MACHINE VISION

DIGITAL ASSIGNMENT – 1

SRIVIDHYA S

20BPS1119

**QUESTION:**

Write a MATLAB program for displaying Image Enhancement techniques.

**CODE:**

clc

clear all

warning off;

%to display color image from file

rgbImage = imread("peppers.png");

subplot(3,3,1)

imshow(rgbImage)

title('color image');

%color to grayscale

A = rgb2gray(rgbImage);

A=double(A);

B=bitget(A,1);

subplot(3,3,2)

imshow((B))

title('Bit plane 1')

B=bitget(A,2);

subplot(3,3,3)

imshow(B)

title('Bit plane 2')

B=bitget(A,3);

subplot(3,3,4)

imshow(B)

title('Bit plane 3')

B=bitget(A,4);

subplot(3,3,5)

imshow(B)

title('Bit plane 4')

B=bitget(A,5);

subplot(3,3,6)

imshow(B)

title('Bit plane 5')

B=bitget(A,6);

subplot(3,3,7)

imshow(B)

title('Bit plane 6')

B=bitget(A,7);

subplot(3,3,8)

imshow(B)

title('Bit plane 7')

B=bitget(A,8);

subplot(3,3,9)

imshow(B)

title('Bit plane 8')

**OUTPUT:**



**CODE:**

clc

clear all

%to display color image from file

rgbImage = imread("peppers.png");

subplot(3,3,1)

imshow(rgbImage)

title('color image');

%color to grayscale

grayImage = rgb2gray(rgbImage);

subplot(3,3,2)

imshow(grayImage);

title('color to grayscale');

%Contrast stretching

%decreases contrast of an image

subplot(3,3,3)

adjustImage = imadjust(rgbImage,[0 0.2],[0.5 1]);

imshow(adjustImage)

title('Decrease contrast');

%Grayscale threshold Transform (or) Gray Level Slicing

j=double(grayImage);

k=double(grayImage);

[row,col]=size(j);

T1=input('Enter the Lowest threshold value:');

T2=input('Enter the Highest threshold value:');

for x=1:row

for y=1:col

if((j(x,y)>T1) && (j(x,y)<T2))

j(x,y)=grayImage(x,y);

k(x,y)=255;

else

j(x,y)=0;

k(x,y)=0;

end

end

end

subplot(3,3,4)

imshow(uint8(j))

title('Graylevel slicing with bg')

subplot(3,3,5)

imshow(uint8(k))

title('Graylevel slicing w/o bg')

%Histogram equalization

%increases contrast of an image according to the nbins value given

nbins = 256

K = histeq(rgbImage,nbins);

subplot(3,3,6)

imshow(K);

title('histogram equalized');

%histogram plot of the high contrasted image

subplot(3,3,7)

imhist(K);

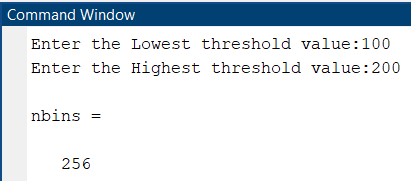
title('Hist plot of eq IMG');

