EL-GY 6123 Image and Video Processing
Matlab Assignment 1

Name: Amitesh Kumar Sah
NYU ID: N19714360

Question 4

Write a matlab program that can

i) read your color image into a matrix (you can use imread() function in MATLAB);

ii) covert it to a 8-bit grayscale image using the RGB to Y conversion formula in the RGB to YCbCr formula provided (You should not use the MATLAB built-in function rgb2ycbcr),

iii) generate a digital negativeversion of your grayscale image, and

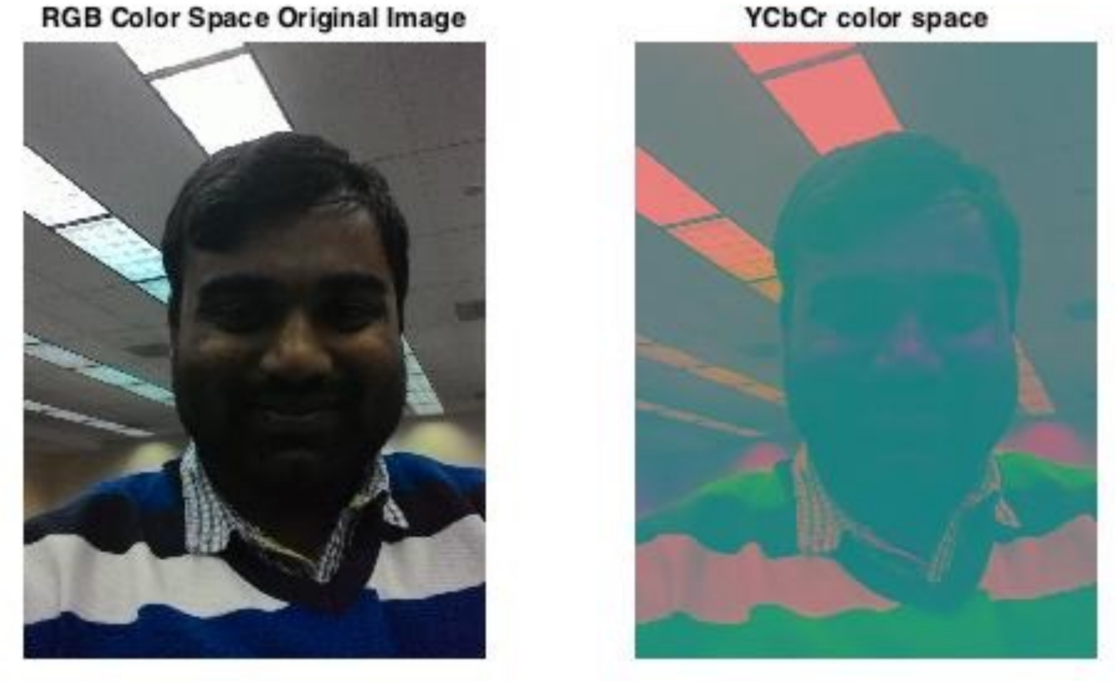
iv) display both the color image, the original gray scale and the negative image. REMEMBER THE CAVEATS!

