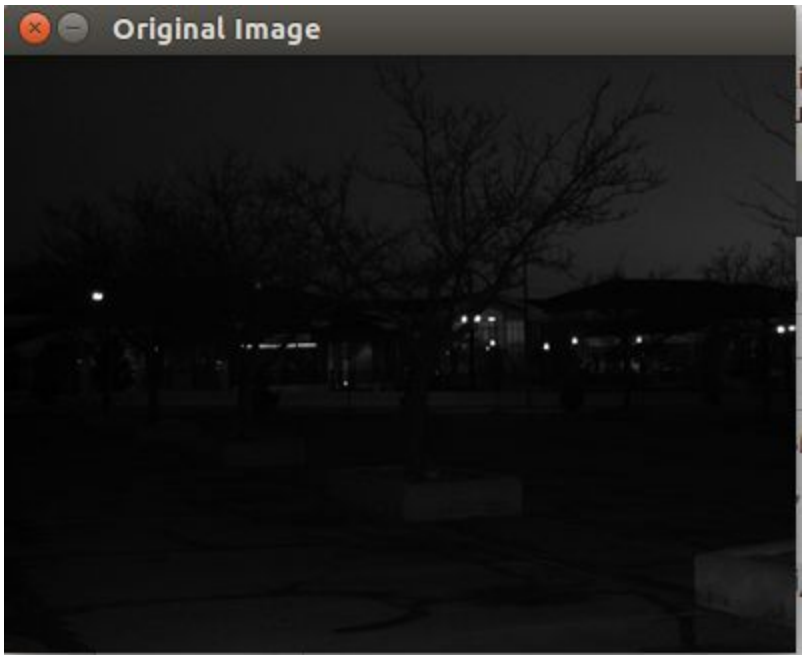
Compute a new threshold value:T = (m1+m2)/2

Repeat until the difference in T in successive iterations is less than a predefined limit T_∞

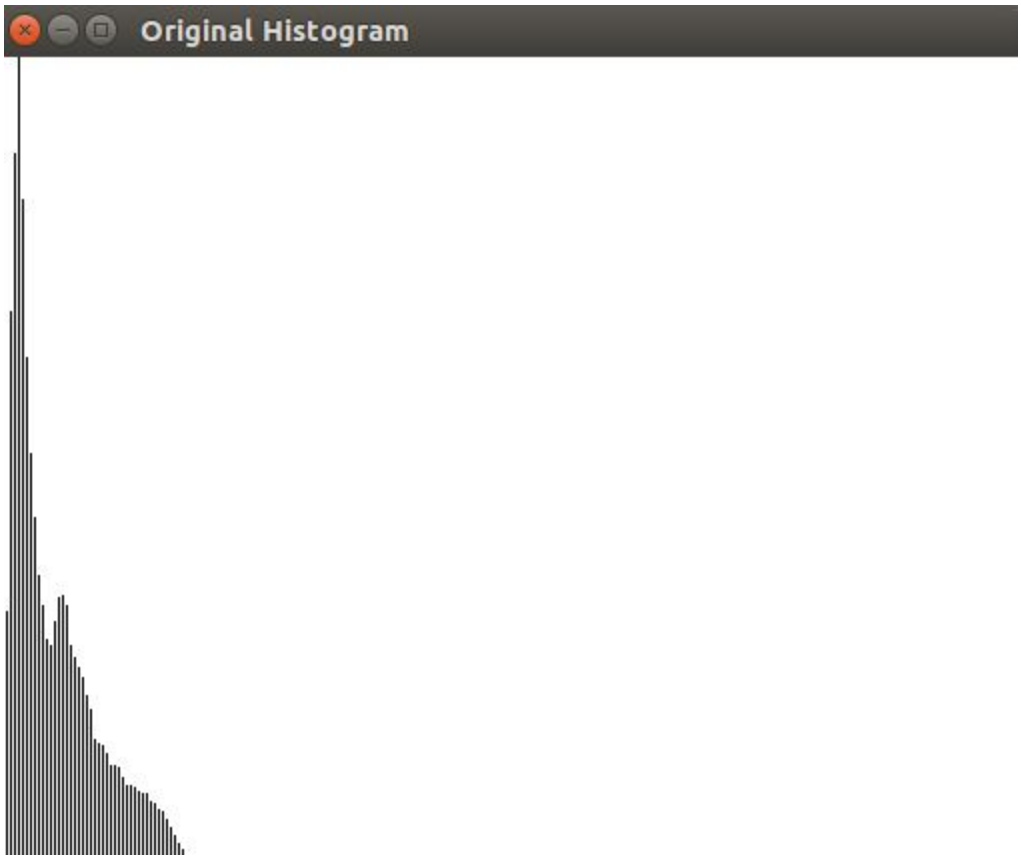
After calculating,threshold value, convert original image to binary image based on 0 , 255 value of pixels
Apply DFS,to get number of connected components.
Calculate Area of components and rank them as large and small
Get a pixel position for all the regions and call DFS
In DFS,change the color of pixel to 200,120,60 based on the area of region
Print image after detection of regions

