**Histogram Equalization**:

**Objective**:

Histogram equalization is used to adjust the contrast of an image using its histogram. The main objective of Histogram Equalization is to make an image visually pleasant. It is mainly used for images with both bright (overexposed) or both dark (underexposed) foreground and background. When Histogram Equalization is applied to such images, their contrast is enhanced. We have implemented Histogram Equalization for both grayscale and colored images.

**Algorithm and Design of Module**:

Grayscale image has intensity values in the range 0 to 255. Its Histogram shows distribution of image intensities over range 0 to 255. For a overexposed image, color intensities are more concentrated towards higher end of histogram and for underexposed dark image they are concentrated towards lower end. Histogram Equalization stretches the concentrated intensities over entire range of image intensity values which enhances the image contrast. Histogram Equalization flattens the histogram of original image. Cumulative probabilities of image intensities are used in this process.

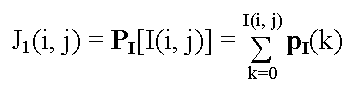
If an image is colored, it has three channel, Red, Green and Blue. Each channel has intensity range from 0 to 255. In this case we cannot separately apply Histogram Equalization to these channel as it may disturb the relative distribution of channels. To get the intensity values from a colored image without disturbing their color balance we we have converted it to YCbCr color space first. YCbCr works best for digital images.

Y is equivalent to intensities of image. Histogram Equalization is performed on intensity channel Y. After that YCbCr image is converted back to RGB format. An algorithm for Histogram Equalization is as follows:

**Algorithm**:

Let I be an input grayscale image which is a 2\*2 matrix with data type uint8. It is a Y channel in case of color image.

1. Compute histogram of an input image I
2. Normalize the histogram which stores probabilities of intensities.
3. Compute cumulative probability histogram P of a normalised histogram.
4. Compute intermediate histogram image J1 :

.

0 ≤ J1(i, j) ≤ 1

Where, I(i, j) is pixel intensity at location (i, j) in an original image matrix I. P[I(i,j)] is a cumulative probability for intensity value I(i, j).

1. Then scale J1 to cover the range 0, ..., K-1, produce the histogram-flattened/equalized image J. K is 256 highest intensity value for grayscale image or Y channel.

