ELL715
Assignment 2

SAKSHAM JAIN 2017MT10747
SHIVAM SINGLA 2016MT10619

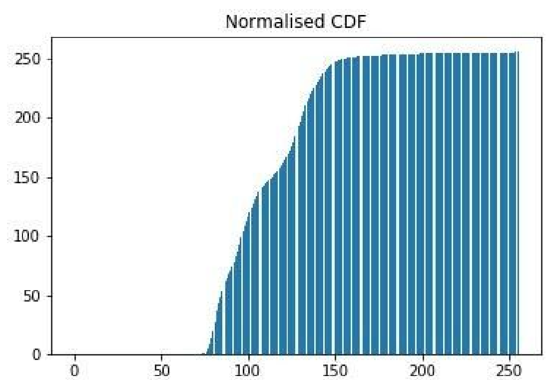
Part 1

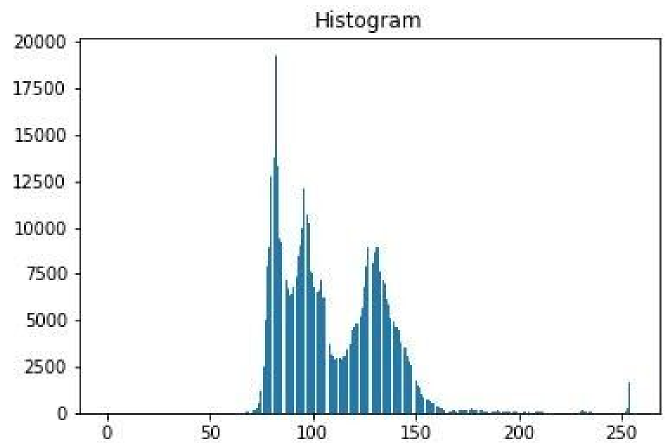
Image Histograms

Write a program (in MATLAB or PYTHON) to compute the histogram of an intensity RGB image

For Image 1:

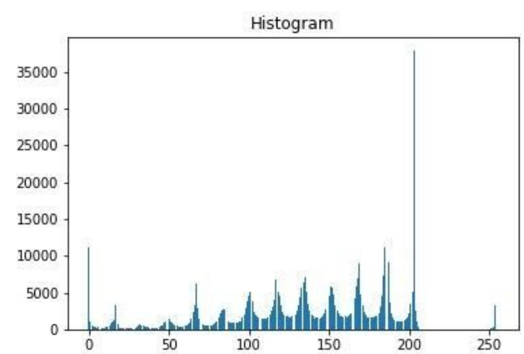
```
Mean: 109.89078472520309
Standard Deviatoin: 25.766057422152127
Energy: 0.015215776588520758
Entropy: 6.298141444588383
Third Moment: 21278.47077632999
Fourth Moment: 2997004.1085854233
```

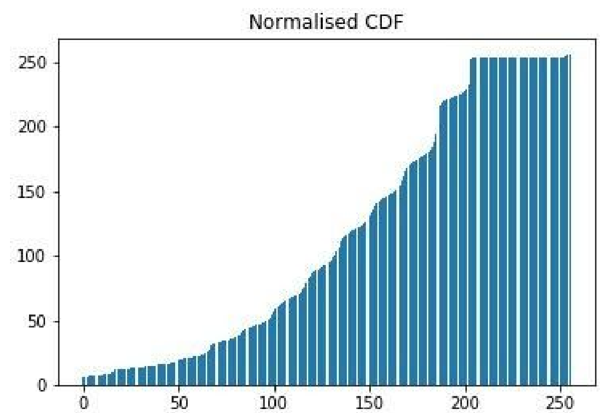




For Image 2:


```
Mean: 138.99586504893264
Standard Deviatoin: 53.652124761665505
Energy: 0.017754054263396532
Entropy: 6.840301898568963
Third Moment: -114470.59060130236
Fourth Moment: 24963055.09887269
```