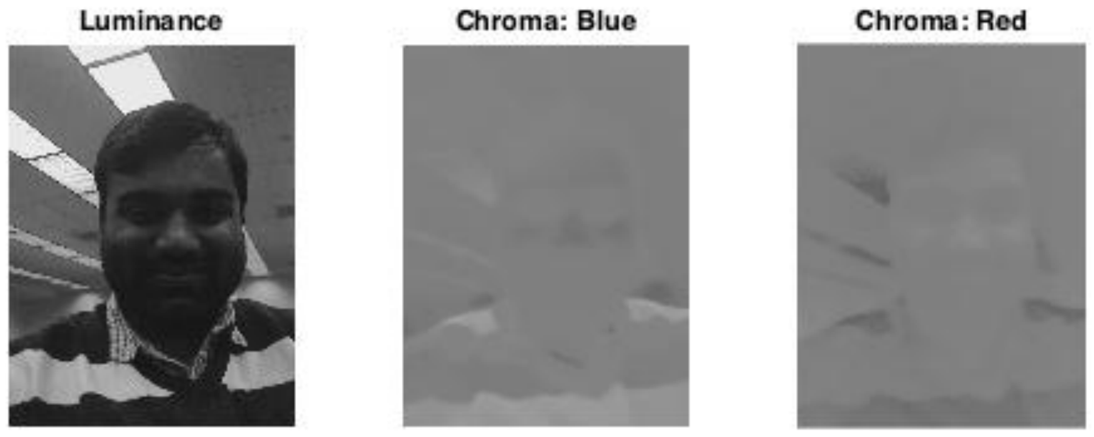
Solution

Matlab Program

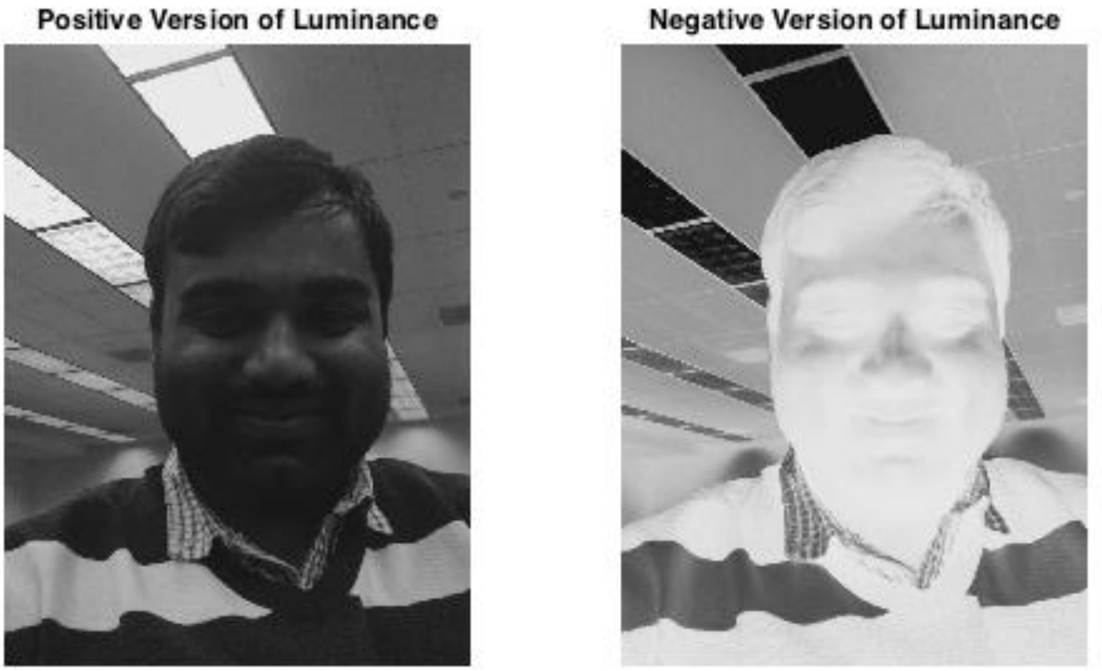
```
clc;clear all;close all;

% Reading a image into a matrix from a file
IRGB_unsigned_raw=imread('amitesh.jpg');
IRGB_unsigned=imrotate(IRGB_unsigned_raw,90);

IRGB_signed=im2double(IRGB_unsigned);

% Conversion of RGB to 8-bit grayscale image
T=[0.257 0.504 0.098;-0.148 -0.291 0.439; 0.439 -0.368 -0.071]; % Conversion matrix from RGB to YCbCr

R=IRGB_signed(:,:,1);
G=IRGB_signed(:,:,2);
B=IRGB_signed(:,:,3);
[Row,Col]=size(R);
for i=1:Row
    for j=1:Col
      Y(i,j)= (0.257*R(i,j))+(0.504*G(i,j))+(0.098*B(i,j))+(16/255);
      Cb(i,j)=(-0.148*R(i,j))+(-0.291*G(i,j))+(0.439*B(i,j))+(128/255);
      Cr(i,j)=(0.439*R(i,j))+(-0.368*G(i,j))+(-0.071*B(i,j))+(128/255);
    end
end
  I_YCbCr= im2uint8(cat(3,Y,Cb,Cr));
  figure,subplot(1,2,1),imshow(IRGB_unsigned);
    title('RGB Color Space Original Image');
  subplot(1,2,2), imshow(I_YCbCr);%8 bit greyscale image Y
    title('YCbCr color space');
  figure,
  Y1=I_YCbCr(:,:,1);
  Cb1=I_YCbCr(:,:,2);
  Cr1=I_YCbCr(:,:,3);
  subplot(1,3,1);
  imshow(Y1);
  title('Luminance');
  subplot(1,3,2);
  imshow(Cb1);
  title('Chroma: Blue');
  subplot(1,3,3);
  imshow(Cr1);
  title('Chroma: Red');

% Generating a digital negative version of your greyscale image of
% Y(Luminance)

Neg_Y=255-Y1(:,:);
figure,subplot(1,2,1),imshow(Y1);
title('Positive Version of Luminance');
subplot(1,2,2),imshow(Neg_Y);
title('Negative Version of Luminance');
```

