EXPERIMENT - 2

Report Submission

<u>Topic</u> : Perform Histogram Matching and Equalization for gray & color images

GROUP - 18

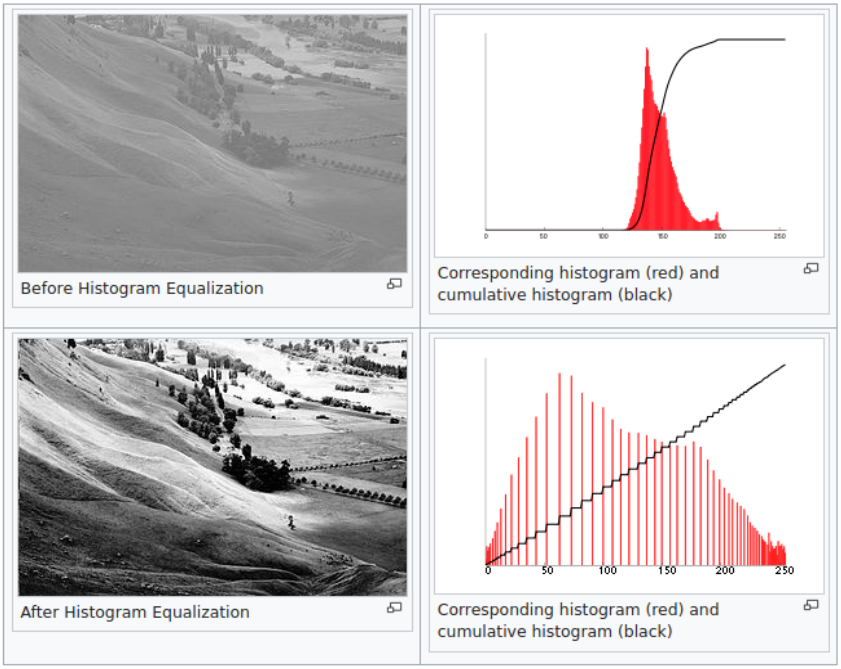
Name	Sumegh Roychowdhury	Arun Sammit Pandey
Roll Number	17EC35033	17EC35006

Table of Contents

Histogram Equalization :

Histogram Equalization is a method in image processing of contrast adjustment using the image's histogram. This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

$p_x(i) = n_i / n$; where n is total pixels
$cdf_x(i) = \Sigma p_x(i)$
$H_i = (L-1)\ cdf_x(i)$; where L is the max intensity



Histogram Matching :

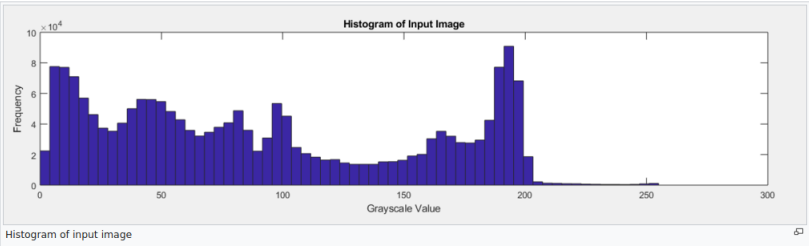
In image processing, **histogram matching** or **histogram specification** is the transformation of an image so that it's histogram matches a specified histogram. The well-known **histogram equalization** method is a special case in which the specified histogram is uniformly distributed.

Suppose the image has pdf $p_r(r_j) = n_j / n$ and desired pdf is $p_z(r_z)$. Now the cdf's will be:

$S(r_k) = \Sigma p_r(r_j)$

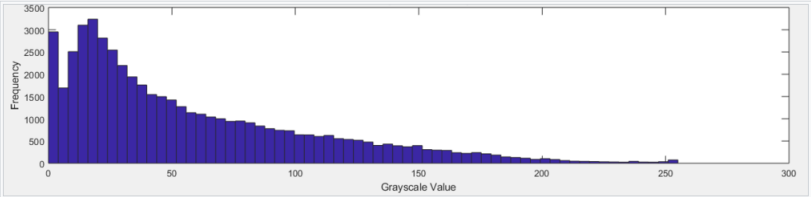
$G(z_k) = \Sigma p_z(r_z)$

The idea is to map each r value in X to the z value that has the same probability in the desired pdf i.e. $z = G^{-1}(S(r))$



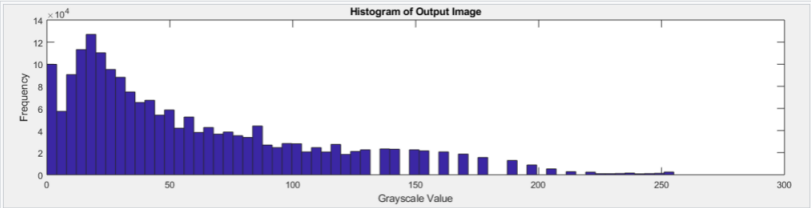
Histogram of input image

It will be matched to this reference histogram to emphasize the lower gray levels.



