Batch: B1 Roll No.:16010121045

Experiment No. 7

Title: Write a program to apply the global processing technique: Histogram equalization on a digital image

Objective: To learn and understand the concept of histogram stretching and equalization in image enhancement operations.

Expected Outcome of Experiment:

CO	Outcome
CO4	Design & implement algorithms for digital image enhancement, segmentation & restoration.

Books/ Journals/ Websites referred:

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html.
3. R. C.Gonsales R.E.Woods, “Digital Image Processing”, Second edition, Pearson Education
4. S.Jayaraman, S Esakkirajan, T Veerakumar “Digital Image Processing “Mc Graw Hill.
5. S.Sridhar,”Digital Image processing”, oxford university press, 1st edition."

Pre Lab/ Prior Concepts:

Image histogram:

In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bit greyscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those greyscale values. Histograms can also be taken of color images either individual histogram of red, green and blue channels can be taken, or a 3-D histogram can be produced, with the three axes representing the red, blue and green channels, and brightness at each point representing the pixel count. The exact output from the operation depends upon the implementation it may simply be a picture of the required histogram in a suitable image format, or it may be a data file of some sort representing the histogram statistics.

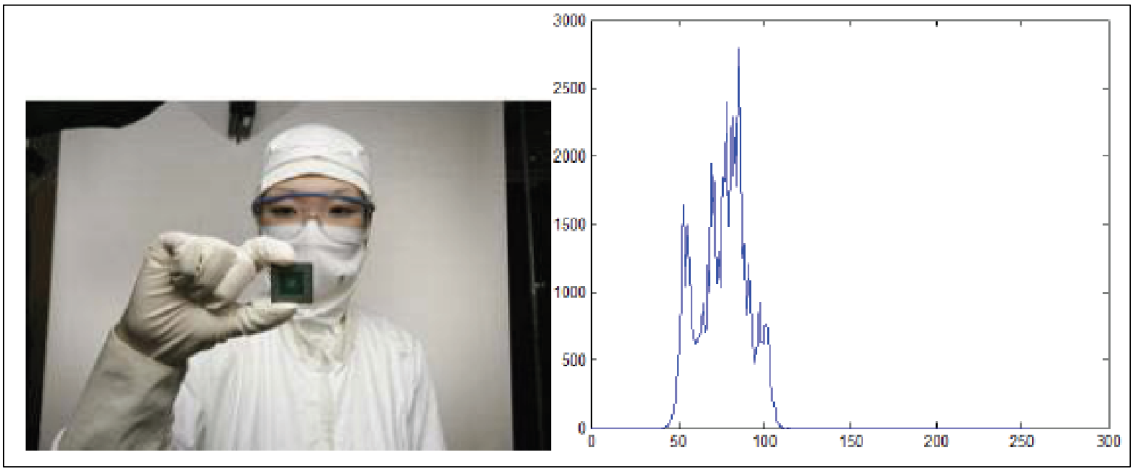


Fig. 1 An image and its histogram

Histogram Equalization:

A perfect image is one which has equal number of pixels in all its grey levels. hence our objective is not only to spread the dynamic range , but also to have equal pixels in all the grey levels. This technique is known as histogram equalization.

Basically the histogram equalization spreads out intensity values along the total range of values in order to achieve higher contrast. This method is especially useful when an image is

represented by close contrast values, such as images in which both the background and foreground are bright at the same time, or else both are dark at the same time. For example, the result of applying histogram equalization to the image in figure 1 is presented in figure 2.

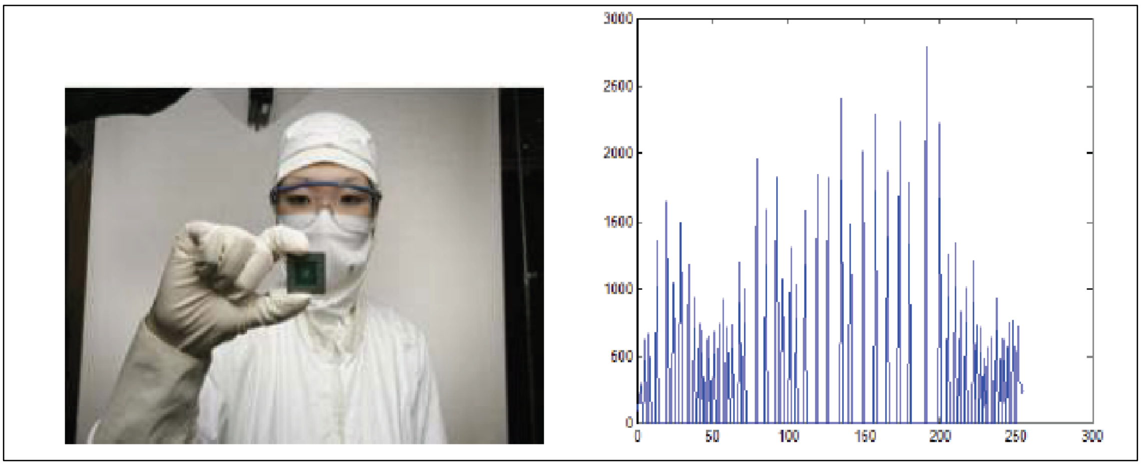


Fig. 2 New image and its equalized histogram

Description of cumulative histogram equalization:

Here are the steps for implementing this algorithm.

