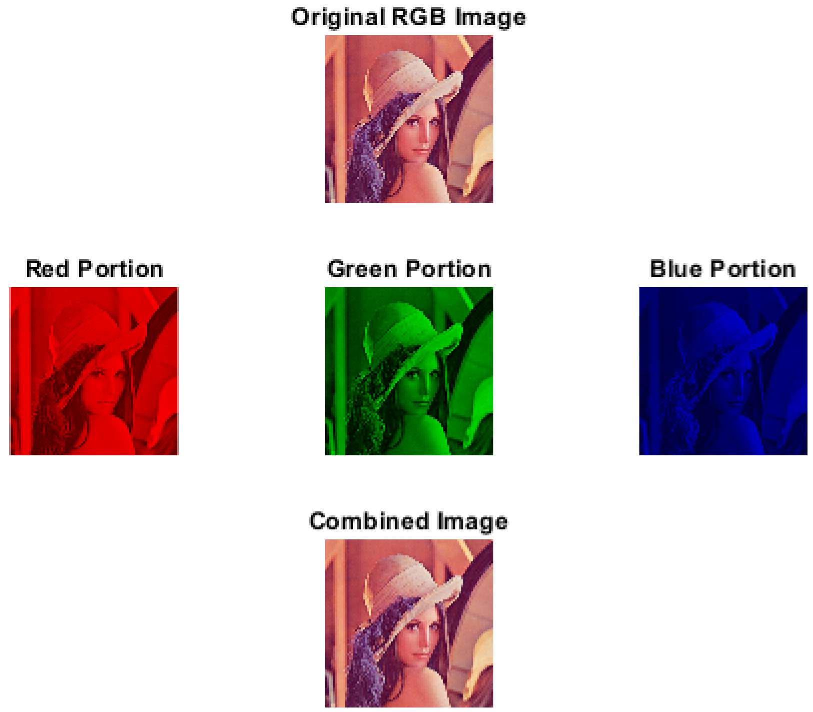
```
%NAME: Krishna Kodali
%INST: IIT Bhubaneswar
%DATE: 19/09/2020
%CATEGORY: BTech
%BRANCH: Computer Science
%Roll Number: 17CS01008

%Removing previous Buffer
clc;clear;close all;
```

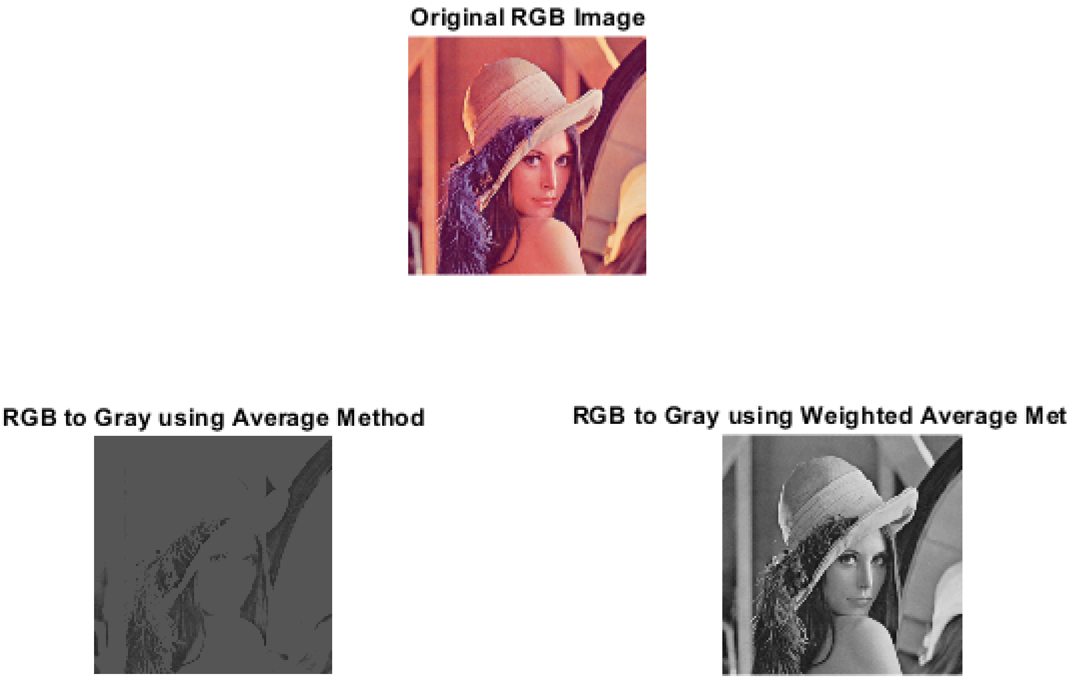
Question 1: For a given input RGB image, separate the intensity and color information using different color models, and display them separately. Then combine the components back into the original RGB image.