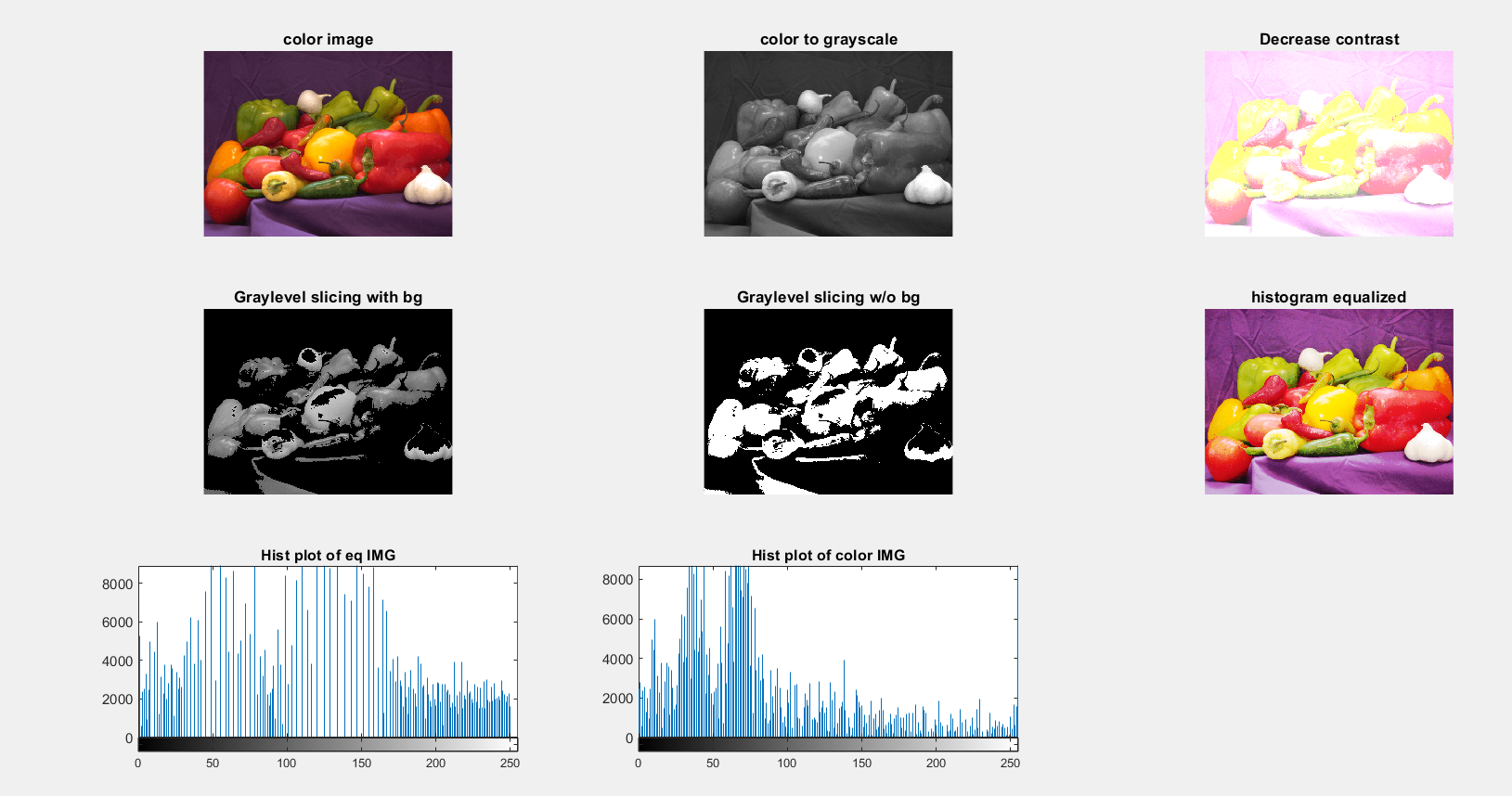
subplot(3,3,8)

imhist(rgbImage);

title('Hist plot of color IMG');

**OUTPUT:**





**Changes Suggested**

**CODE:**

clc

clear all

%Original Image

rgbImage = imread("peppers.png");

score = piqe(rgbImage)

subplot(2,2,1)

imshow(rgbImage)

title(['Original: score = ',num2str(score)])

%Noisy Image

Anoise = imnoise(rgbImage,'Gaussian',0,0.05);

score\_noise = piqe(Anoise)

subplot(2,2,2)

imshow(Anoise)

title(['Noisy: PIQE score = ', num2str(score\_noise)])

* **First, create a point-spread function, PSF, by using the [fspecial](https://in.mathworks.com/help/images/ref/fspecial.html) function and specifying linear motion across 21 pixels at an angle of 11 degrees. Then, convolve the point-spread function with the image by using [imfilter](https://in.mathworks.com/help/images/ref/imfilter.html).**
* **To reduce quantization errors, convert the image to double before calling imfilter.**

%Blurred Image - Gaussian Smoothing Filters

PSF = fspecial('motion',21,11);

Idouble = im2double(rgbImage);

Ablur = imfilter(Idouble,PSF,'conv','circular');

score\_blur = piqe(Ablur)

subplot(2,2,3);

imshow(Ablur)

title(['Blurred: PIQE score = ', num2str(score\_blur)])

%Sharpening Image

%Image without noise

ASharpen = deconvwnr(Ablur,PSF);

score\_sharp = piqe(ASharpen)

subplot(2,2,4)

imshow(ASharpen)

title(['Sharpened: PIQE score = ', num2str(score\_sharp)])

**OUTPUT:**



* **Based on the PIQE score, we can decide the quality of the image.**