Q1:

CODE:Image Enhancement

clear all, close all ;

%Read a low-contrast grayscale image into the workspace and display it.

I = imread('pout.tif');

imshow(I)

%Adjust the contrast of the image so that 1% of the data is saturated at low and high intensities, and display it.

J = imadjust(I);

figure

imshow(J)

%Adjust the contrast of the image, specifying contrast limits.

K = imadjust(I,[0.3 0.7],[]);

figure

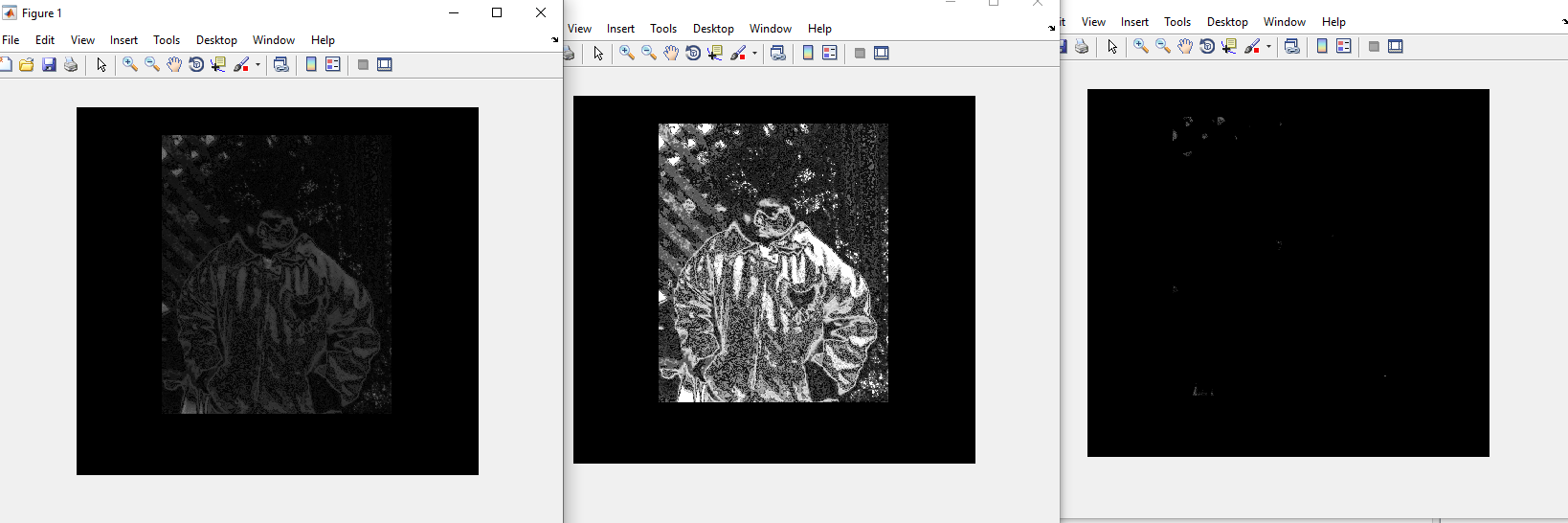
imshow(K)

%%%adjust contrast of a colour image

%RGB = imread('football.jpg');

%imshow(RGB)

OUTPUT:



CODE:Image Enhancement on colored image

%%%adjust contrast of a colour image

RGB = imread('stone.jpg');