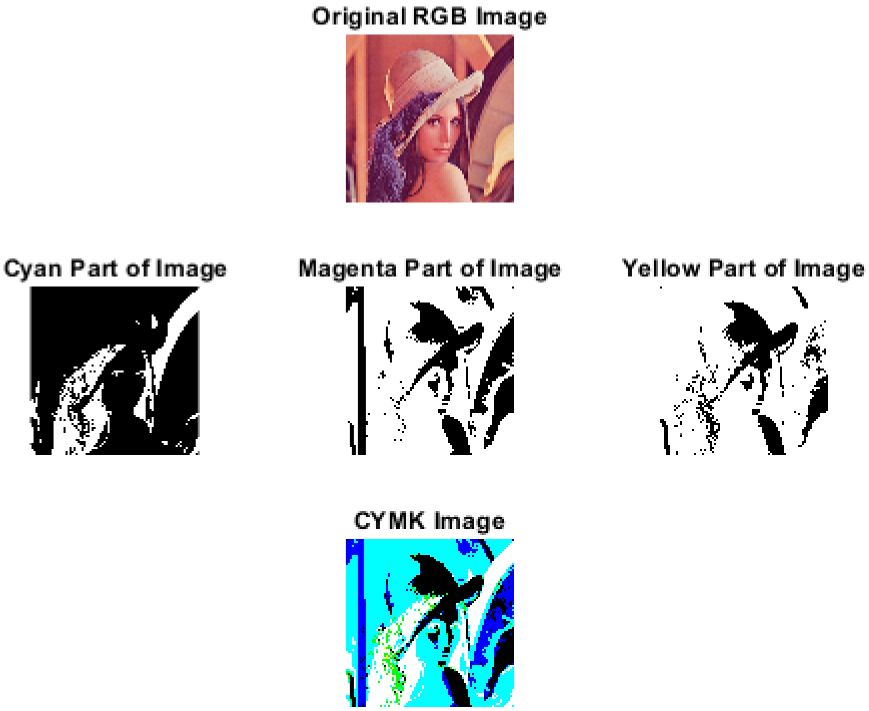
```
%Read the tif image file format
Image = Tiff('lena_color_256.tif', 'r');
rgbImage = read(Image);

%Seperating rgb channels
redC = rgbImage(:, :, 1);
greenC = rgbImage(:, :, 2);
blueC = rgbImage(:, :, 3);

%Creating a 2d matrix of zeroes
backgroud = zeros(size(rgbImage, 1), size(rgbImage, 2));

%Getting red blue and green parts of the image by concactenating them.
redImage = cat(3, redC, backgroud, backgroud);
greenImage = cat(3, backgroud, greenC, backgroud);
blueImage = cat(3, backgroud, backgroud, blueC);

%Combining the rgb Channels to form  the
Combinedrgb = cat(3, redC, greenC, blueC);

%Plotting all the images using subplot
figure
subplot(3,3,2);
imshow(rgbImage);
title('Original RGB Image');
subplot(3,3,4);
imshow(redImage);
title('Red Portion');
subplot(3,3,5);
imshow(greenImage);
title('Green Portion');
subplot(3,3,6);
imshow(blueImage);
title('Blue Portion');
subplot(3,3,8);
imshow(Combinedrgb);
title('Combined Image');

% Converting RGB to GrayScale
```