Desired reference histogram

After matching, the output image has the following histogram



Histogram of output image after matching

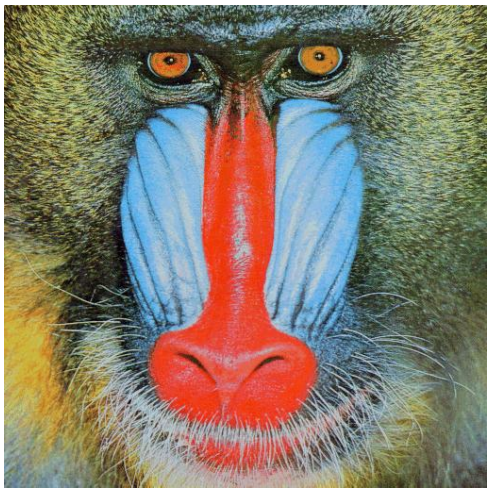
Algorithm :

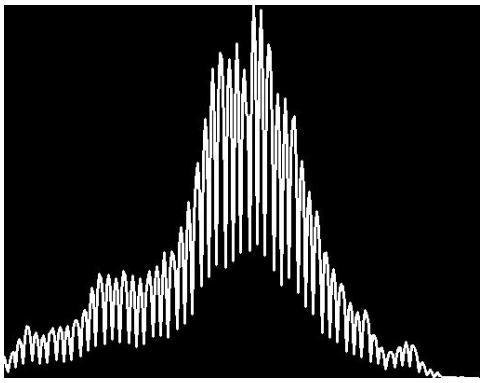
- First we define the **transformation** function for the Histogram Equalization named as - `EqualizeTransform::EqualizeTransform(Mat input, int maxIntensityVal)` which takes input image and max intensity as input and returns the histogram equalized array.
- To do the above, we iterate through all pixels & channels and update *transformation[intensity]++* to keep track.
- Next we find the *cumulative sum* to find the CDF and then divide by *max_intensity* to get the distribution. Finally we apply the formula $(L-1)*cdf_x(i)$ to get the equalized values.
- For **matching** we define the function `int EqualizeTransform::invTransform(int val)`

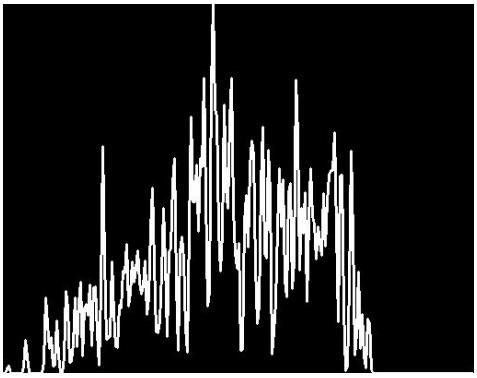
- The above function uses binary search to find out index *i* such that G(val) = H(i). This is the matching function **applied** to all pixels using the function `Mat applyToEach(Mat input,function<int(int)> f)`

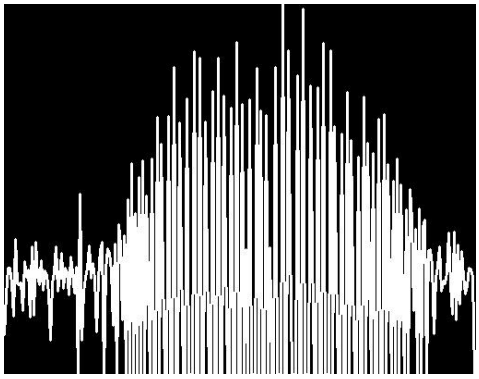
Results :

Original and Target Images :

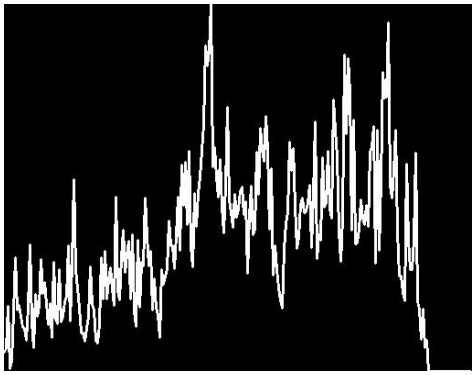
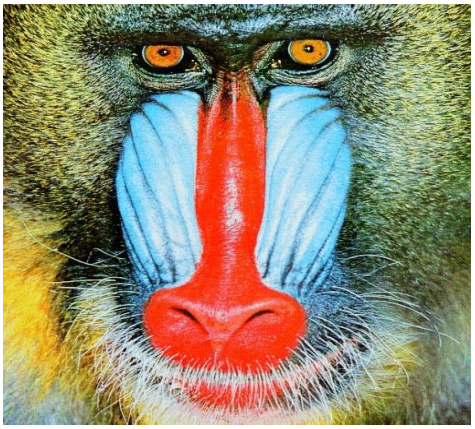

(original)

(target)


(original histogram)

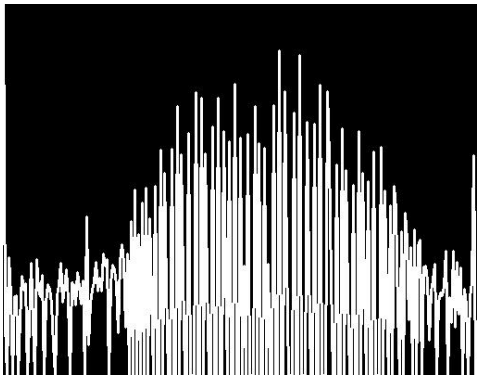

(target histogram)




(original equalized)

(target equalized)




(matched histogram & image)

Finally, we can see that near the centre of the target image the contrast has been enhanced since the original target image had a similar contrast profile. This verifies our understanding.

Conclusion :

In this experiment we saw how to perform histogram equalization to **improve contrast** and also do histogram matching using the same logic but having an inverse transform. The entire code is written in C++ modular format and uses opencv-c++ library.

THANK YOU