1. Create the histogram for the image.
2. Calculate the cumulative distribution function histogram.
3. Calculate the new values through the general histogram equalization formula.
4. Assign new values for each gray value in the image.

Thus processed image is obtained by mapping each pixel with level rk into a corresponding pixel with level sk in o/p image. This transformation is called Histogram equalization

Resources Used: Matlab

Implementation Details:

Code:

```
I = imread('babyincradle.png');
if ndims(I) == 3
  original_img = rgb2gray(I);
else
  original_img = I;
end


[rows, cols] = size(original_img);

histogram = zeros(1, 256);
pdf = zeros(1, 256);


for i = 1:rows
  for j = 1:cols
    intensity = original_img(i, j);
    histogram(intensity + 1) = histogram(intensity + 1) + 1;
  end
end
pdf = histogram / (rows * cols);

cdf = zeros(1, 256);


sum = 0;
for i = 1:256
  sum = sum + pdf(i);
  cdf(i) = round(sum * 255);  % Scale CDF to intensity range (0-255)
end

equalized_img = uint8(zeros(rows, cols));
for i = 1:rows
  for j = 1:cols
    intensity = original_img(i, j);
    equalized_img(i, j) = cdf(intensity + 1);
  end
end


equalized_histogram = zeros(1, 256);
for i = 1:rows
  for j = 1:cols
    intensity = equalized_img(i, j);
    equalized_histogram(intensity + 1) = equalized_histogram(intensity + 1)
+ 1;
  end
end

figure(1);
```

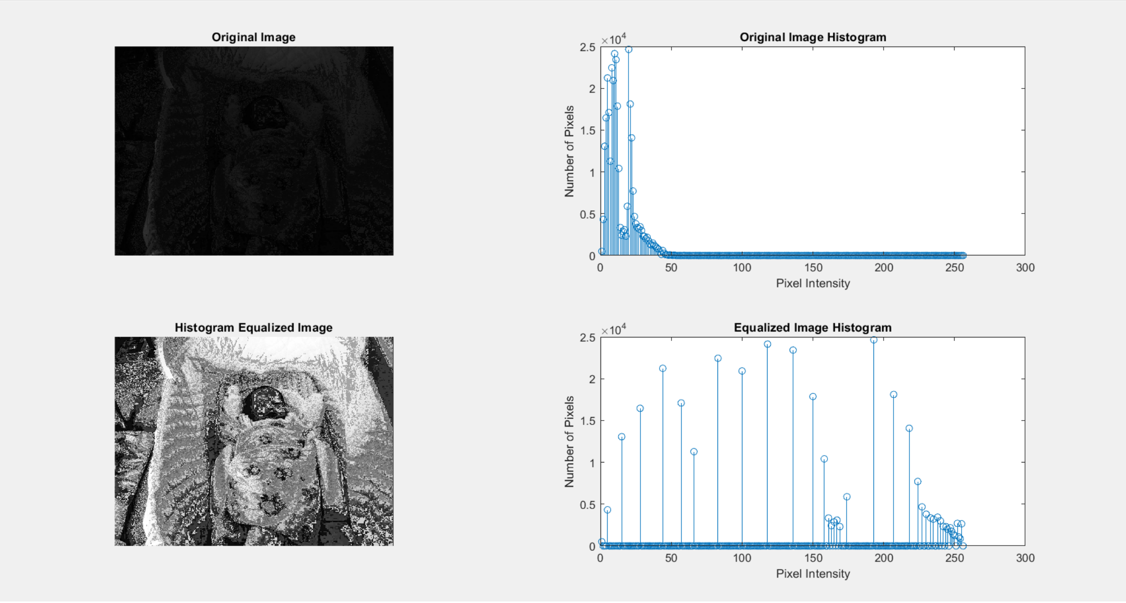
```
subplot(221);
imshow(original_img);
title('Original Image');

subplot(222);
stem(1:256, histogram);
xlabel('Pixel Intensity');
ylabel('Number of Pixels');
title('Original Image Histogram');


subplot(223);
imshow(equalized_img);
title('Histogram Equalized Image');

subplot(224);
stem(1:256, equalized_histogram);
xlabel('Pixel Intensity');
ylabel('Number of Pixels');
title('Equalized Image Histogram');
```

Output:



Conclusion:- Histogram Equalisation was implemented successfully.

Post Lab Descriptive Questions

1.Compare between contrast stretching and histogram equalization.

Contrast Stretching:
Simpler method: Involves linearly stretching the existing range of pixel intensities in the image. Function: Minimum pixel value is mapped to 0 and the maximum to the maximum possible value (often 255 for 8-bit images). Intermediate values are scaled proportionally. Impact on histogram: Overall shape remains similar, just stretched across a wider range. Result: Enhances contrast in images with a narrow intensity range, but might introduce unwanted noise amplification in areas with close intensity values. Reversibility: The original image can be recovered from the stretched version.

Histogram Equalization:
More advanced technique: Analyzes the image's histogram (distribution of pixel intensities). Function: Aims for a uniform distribution by redistributing pixel values.
Impact on histogram: Transforms the histogram into a flat shape, where each intensity level has (ideally) an equal number of pixels. Result: More effective in improving contrast, especially in images with uneven lighting or limited dynamic range. Can reveal hidden details but might create artificial details in areas with low variations. Irreversibility: Due to the non-linear transformation, the original image cannot be retrieved directly.