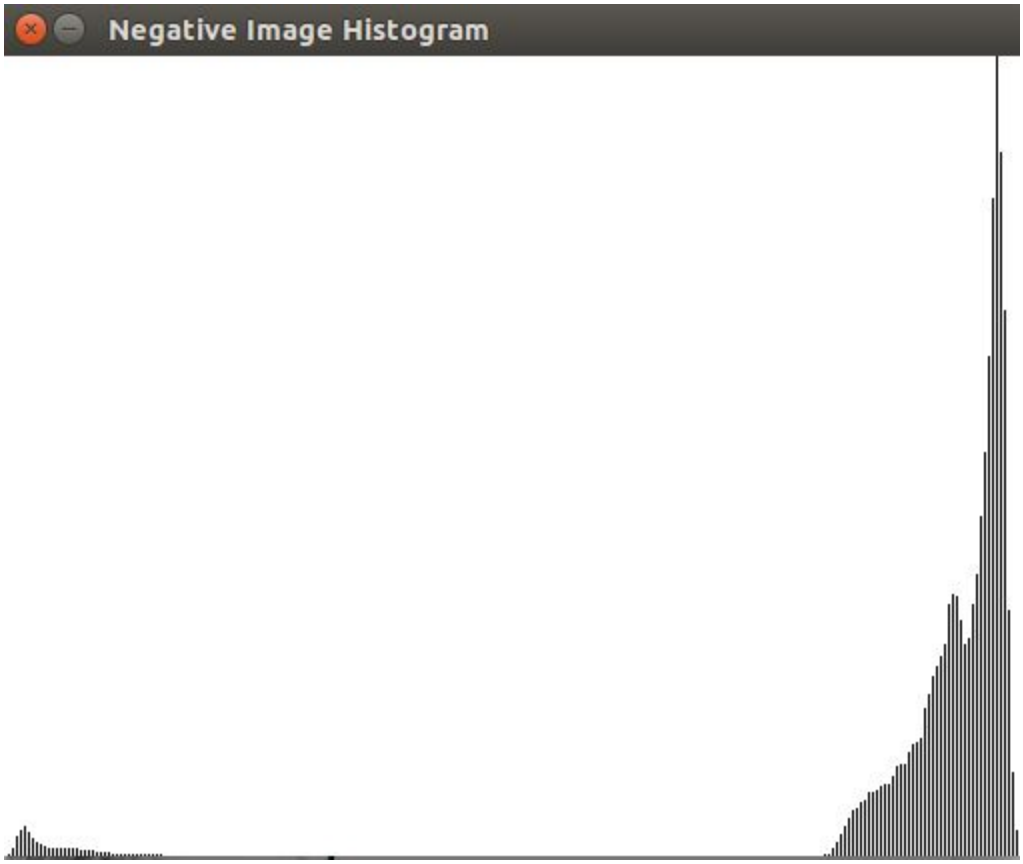
Results

1) Grey Scale Transformation-



2) Histogram of image-

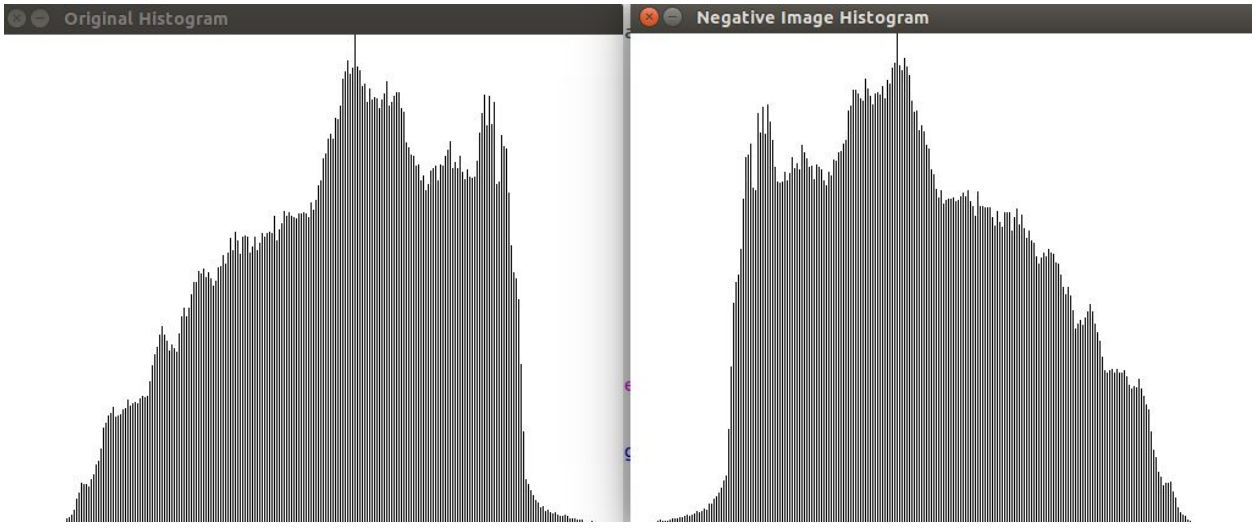


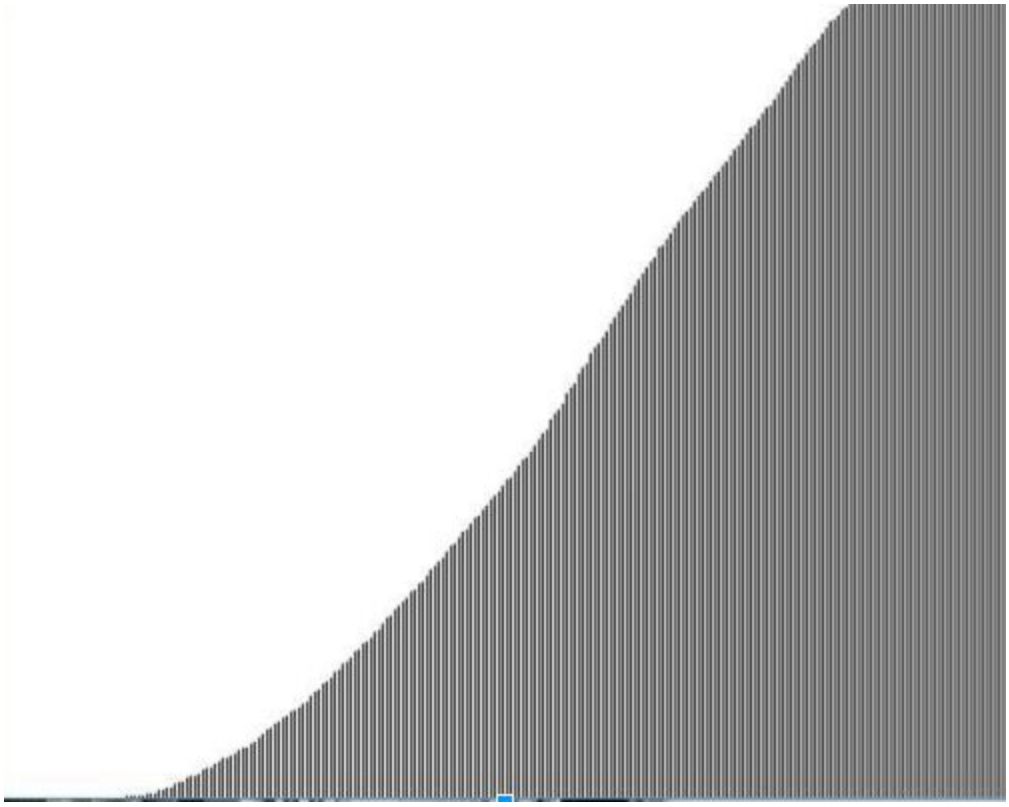