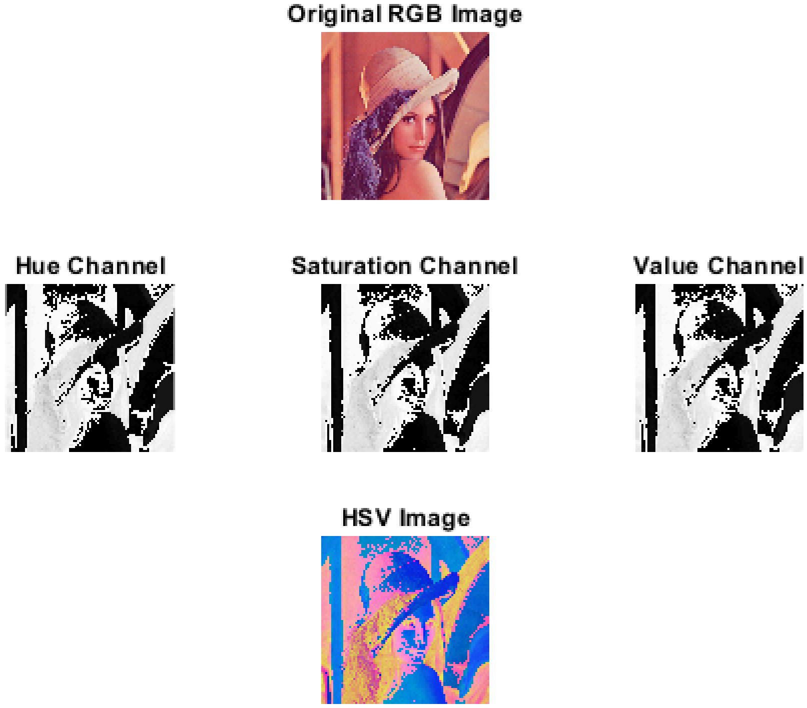
```
figure
subplot(2,3,2);
imshow(rgbImage);
title('Original RGB Image');
subplot(2,3,4);
grayImage = (redC + greenC + blueC)/3;
imshow(grayImage);
title('RGB to Gray using Average Method');
subplot(2,3,6);
grayImage = (0.21*redC) + (0.72*greenC) + (0.07*blueC);
imshow(grayImage);
title('RGB to Gray using Weighted Average Method');


% Converting RGB to CMKY Model
cmykImage = zeros(size(rgbImage));
for i = 1:size(rgbImage, 1)
    for j = 1:size(rgbImage, 2)
        r = rgbImage(i,j,1)/255;
        g = rgbImage(i,j,2)/255;
        b = rgbImage(i,j,3)/255;
        cmykImage(i,j,1) = uint8((1-r)*255);
        cmykImage(i,j,2) = uint8((1-g)*255);
        cmykImage(i,j,3) = uint8((1-b)*255);
    end
end

%Plotting different components using subplot
figure
subplot(3,3,2)
imshow(rgbImage)
title('Original RGB Image');
subplot(3,3,4)
imshow(cmykImage(:, :, 1));
title('Cyan Part of Image');
subplot(3,3,5)
imshow(cmykImage(:, :, 2));
title('Magenta Part of Image');
subplot(3,3,6)
imshow(cmykImage(:, :, 3));
title('Yellow Part of Image');
subplot(3,3,8)
imshow(cmykImage);
title('CYMK Image');

% Converting RGB Image to HSV Image
hsvImage = rgb2hsv(rgbImage);

%Plotting different components using subplot
figure
subplot(3,3,2);
imshow(rgbImage);
title('Original RGB Image');
subplot(3,3,4);
```

```
imshow(hsvImage(:,:,1));
title('Hue Channel');
subplot(3,3,5);
imshow(hsvImage(:,:,1));
title('Saturation Channel');
subplot(3,3,6);
imshow(hsvImage(:,:,1));
title('Value Channel');
subplot(3,3,8);
imshow(hsvImage);
title('HSV Image');
```



