# SGN-12007 Introduction to Image and Video Processing

## EXERCISE 1

## 28.10.2019 - 30.10.2019

This exercise is an introduction to MATLAB Image Processing Toolbox. The tasks should be completed and presented to TA during the lab session. **Do not forget to upload your solutions to Moodle!** Questions about exercises should be addressed to the TA personally, through Moodle messages or via email, which can be found on the Moodle page of the course.

### 1. 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.

a. Convert the color image to a grayscale image `I1` and display it. (Hint: `rgb2gray`)
b. 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.)
c. 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.
d. 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`)
e. 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

a. Clean the workspace memory and close all the figures in the previous task. (`clear all, close all`).
b. Download 6 images (inside Ex1.zip) to your local path in Matlab. Create a new m-file '`Ex1_batch.m`'.
c. 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

d. In the same m-file, but above the function '`Process`' create a function called '`Ex1_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

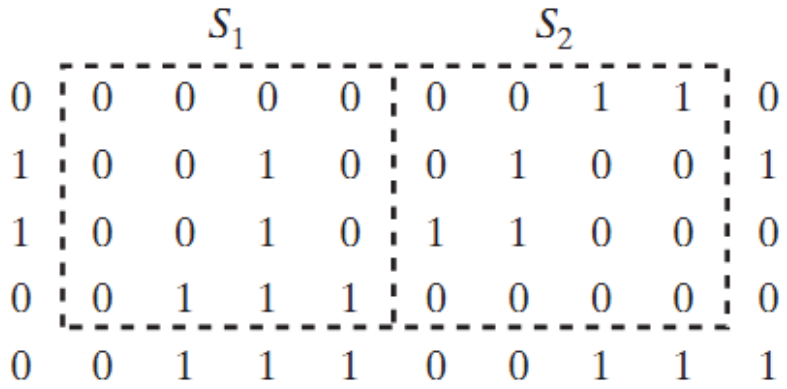


Figure 1. Example binary image

a. Create a binary image as shown in Figure 1. Display it properly in only black and white colors without manually resizing. (Hint: `imshow`)
b. Consider two image subsets as `S1` and `S2`. 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`.
c. Load the provided file 'S.mat', run your function from b) with new matrix `S` and show the result.