Result:





Digital negative version of my grayscale image



Question 5:

Write a Matlab function that can compute the histogram of a grayscale image (assuming 256 levels of gray). Try the three possible ways described in slide 7, and see which one is faster. Finalize your program with the fastest method. In a separate main program, apply the program to a test image, and display the histogram as a stem plot besides the image (using "subplot" function). You are not allowed to simply use the "imhist" or "hist" function in Matlab, although you are encouraged to compare your results with those obtained using these functions.

Solution
Matlab Program

```
% Implementing the fastest histogram calculation technique

clc;clear all; close all;
img=imread('baboon.png');
if ndims(img)>2       % To check if it is rgb or gray image
   img=rgb2gray(img);
end

[N M]=size(img);
h=zeros(256,1);
  for I=0:255
      h(I+1)=sum(sum(img==I));
  end
  figure,subplot(1,2,1), imshow(img);
  title('Original Gray level Image');
  subplot(1,2,2),stem(h);
  title('Histogram of the original Image');
```

Function of histogram created

Method 1

```
function h = histogram_m1(imagename)
% This function calculates the histogram of a gayscale image using the 1st
% method. Here, each pixel value is checked 256 times

img=imread(imagename);
figure, imshow(img);
tic
[N M]=size(img);
h=zeros(256,1);
for I=0:255
   for i=1:N
      for j=1:M
         if img(i,j)==I
            h(I+1)=h(I+1)+1;
         end
      end
   end
end
figure,bar(h);
toc
```

Method 2

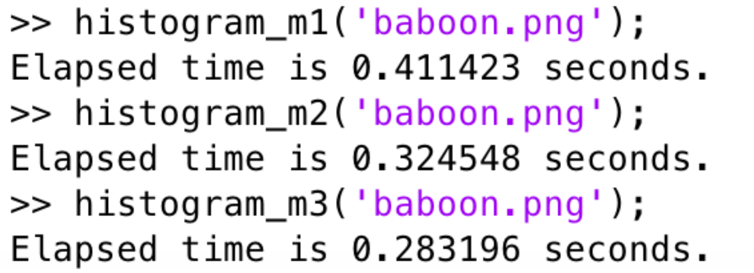
```
function h = histogram_m2(imagename)
% This function calculates the histogram of a gayscale image using the 2nd method
% Here, each pixel value is checked only once and an increment is done at
% the location where location value is the pixel value

img=imread(imagename);
figure, imshow(img);
tic
[N M]=size(img);
h=zeros(256,1);
   for i=1:N
      for j=1:M
         f= img(i,j);
            h(f+1)=h(f+1)+1;
      end
   end
figure,bar(h);
toc
```

Method 3

```
function h = histogram_m3(imagename)
% This function calculates the histogram of a gayscale image using the 3rd method
% Here, particular gray value is checked in whole image and then added all
% at once, 1st row is checked and summed, then all row is checked and then
% summation.
img=imread(imagename);
figure, imshow(img);
tic
[N M]=size(img);
```

```
h=zeros(256,1);
  for I=0:255
      h(I+1)=sum(sum(img==I));
  end
figure,bar(h);
toc
```

Result:



```
>> histogram_m1('baboon.png');
Elapsed time is 0.411423 seconds.
>> histogram_m2('baboon.png');
Elapsed time is 0.324548 seconds.
>> histogram_m3('baboon.png');
Elapsed time is 0.283196 seconds.
```

Hence, it shows that method 3 is the fastest among all the three methods.

