

# CO543 – Image Processing Lab 2

20/7/2017

## 1. Image thresholding

Write a function to perform image thresholding using point processing taking the image file and the threshold value from the user.

## 2. Image arithmetic operations

Read two images and perform addition and subtraction.

$I = I1 + I2$ ; # Addition of two

$I = I1 - I2$ ; # Subtraction of two images

Now, use inbuilt functions

- OpenCV function, **cv2.add()** or simply by numpy operation, **res = img1 + img2**. Both images should be of same depth and type, or second image can just be a scalar value.
- OpenCV function, **cv2.subtract()** or simply by numpy operation, **res = img1 - img2**.

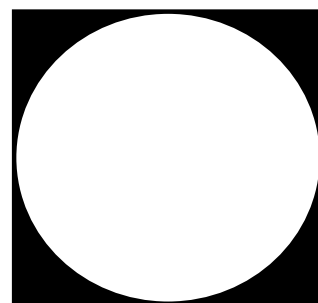
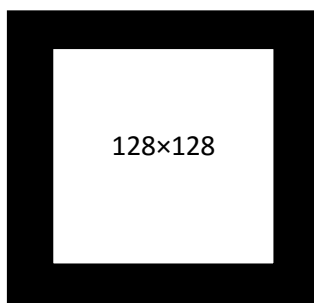
## 3. Write simple programs to demonstrate

- a. Log transformation
- b. Power transformation
- c. Gray level slicing
- d. Bit plane slicing

Show the original and resultant images in same figure to compare them easily.

## 4. Masking

Write a program to read any image, resize it to 256x256. Apply the masks shown in following figures so that only the middle part of the image is visible.



## 5. Brightness

Write your own Python OpenCV function `addbrightness()` and use it to increase brightness of given image.(Hint: Use Image arithmetic operations)

## 6. Histogram

- Histogram Calculation in OpenCV

Use inbuilt OpenCV **`cv2.calcHist()`** function to display the histogram of a given image.

- Histogram Calculation in Numpy

Use inbuilt numpy **`np.histogram()`** function to display the histogram of a given image.

Then write your own histogram functions for the following scenarios

a. Show a histogram plot for a grayscale image. b. Show three histograms for a given RGB image.