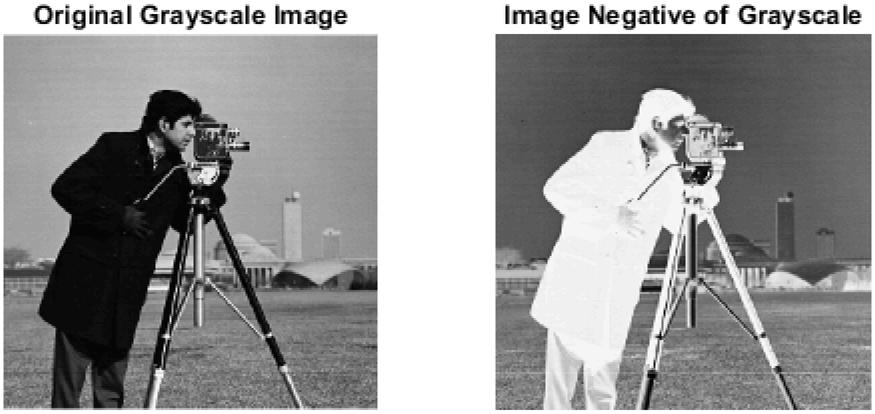

Original RGB Image
RGB to Gray using Average Method
RGB to Gray using Weighted Average Met


Original RGB Image
Cyan Part of Image
Magenta Part of Image
Yellow Part of Image
CYMK Image



Question 2. Find the Image Negative of a GrayScale Image.

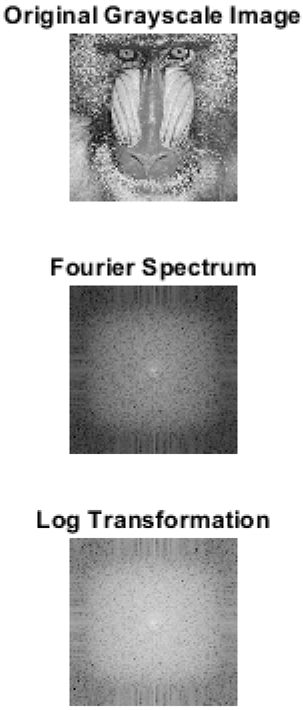
```
% Reading tif Image file format
grayImage = imread('cameraman.tif');

% Finding the Image Negative of the gray scal image
imageNegative = 255-grayImage;

% Plotting all the images using subplot
subplot(1,2,1)
imshow(grayImage);
title('Original Grayscale Image');
subplot(1,2,2)
imshow(imageNegative);
title('Image Negative of Grayscale');
```



Question 3. For a given gray scale image find 2D Fourier magnitude spectrum and apply Log transformation. Comment on Enhancement.

```
%Reading the grayscale image
grayImage = imread('mandril_gray.tif');

% Checking the dimensions of the image to verify the grayscale image.
if size(grayImage,3) == 3
  grayImage = grayImage(:, :, 2);
end

% Finding the fourier spectrum of the image.
Fourier = fft2(grayImage);
Spectrum = fftshift(log(1+abs(Fourier)));

% Applying log transformation of the Fourier spectrum.
logtrans = 2*log(1+(Spectrum));

% Plotting all the images using subplot
subplot(3, 1, 1);
imshow(grayImage, []);
title('Original Grayscale Image');
subplot(3, 1, 2);
imshow(Spectrum, []);
title('Fourier Spectrum');
subplot(3, 1, 3);
```

```
imshow(logtrans, []);
title('Log Transformation');
```



