**Image Processing Lab – Experiment 1**

Auroshis Ray

B317012

ETC

**TASK 1.1 WAP to create a Checkerboard image.**

**I = checkerboard(20,5,7);**

**figure, imshow(I)**

**Task: Write a 'user defined function() i.e. my\_checkerboard' instead of using the inbuilt ‘checkerboard()' function.**

**Hint: ones(20), Zeros(20)**

**Code (function)**

% Experiment 1 - creating images || Roll no- B317012

% write a user defined function 'my\_checkerboard' instead of using the

% inbuilt function

% 20 is number of pixels in each tile, 5 rows and 7 columns, each small

% box contains 20 pixels

%Hope this helps you.

function a = my\_checkerboard(n, p, m)

img = [zeros(n),ones(n);ones(n),zeros(n)]

pic = repmat(img, p, m)

imshow(pic)

end

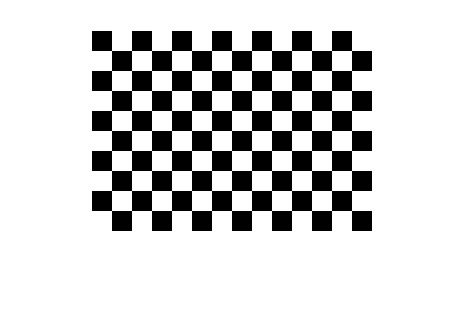
**Code (function)**

% B317012 driver code to test my\_checkerboard

clc;

my\_checkerboard(20,5,7)

**Output**

****

**TASK 1.2 Q. WAP to read, display and save an image**

**with a 'user defined name'. Read an image Display it**

**Save the image in a new name”B217XXX.jpg’**

**Code**

% B317012

%WAP to read, display and save an image

%with a 'user defined name'.

%Read an image

%Display it

%Save the image in a new name”B217XXX.jpg’

img = imread('dp.jpg')

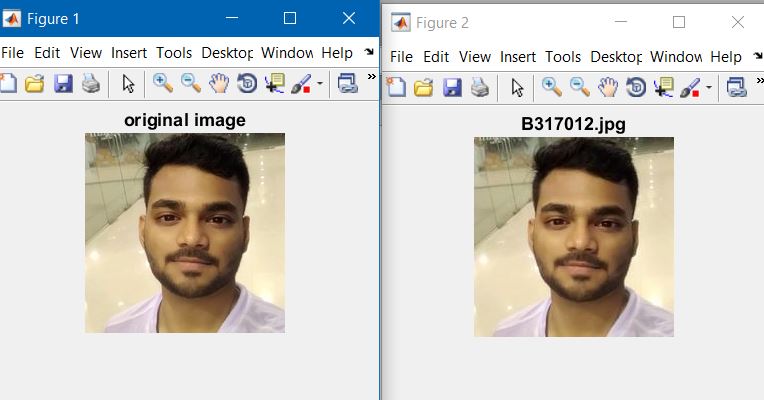
figure,imshow(img)

imwrite(img,'B317012.jpg')

new\_img = imread('B317012.jpg')

figure, imshow(new\_img)

**Output**

****

**TASK 1.3 WAP to subtract, add and scale the image.**

**Read an image, Subtract 100 from it and**

**display. Add 100 and display, Multiply with 0.7 and display**

**Subplot 2\*2**

**Code**

% B317012 addition, subtraction

% B317012

I = imread('dp.jpg') ;

I1 = I+100 ;

I2 = I-100;

I3 = I\*0.7;

subplot(2,2,1);

imshow(I) ;

title('original');

subplot(2,2,2);

imshow(I1) ;

title('addition');

subplot(2,2,3);

imshow(I2);

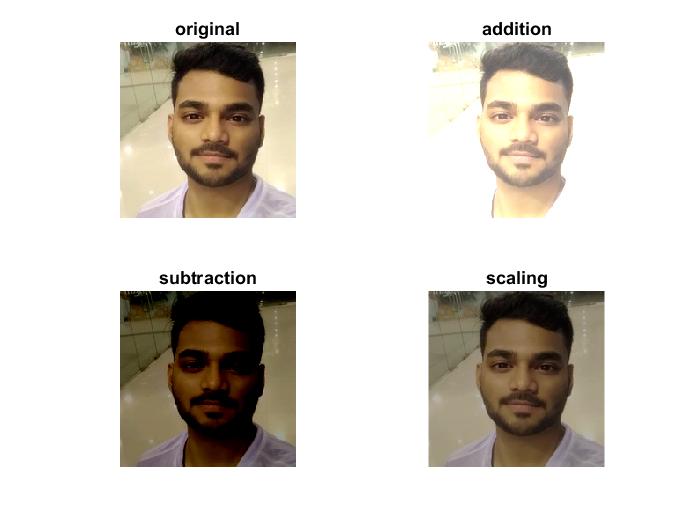
title('subtraction');

subplot(2,2,4);

imshow(I3);

title('scaling');

**Output**

****

**TASK 1.4.1 WAP to perform RGB to gray scale and then to binary image. Subplot, colour image, grayscale image, binary image**

**CODE**

% B317012 WAP to perform RGB to gray scale and then to binary image.

