**Assignment 1 Report**

**Contact Information:**

Name: Ved Ranade

Email ID: [vranade1@binghamton.edu](mailto:vranade1@binghamton.edu)

B-Number: B00715508

Phone: +1-607-232-1815

**Contents:**

1. Purpose of the Assignment
2. Algorithms Used

2.1. Algorithm to create negative greyscale image

2.2 Algorithm to obtain histogram of an image in array form

2.3 Algorithm to display histogram

2.4 Algorithm to equalize histogram

2.5 Algorithm to global threshold T

2.6 Algorithm to convert image to binary image

1. Running the programs
2. Results
3. Bug Report
4. **Purpose of the Assignment:**

The purpose of completing this assignment is to understand the following basic visual information processing concepts and to be able to apply them by implementing the concepts in Visual C++ programs:

* Image pixel connectivity
* Connected components concept
* Pixel operations
* Image histogram concept
* Histogram equalization and transformation
* Image thresholding
* Region detection

**2. Algorithms Used:**

2.1 Algorithm to create negative greyscale image:

Step 1: Load the image

Step 2: Scan each pixel in the image

* Change the value of the pixel to 255 – (value of pixel)

Step 3: Display the image

2.2 Algorithm to obtain histogram of an image in array form:

Step 1: Load the image

Step 2: Create an integer array of size [256] and initialize all its elements to zero

Step 3: For every pixel in the image, do:

* Obtain the intensity of the pixel
* Increment the value at the index of the array that corresponds to the pixel intensity

2.3 Algorithm to display histogram:

Step 1: Obtain the histogram array that is to be displayed

Step 2: Create a blank image having height of 400 pixels, width of 512 pixels

Step 3: Divide the height of the image into equal parts on the basis of the histogram array index that has the highest value

Step 4: For each index in the histogram array, draw a line from the base of the image created in step 2 up to the value of the histogram array index

2.4 Algorithm to equalize histogram:

Step 1: Find the probability of each pixel’s intensity value

Step 2: Generate the cumulative histogram of each pixel.

Cumulative histogram of a current pixel = cumulative histogram of previous + frequency of current pixel

Step 3: Calculate the cumulative distribution frequency (CDF) of each pixel

CDF of pixel 1 = cumulative histogram of 1/no. of pixels

Step 4: Calculate the final intensity value of each pixel by multiplying the CDF with 256 and rounding off the result

Step 5: Replace the value of each pixel by the value obtained in step 4

2.5 Algorithm to global threshold T:

Step 1: Select an initial threshold estimate as 128

Step 2: Divide the image into G1 and G2, where G1 contains all pixels with intensity < 128 and G2 contains all pixels with intensity > 128

Step 3: Calculate the average intensity values m1 and m2 for the pixels in G1 and G2

Step 4: Calculate the new threshold value T = (m1+m2)/2

Step 5: Repeat steps 2 to 4 until the difference between the values of T in successive iterations is smaller than the value 2

2.6 Algorithm to convert image to binary image:

Step 1: Obtain the threshold of an image using algorithm 2.5

Step 2: For each pixel in the image

* If value of the pixel is less than or equal to threshold, set it’s value to zero
* If value of the pixel is greater than the threshold, set it’s value to 255

1. **Running the programs**

\*\*\*OpenCV version 3.3 was used for the execution of this assignment\*\*\*

\*\*\*Solution1.1 folder contains the program solutions for question (1) of the Programming Design section\*\*\*

Step 1: Unzip the uploaded files to a location

Step 2: Open the Command Prompt in Windows

Step 3: Execute the Solution1.1 executable file using the command prompt

Step 4: Enter the full path of an image file when the program displays a prompt

\*\*\*Solution1.2 folder contains the program solutions for question (2) of the Programming Design section\*\*\*

