

# COM-SGN.110 Introduction to Image and Video Processing

## EXERCISE 1

This exercise is an introduction to MATLAB Image Processing Toolbox **Image basics and useful commands**

Load the image 'peppers.png' as matrix `I`. (Hint: `imread`)

You can use the function `imshow` to display images in MATLAB.

- Convert the color image to a grayscale image `I1` and display it. (Hint: `rgb2gray`)
- Using MATLAB indexing, display only the red (1st) component of the original image. Compare it with the previous result. (Call the function `figure` before displaying an image to create a new window for it.)
- Make a copy of the original image and name it `I2`. Add value 50 to the green component of `I2`. Display the result and compare with the original image.
- Store the Red, Green and Blue components of the original image as matrices `R`, `G`, `B`. Create an image `I3` by recombining the components in the order BGR. (Hint: `cat`)
- Display 4 images in a 2-by-2 layout in the order of top left to bottom right: `I`, `I1`, `I2`, `I3`. Make sure image content is visible. (Hint: `subplot`)

### 2. Batch processing

- Clean the workspace memory and close all the figures in the previous task. (`clear all, close all`).
- Download 6 images (inside `Ex1.zip`) to your local path in MATLAB. Create a new m-file 'Ex0\_batch.m'.
- Create a function called 'Process' in this m-file. The function takes an image matrix `I` as an argument and performs the following operations on it:
  - scale down to 75% of the original size (`imresize`)
  - take the right half of the image and mirror it along the center (`size, fliplr`)
  - rotate the flipped right half 90 degrees counterclockwise (`imrotate`)
  - return the result as the output
- In the same m-file, but above the function 'Process' create a function called 'Ex0\_batch' without arguments. The function should read the 6 images one at a time inside a for loop, call `Process` on each and then save the result with the same name, but in bitmap format (\*.bmp). (Hint: `imwrite, num2str`).

### 3. Binary processing

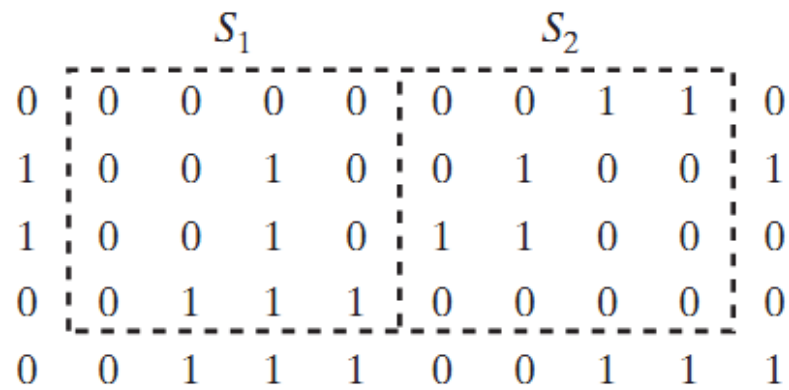


Figure 1. Example binary image

- Create a binary image as shown in Figure 1. Display it properly in only black and white colors without manually resizing. (Hint: `imshow`)
- Consider two image subsets as  $S_1$  and  $S_2$ . Implement a function with input parameter  $S$  to return the total number of non-zero values in a matrix. Print the output of your function for each subset using `fprintf`.
- Load the provided file 'S.mat', run your function from b) with new matrix  $S$  and show the result.