Formulas Used:

MEAN : $$\frac{\sum f(x, y)}{MN}$$

VARIANCE : $$\frac{\sum_{(x,y)\in I} \left(f(x, y) - \mu\right)^2}{MN} \quad \{\mu = \text{Mean}\}$$

STANDARD DEVIATION : $$\sqrt{\text{Variance}}$$

ENERGY : $$\sum_k \left(p(r_k)\right)^2 \quad \text{when } r_k = \frac{f_k}{MN}$$

ENTROPY : $$\sum_k \left(-p(r_k) \cdot \log_2\left(p(r_k)\right)\right)$$

THIRD MOMENT : $$\frac{\sum_{(x,y)\in I} \left(f(x, y) - \mu\right)^3}{MN}$$

FOURTH MOMENT : $$\frac{\sum_{(x,y)\in I} \left(f(x, y) - \mu\right)^4}{MN}$$

Part 2

Write a program (in MATLAB or PYTHON) to improve the image contrast of a grayscale image using various mentioned methods. Don't use inbuilt functions.

- **We first converted the RGB image to grayscale image using matplotlib image library.**

<u>**The two images used for experiments**</u>

<u>Contrast Stretching :-</u>

Contrast stretching (often called normalization) is a simple image enhancement technique that attempts to improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values, *e.g.* the the full range of pixel values that the image type concerned allows.

The simplest sort of normalization scans the image to find the lowest and highest pixel values currently present in the image. Call these *c* and *d*. We take a as 0 and b as 255. Then each pixel *P* is scaled using the following function:

$$P_{out} = (P_{in} - c)\left(\frac{b-a}{d-c}\right) + a$$

We use this function to get the following images as results :-

Original Image vs Improved Image

Original Image vs Improved Image

As we can see already the features are much more clear in the improved images.

Gamma Correction :-

Gamma correction controls the overall brightness of an image. Images which are not corrected can look either bleached out or too dark. Suppose a computer monitor has 2.2 power function as an intensity to voltage response curve. This just means that if you send a message to the monitor that a certain pixel should have intensity equal to x, it will actually display a pixel which has intensity equal to x2.2 Because the range of voltages sent to the monitor is between 0 and 1, this means that the intensity value displayed will be less than what you wanted it to be. Such a monitor is said to have a gamma of 2.2.

We use the following formula with gamma = 2.2, but we can use different values as well.

```
encoded = ((original / 255) ^ (1 / gamma)) * 255
```

Original Image vs Improved Image

Original Image vs Improved Image

As we see that the first image has improved a lot, but it kind of blows off the second one. Normally using gamma as 2.2 is a standard procedure, but the best part of this

technique is that we can adjust gamma to get better results as per the image. For eg. we use gamma 1.2 in the following picture.

This improves the image without completely blowing off the pixel values.

<u>Histogram Equalization :-</u>

As discussed in class, this is a really standard technique of normalising and increasing contrast for both RGB and grayscale images.

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$$

We use the following formula from slides, where L = 256, MxN is the size of the image.

Original Image vs Improved Image

Original Image vs Improved Image

This is one of the best techniques for improving the contrast, and these results just show it why.

Unsharp Masking :-

Sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image
e.g., printing and publishing industry

Steps :-

- **Blur the original image**
- **Subtract the blurred image from the original**
- **Add the mask to the original**

We use a gaussian filter here but other filters can also be used.

Original Image vs Improved Image

Original Image vs Improved Image

It's a relatively smoother technique than all other techniques. But we can adjust that with the parameters of the gaussian distribution used. Here we have used standard gaussian distribution.

References:

https://stackoverflow.com/questions/12201577/how-can-i-convert-an-rgb-image-into-grayscale-in-python

https://stats.stackexchange.com/questions/235270/entropy-of-an-image

https://stackoverflow.com/questions/4562801/what-is-energy-in-image-processing

https://www.tutorialspoint.com/dip/histograms_introduction.htm