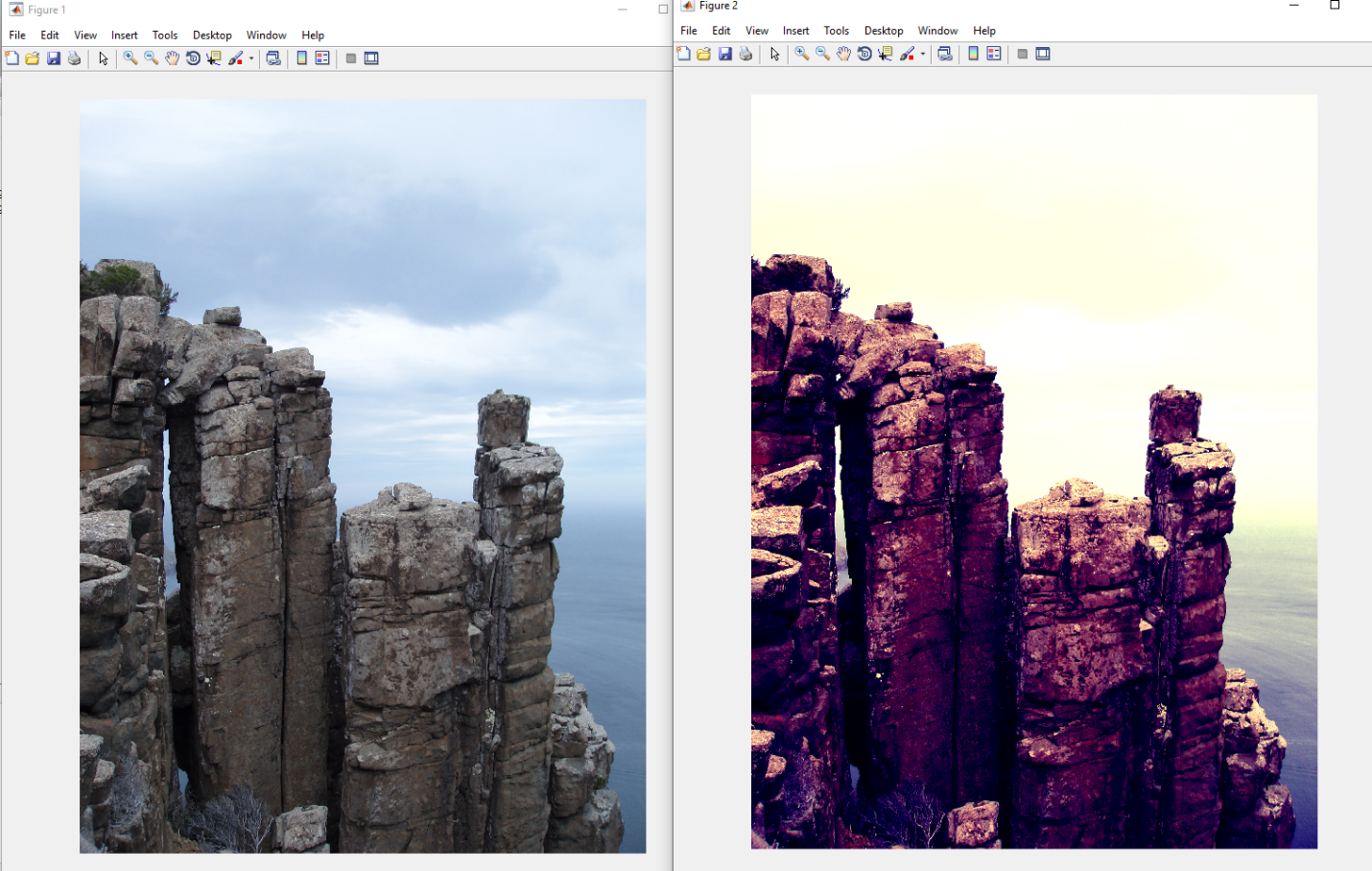
figure,imshow(RGB)

RGB2 = imadjust(RGB,[.2 .3 0; .6 .7 1],[]);

figure

imshow(RGB2)

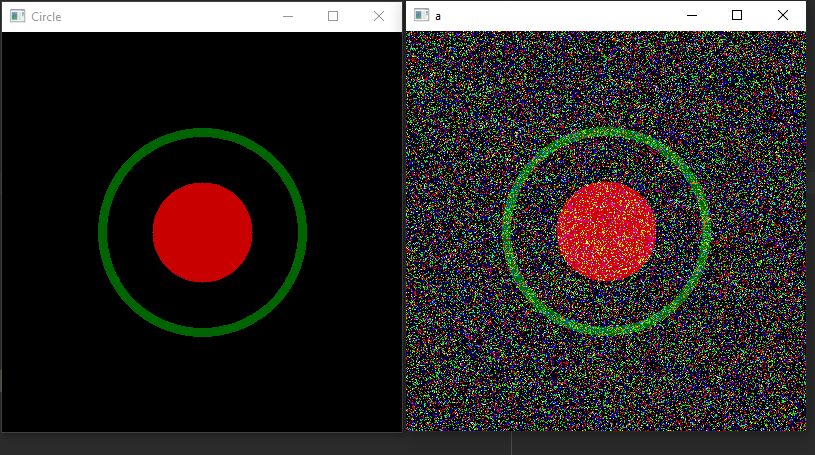
OUTPUT:



Q2:Generating & storing an image with .tif extension & introducing noise to it

import numpy as np  
import cv2  
import matplotlib.pyplot as plt  
import matplotlib.image as mpimg  
#create a 400x400 black image  
img=np.zeros((400,400,3),np.uint8)  
#############to convert to gray image  
gray=np.dot(img[...,:3], [0.299, 0.587, 0.144])  
#filled circle  
img1 = cv2.circle(img,(200,200),50, (0,0,200), -1)  
#filled circle  
img1 = cv2.circle(img,(200,200),100, (0,100,0), 8)  
#now use a frame to show it just as displaying a image  
cv2.imshow("Circle",img1)  
  
#storing image in tif form  
cv2.imwrite('output.tif', img1)  
########introducing gaussian noise  
# Generate Gaussian noise  
gauss = np.random.normal(0, 1, img1.size)  
gauss = gauss.reshape(img1.shape[0], img1.shape[1], img1.shape[2]).astype('uint8')  
# Add the Gaussian noise to the image  
img\_gauss = cv2.add(img1, gauss)  
# Display the image  
cv2.imshow('a', img\_gauss)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