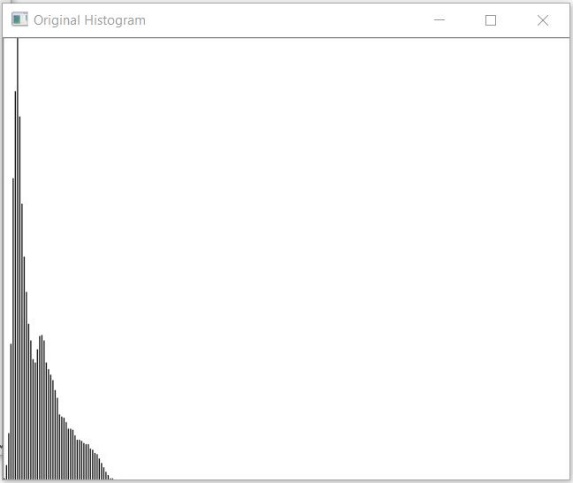
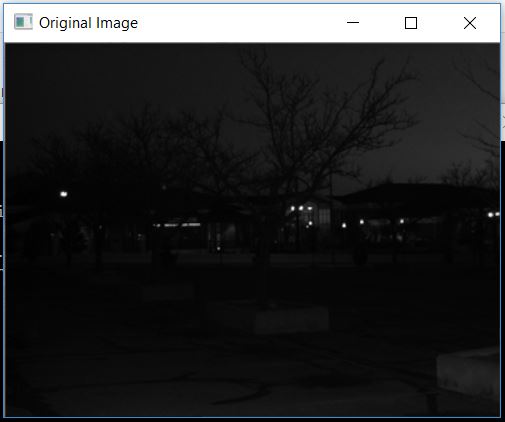
Step 1: Unzip the uploaded files to a location

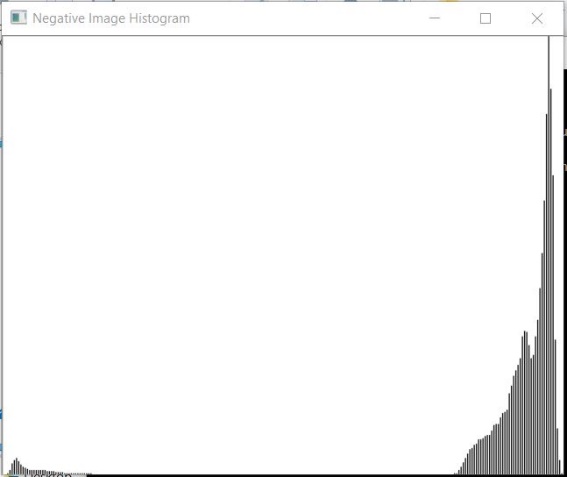
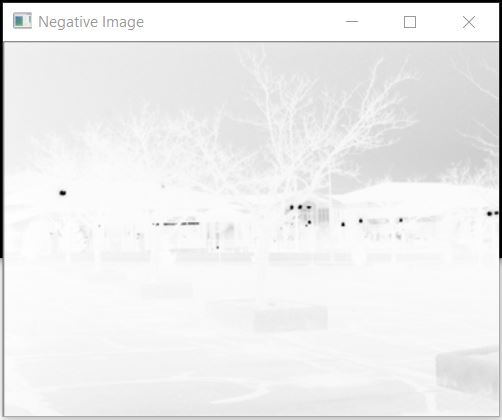
Step 2: Open the Command Prompt in Windows

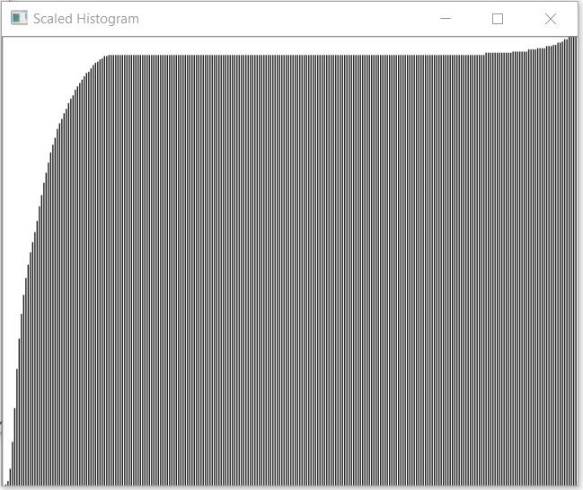
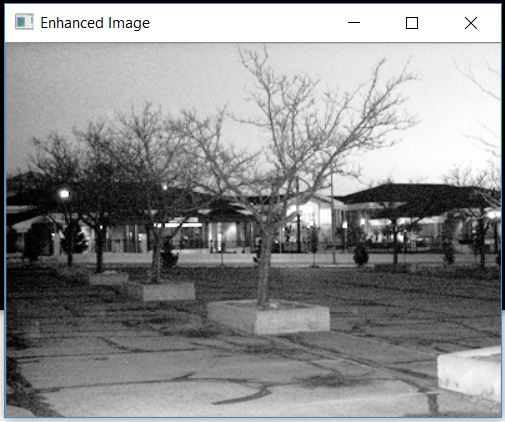
Step 3: Execute the Solution1.1 executable file using the command prompt and provide an image file path as the command line argument