J(i, j) = INT [ (K-1)·J1(i, j) + 0.5 ],

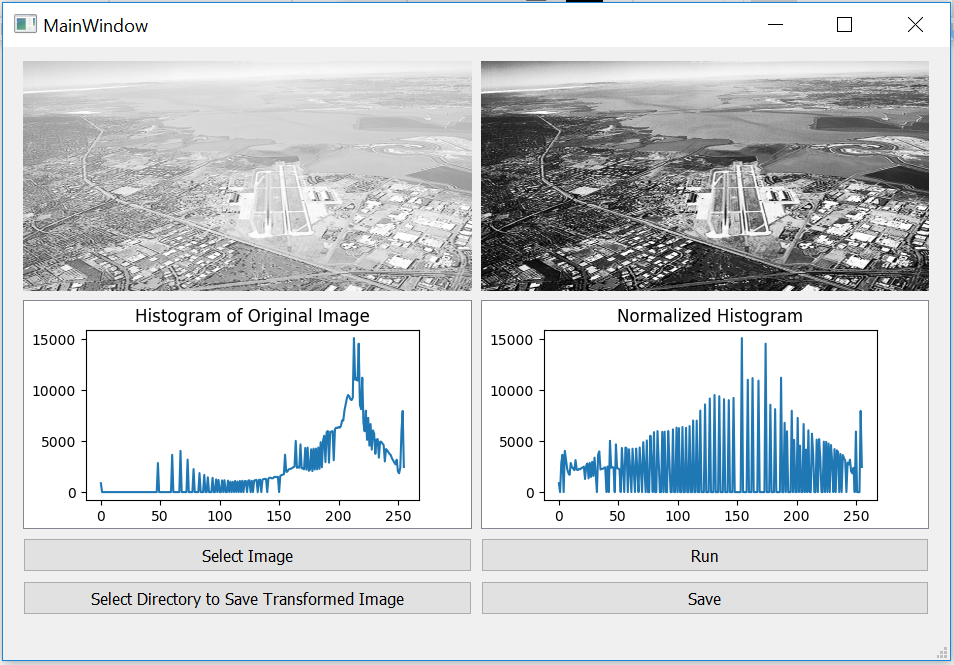
1. Return Histogram Equalized image J

For a colored image, combine equalized channel Y and Cb, Cr of original image. Convert YCbCr format back to RGB format which is a histogram equalized colored image.

**Results**:

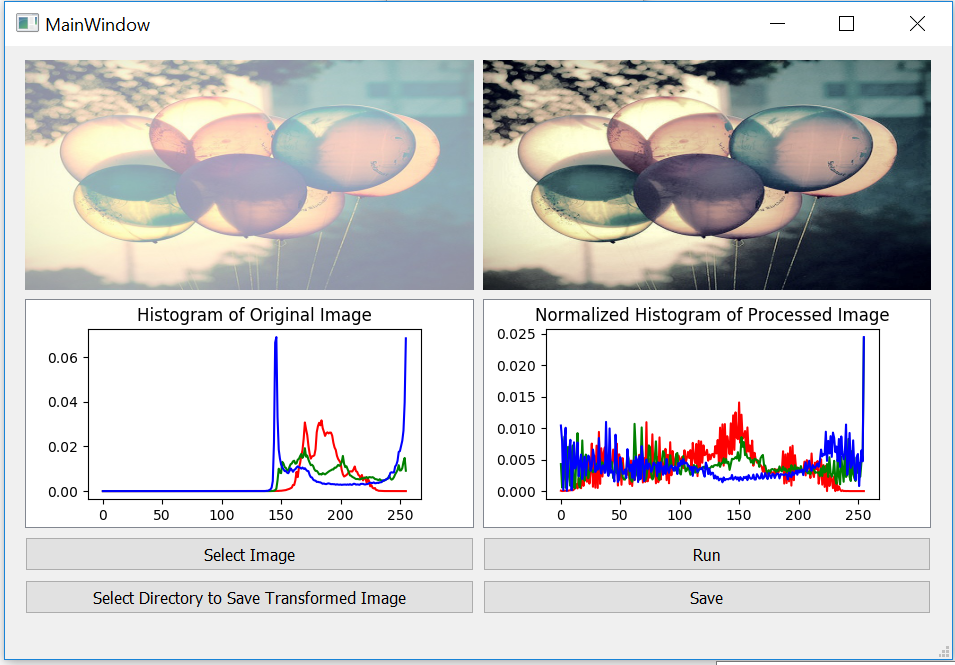
**Grayscale image**: Select ‘Histogram Equalization’ option from main UI window.

Below screenshot shows a sample output of Histogram Equalization of a grayscale image. First image is an input image which is a washed out aerial view. Histogram below it is its histogram. As the image has more bright intensities, it’s histogram shows more intensities concentrated towards higher (brighter) end. Second image is a output image after performing Histogram Equalization. It has a enhanced contrast. Its histogram below it, is more spread out.



**Color image**: Select option ‘Histogram Equalization Color’ on main UI window

Input image of balloons appears washed out. Below input image there is its histogram for three channel - Red, Green and Blue. Most of the intensities of of three channels look concentrated towards brighter intensities values in histogram. After performing Histogram Equalization on it, the resultant image (at top right) looks better with enhanced contrast. Second histogram is output image histogram with spread out intensities and it looks flattened after equalization. Y Scale of output histogram is smaller than that of input image histogram.



References:

<https://en.wikipedia.org/wiki/Histogram_equalization>

<https://prateekvjoshi.com/2013/11/22/histogram-equalization-of-rgb-images/>