% Subplot, colour image, grayscale image, binary image

img = imread('peppers.png')

subplot(2,2,1);

title('original')

imshow(img)

rgb1 = rgb2gray(img);

subplot(2,2,2);

title('grayscale')

imshow(rgb1)

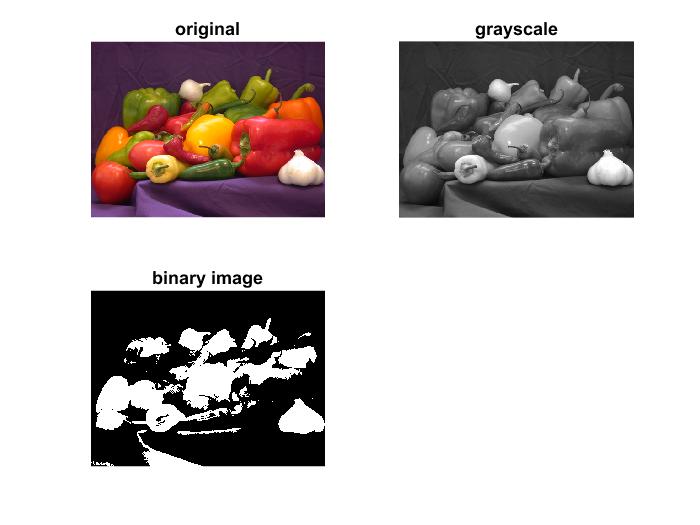
BW = im2bw(img,0.4)

subplot(2,2,3);

title('binary image')

imshow(BW)

**OUTPUT**

****

**TASK 1.4.2 Reshuffle the RGB components to create new color images. Subplot (2\*2), original image, GRB, BGR, BRG**

**CODE**

rgbImage = imread('peppers.png');

redChannel = rgbImage(:, :, 1);

greenChannel = rgbImage(:, :, 2);

blueChannel = rgbImage(:, :, 3);

subplot(2,2,1)

imshow(rgbImage)

title('RGB IMAGE')

subplot(2,2,2)

brgImage = cat(3, blueChannel,redChannel, greenChannel);

imshow(brgImage)

title('BRG IMAGE')

subplot(2,2,3)

grbImage = cat(3, greenChannel,redChannel, blueChannel);

imshow(grbImage)

title('GRB IMAGE')

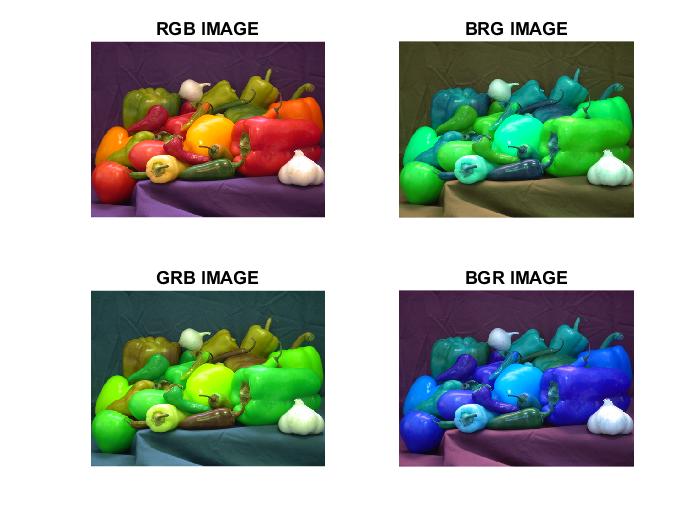
subplot(2,2,4)

bgrImage = cat(3, blueChannel,greenChannel,redChannel);

imshow(bgrImage)

title('BGR IMAGE')

**OUTPUT**

****

**TASK 1.5.1: WAP to Flip the image .**

**TASK 1.5.2: WAP to Mirror the image .**

**TASK 1.5.3: WAP to partition the image in 4 equal**

**parts.**

**CODE**

% B317012 flip, mirror and scale the image

img = imread('peppers.png');

subplot(2,2,1);

imshow(img)

title('original image')

subplot(2,2,2);

vertFlip\_img = flip(img, 1);

imshow(vertFlip\_img);

title('Vertically flipped image');

subplot(2,2,3);

mirror\_img = flip(img,2);

imshow(mirror\_img);

title('mirrored image')

**Dividing into 4 parts**

% B317012 divide into 4 parts

I = imread('B317012.jpg')

I1=I(1:size(I,1)/2,1:size(I,2)/2,:);

I2=I(size(I,1)/2+1:size(I,1),1:size(I,2)/2,:);

I3=I(1:size(I,1)/2,size(I,2)/2+1:size(I,2),:);

I4=I(size(I,1)/2+1:size(I,1),size(I,2)/2+1:size(I,2),:);

subplot(2,2,1);

imshow(I1);

subplot(2,2,2);

imshow(I3);

subplot(2,2,3);

imshow(I2);

subplot(2,2,4);

imshow(I4);

**OUTPUT**