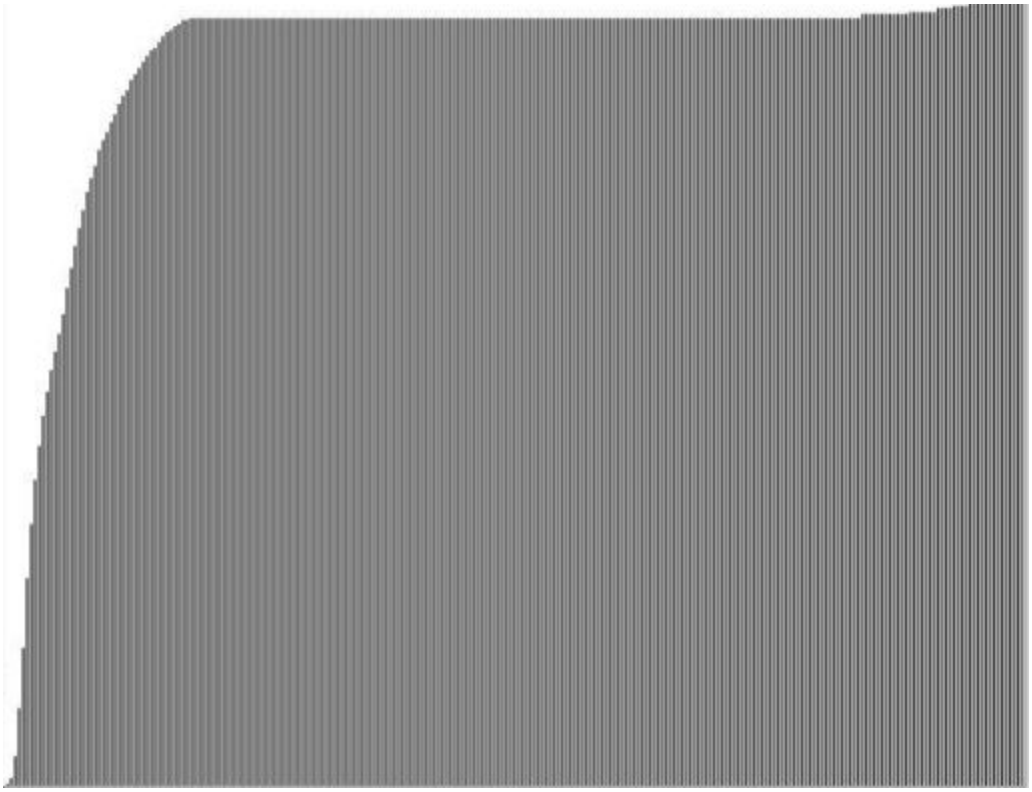
3) Histogram Equalization


Original Image

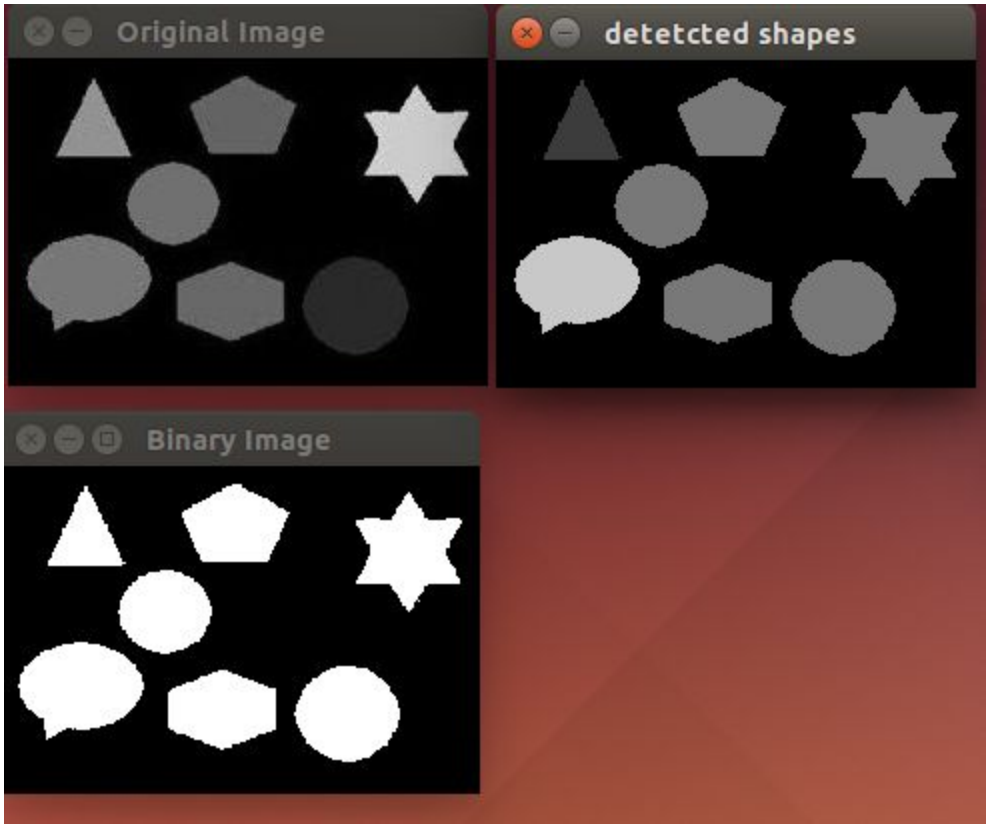

Negative Image


Original Histogram
Negative Image Histogram



- 

2.Region detection


Original Image
detetcted shapes
Binary Image


Binary Image
DON'T PANIC!
Original Image
DON'T PANIC!
detetcted shapes
DON'T PANIC!