Sensors and Control for Mechatronics Systems
Tutorial 1

1. Matlab Image Processing ToolBox

Section 1 : Read images from a file

1.1 : Use *imread* function to read an image file and store it in the workspace.

1.2 : Display the image using *imshow* function.

1.3 : Observe how the image is stored in the Matlab workspace. What is the data structure used?

Section 2 : Image types in Matlab

2.1 : Read about the image types available in Matlab at :
https://au.mathworks.com/help/images/image-types-in-the-toolbox.html.
What is the type of the image you stored in the workspace in 1.3?

Image Type	**Interpretation**
Binary (Also known as abilevelimage)	Logical array containing only 0s and 1s, interpreted as black and white, respectively.
Indexed (Also known as a pseudocolor image)	Array of class logical, uint8, uint16, single, or double whose pixel values are direct indices into a colormap. The colormap is an m-by-3 array of class double. For single or double arrays, integer values range from [1, p]. For logical, uint8, or uint16 arrays, values range from [0, p-1].
Grayscale (Also known as an intensity, gray scale, or gray level image)	Array of class uint8, uint16, int16, single, or double whose pixel values specify intensity values. For single or double arrays, values range from [0, 1]. For uint8, values range from [0,255]. For uint16, values range from [0, 65535]. For int16, values range from [-32768, 32767].
Truecolor (Also known as an RGB image)	m-by-n-by-3 array of class uint8, uint16, single, or double whose pixel values specify intensity values. For single or double arrays, values range from [0, 1]. For uint8, values range from [0, 255]. For

	uint16, values range from [0, 65535].

2.2 : Write an algorithm to convert the image you stored in 2.3 into greyscale image. Note : Do not use the Matlab Image Toolbox native functions.

2.3 : Discuss your algorithms. Obtain a new greyscale image using *rgb2gray* function and compare your results.

2.4 : Write an algothim to convert the image you stored in 3.2 into a a binary image. Note : Do not use the Matlab Image Toolbox native functions.

2.5 : Discuss your algorithms. Obtain a new binary image using *im2bw* function and compare your results.

2.6 : Write your image data into files using *imwrite* function.

Section 3 : Image File Formats

3.1 Read ‘image1.png’ image into Matlab workspace. Note it’s file size on the disk.

3.2 Write the image back to the disk in ‘JPG’ format as ‘image1.jpg’.

3.3 Read the ‘image1.jpg’ image back into Matlab workspace.

3.4 Use *imshow* function to visually compare the two images. Discuss your observations. Note the file size of ‘image1.jpg’ on the disk.

3.5 Follow the same approach and convert ‘image2.jpg’ to ‘image2.png’. Observe the difference between the file sizes on the disk. Compare the two images visually and discuss your observations.

3.6 Discuss the pros and cons of lossless and lossy data compression for image storage.