Original Grayscale Image

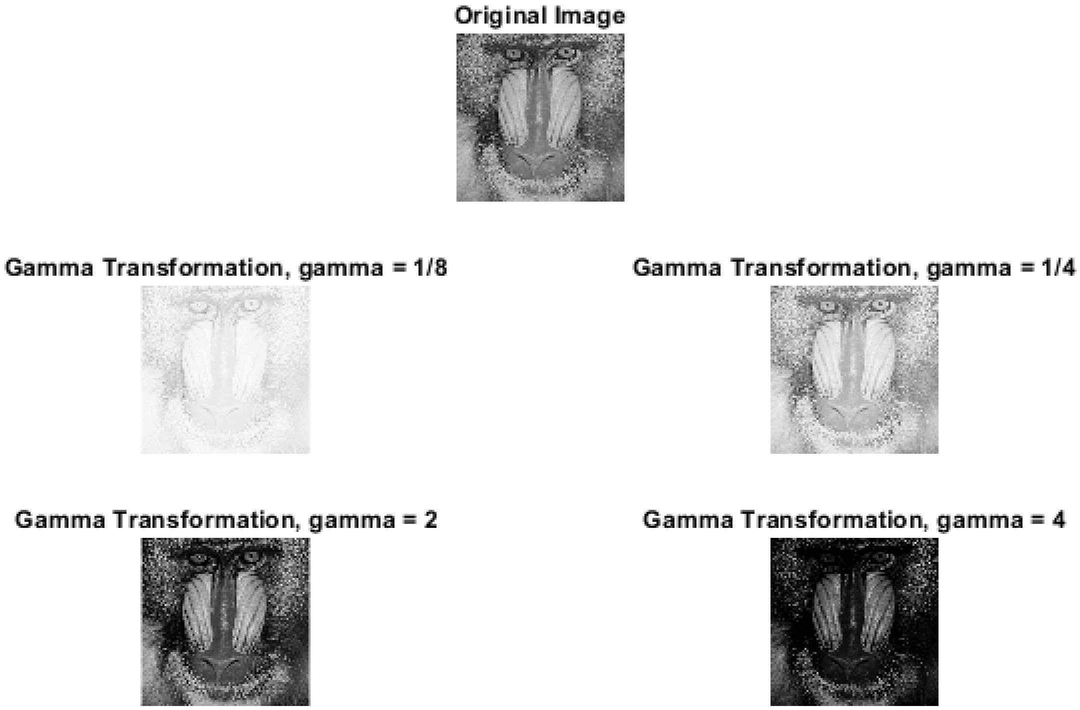
Fourier Spectrum

Log Transformation

Question 4:Implement Gamma Correction for images with over exposure or under exposure