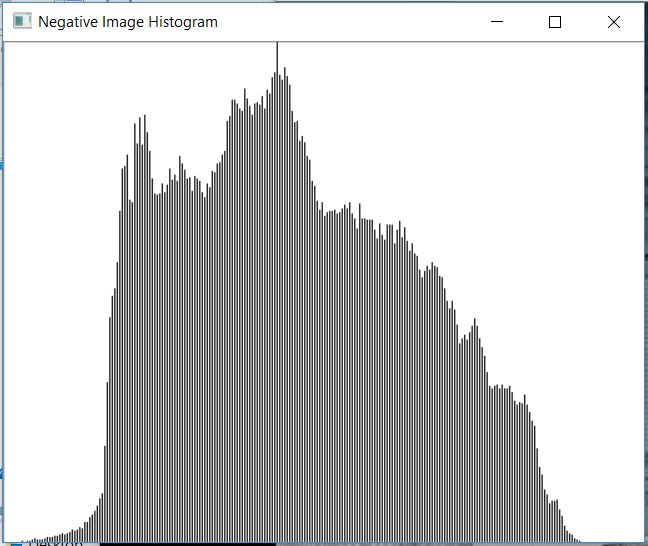
1. **Results**

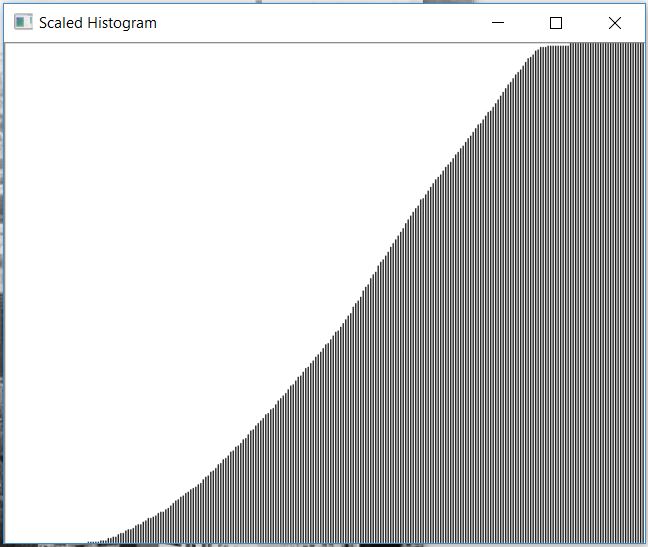
Screenshots for question (1) of Programming Design section:



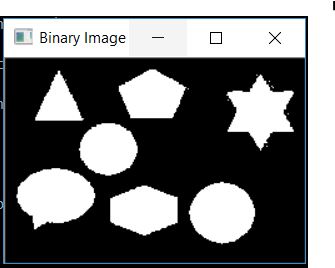


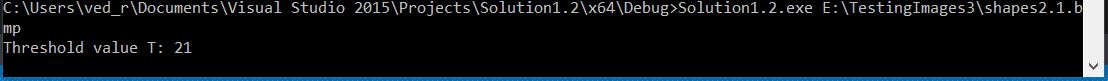


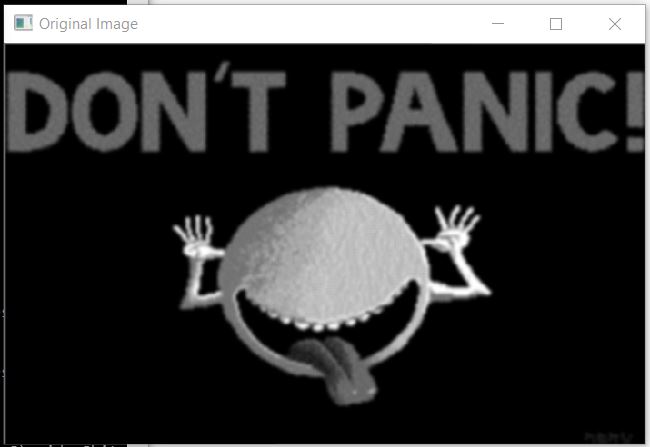


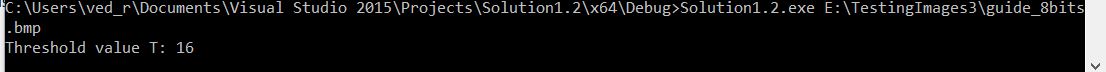


Screenshots for question (2) of Programming Design section:









1. **Bug report**

Bug Number 1: Before exiting, the Solution1.2.exe program will sometimes raise a Debug Error warning. It can be ignored and does not affect running of the program

Parts that are not complete:

Applying the 4-connected component algorithm to detect regions in an image and label the different regions