Question 6

Write a Matlab program that performs histogram equalization on a grayscale image. Your program should: i) compute the histogram of the input image by calling your own histogram function from previous problem; ii)compute the histogram equalizing transformation function; iii) apply the function to the input image; iv) compute the histogram of the equalized image;v) display (and print) the original and equalized images, as well as their corresponding histograms, all in one figure. You are not allowed to simply use the “histeq” function in Matlab, although you are encouraged to compare your results with those obtained using these functions.

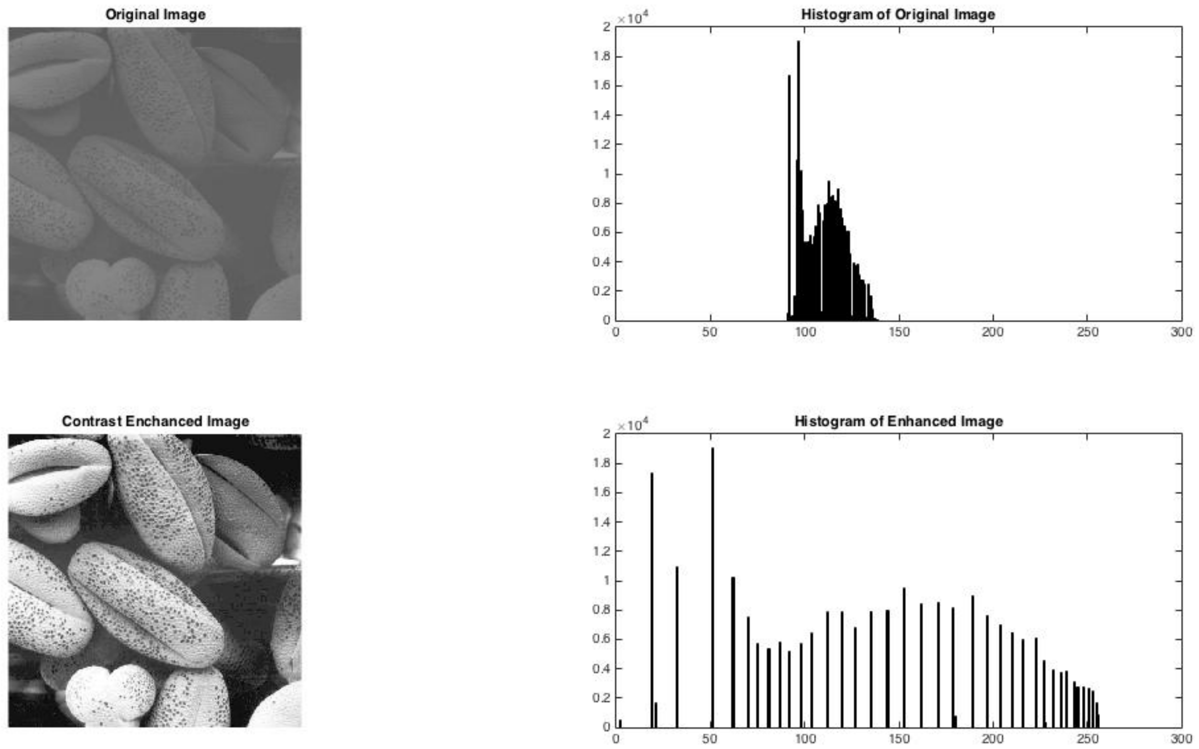
Solution

Matlab Program

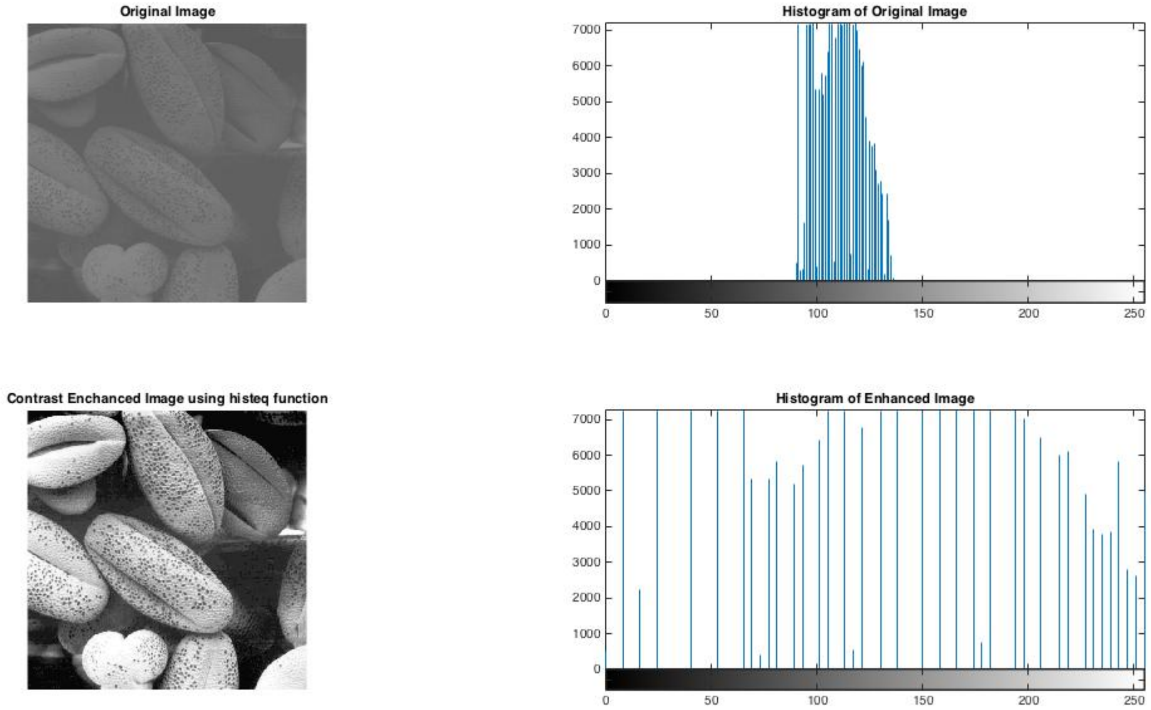
```
clc;clear all; close all;

img=imread('7.png');
if ndims(img)>2        % To check if it is rgb or gray image
   img=rgb2gray(img);
end
f=histogram_m3(img);
 [r,c]=size(img);
 count=r*c;
p_f=f./count;
% Making a cumulative function
gl_bar=p_f(1,1);
for i=2:256
   gl_bar(i,1)=gl_bar(i-1,1)+p_f(i,1);
end
gl=round(gl_bar.*255);
% figure,plot(gl);

y=0:255;
p_g=zeros(256,1);
```

```
histeq_img=zeros(r,c);
% Now take the individual pixel from the image and store the mapped value
for i=1:r
  for j=1:c
    gray_level=img(i,j)+1; %Gray level of the existing pixel of the original image
    histeq_img(i,j)=gl(gray_level);
  end
end
% figure, imshow(uint8(histeq_img));
k=histogram_m3(histeq_img);

figure,
subplot(2,2,1),imshow(img);title('Original Image');
subplot(2,2,2),bar(f); title('Histogram of Original Image');
subplot(2,2,3),imshow(uint8(histeq_img));title('Contrast Enchanced Image using using transformation function');
subplot(2,2,4),bar(k); title('Histogram of Enhanced Image');
Ieq=histeq(img);
figure,
subplot(2,2,1),imshow(img);title('Original Image');
subplot(2,2,2),imhist(img); title('Histogram of Original Image');
subplot(2,2,3),imshow(uint8(Ieq));title('Contrast Enchanced Image using histeq function');
subplot(2,2,4),imhist(Ieq); title('Histogram of Enhanced Image');
```

Result:
Histogram equalization of Original Image using histogram function created using method3



Histogram equalization using inbuilt function histeq()



Here, we can conclude that histogram equalization done by histeq() is better than the histogram equalization function created by method 3 and there can be better function than method3.