OUTPUT:



Q3:Reducing intensity levels

clear all;

close all;

clc;

a=imread('output.tif');

grayimage=rgb2gray(a);

figure,imshow(grayimage);

%getting resolution of image; resolution is not equivalent to

%size=resolution\*bits per pixel

[r,c]=size(grayimage);

%reducing intensity levels to 16 by using nested loop

for i = 1 : r

for j = 1 : c

grayimage(i,j) = randi(16,1,1);

end

end

figure

imshow(grayimage);

%reducing intensity levels to 160 by using nested loop

for i = 1 : r

for j = 1 : c

grayimage(i,j) = randi(160,1,1);

end

end

figure

imshow(grayimage);

%down sampling starts

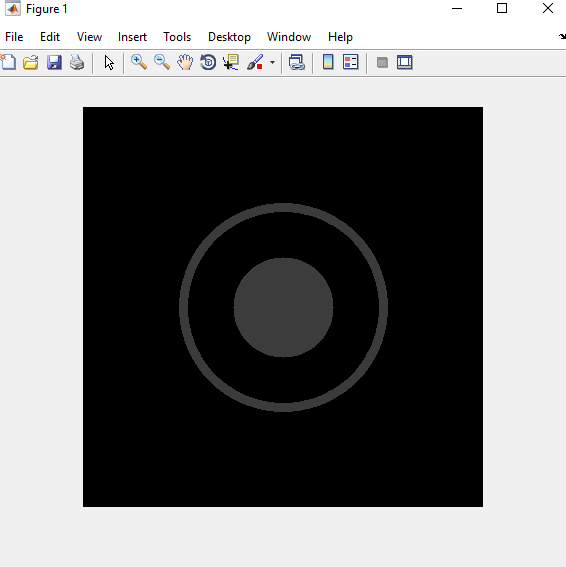
b1=grayimage(1:25:r,1:25:c);

figure,imshow(b1);

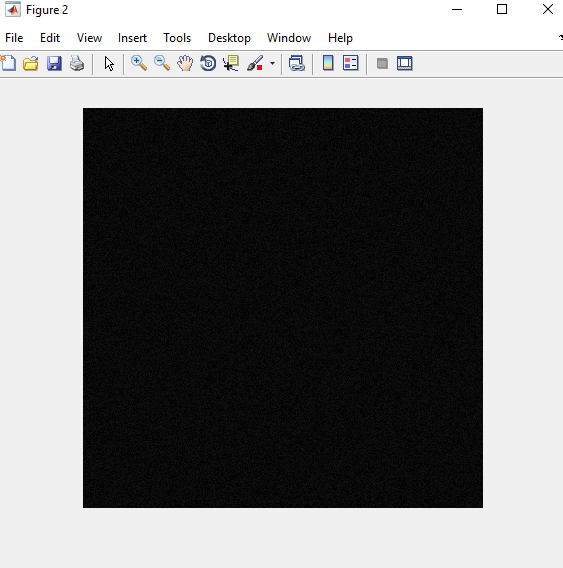
title('image:16\*16');

OUTPUT:

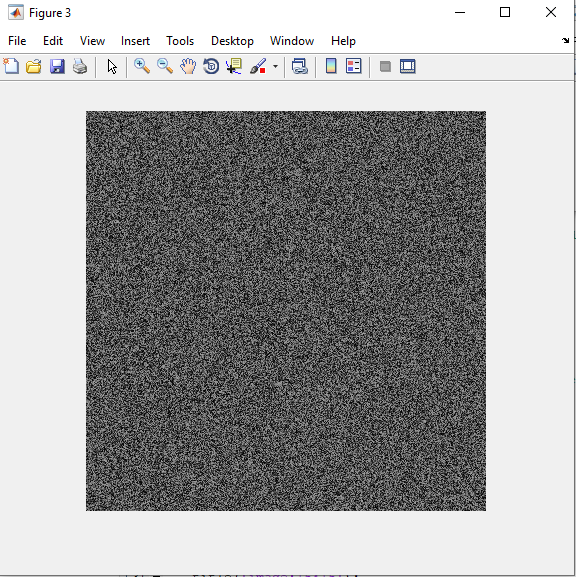
ORIGINAL IMAGE



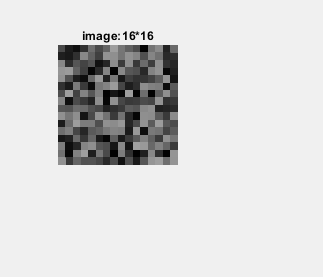
INTENSITY REDUCED TO 16 LEVELS



INTENSITY REDUCED TO 160 LEVELS



RESOLUTIION REDUCED TO 16\*16



***Conclusion***: I have completed all the tasks as assigned by the

Teacher.