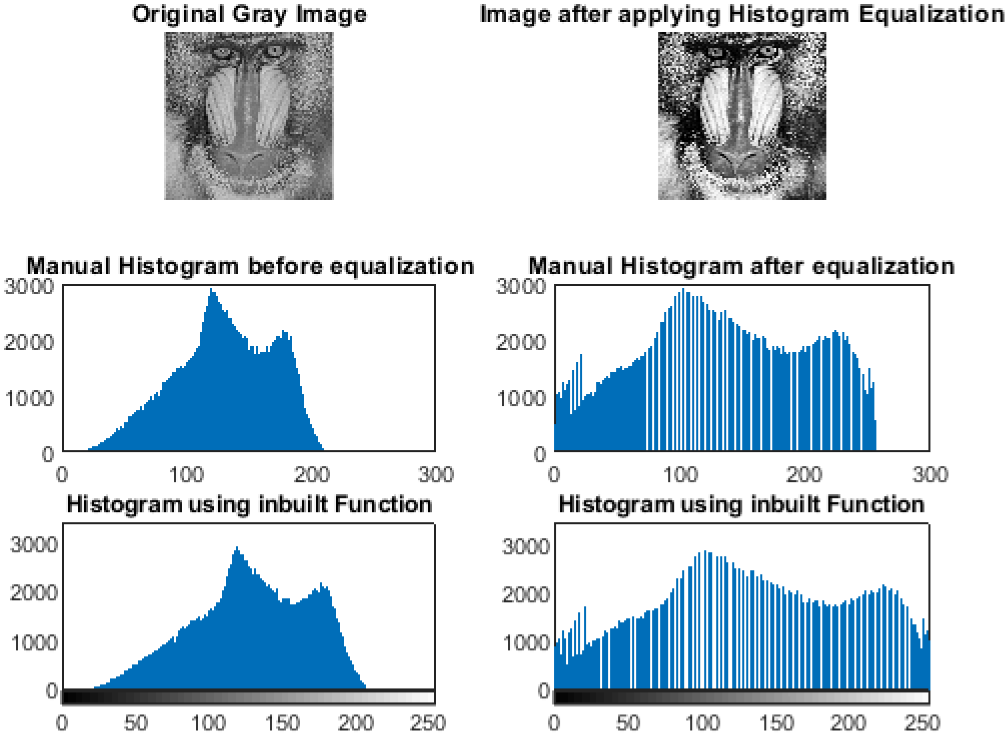
```
% Reading the gray scale image
Image = imread('mandril_gray.tif');

% Plotting the images using subplot.
subplot(3,3,2);
imshow(Image);
title('Original Image');
subplot(3,3,4);
gamma = 1/8;
G = double(Image).^(double(gamma));
imshow(G, [])
title('Gamma Transformation, gamma = 1/8');
subplot(3,3,6);
gamma = 1/2;
G = double(Image).^(double(gamma));
imshow(G, [])
title('Gamma Transformation, gamma = 1/4');
subplot(3,3,7);
gamma = 2;
G = double(Image).^(double(gamma));
imshow(G, [])
title('Gamma Transformation, gamma = 2');
```

```
subplot(3,3,9);
gamma = 4;
G = double(Image).^(double(gamma));
imshow(G, [])
title('Gamma Transformation, gamma = 4');
```


Original Image
Gamma Transformation, gamma = 1/8
Gamma Transformation, gamma = 1/4
Gamma Transformation, gamma = 2
Gamma Transformation, gamma = 4

```
% Question 5: Implement a Histogram Equalization for a gray scale
 image and observe the differences and histograms.

grayImage = imread('mandril_gray.tif');
[r, c] = size(grayImage);

%Finding Histogram from given image manually
freq = zeros(1, 256);
prob = zeros(1, 256);
histImage = zeros(size(grayImage));
newFreq = zeros(1, 256);
for i = 1:r
    for j = 1:c
        k = grayImage(i, j)+1;
        freq(k) = freq(k) + 1;
        prob(k) = freq(k)/(r*c);
    end
end

prob = freq/sum(freq);
```

```
for i = 2:256
    prob(i) = prob(i) + prob(i-1);
end

scaled = round(prob*255);
for i = 1:r
    for j = 1:c
        k = grayImage(i,j)+1;
        histImage(i, j) = scaled(k);
    end
end

for i = 1:r
    for j = 1:c
        k = histImage(i, j)+1;
        newFreq(k) = newFreq(k) + 1;
    end
end

histImage = uint8(histImage);
intensity = (1:256);

subplot(3,2,1);
imshow(grayImage);
title('Original Gray Image');
subplot(3,2,2);
imshow(histImage);
title('Image after applying Histogram Equalization');
%Plotting Histograms
subplot(3,2,3)
Hist = stem(intensity, freq);
set(Hist, 'Marker', 'none');
title('Manual Histogram before equalization');
subplot(3,2,4)
Hist = stem(intensity, newFreq);
set(Hist, 'Marker', 'none');
title('Manual Histogram after equalization');
subplot(3,2,5);
imhist(grayImage);
title('Histogram using inbuilt Function');
subplot(3,2,6);
imhist(histImage);
title('Histogram using inbuilt Function');
```



Published with MATLAB® R2020a