Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Course No: CSE 4228

Course Title: Digital Image Processing Lab

LAB MANUAL – 2

Objective:

The objective of this lab session is to getting familiar with MATLAB and Image Processing Toolbox. This will cover the following topics –

1) MATLAB scripts, m-files
2) Control statements, functions
3) Introduction to Image Processing Toolbox.

Software: MATLAB (any version higher than 2009a)

Pre-requisite: Basic on MATLAB, Fundamental concept of digital images, C/ C++.

Getting started with m-File:

Create a text file with .m extension. Now you can put multiple commands inside of this file and run.

myscript.m

```
1 -    x = 1:50;
2 -    y = x.^3;
3
4 -    figure; plot(x,y,'.-b');
5
6 -    hold on;
7
8 -    y = (x+10).^3;
9 -    plot(x,y,'o-r')
10
11 -   hold off;
12
13 -   figure; bar(x,y)
14
```

Control Statements:

Sample code - 1:

```
A = floor(rand(5,5)*10);

B = ones(5,5)*9;

C = A + B;

[row, col] = size(C); % size() returns the dimension of a matrix

D = zeros(row, col);


for i = 1:row
    for j = 1:col

        if i==j
            D(i,j) = (C(i,j));
        end

    end
end

disp(D) %disp() prints a variable in command
```

Sample Code-2:

Sample code -1 can be done easily without control statements.

```
%% Same task is done without loop + if
A = floor(rand(5,5)*10);

B = ones(5,5)*9;

C = A + B;

[row, col] = size(C);

I = eye(row, col);

D = C.*I;

disp(D)
```

Getting started with image processing toolbox:

It is convenient to keep the m-script and the image you want to work with in the same directory. Let's say we have `cameraman.png` in the same directory.

Now we will go quickly over the basic commands to get started with image processing.

```
I = imread('cameraman.png');
```

`I` will contain the matrix of the image. In workspace, you should see variable `I` with 256 x 256 unit8. That means, the image has 256 rows and 256 columns and every pixel occupies a space of unsigned 8 bit integer. You can double click on the variable and open the Variable panel. Here, every element is a pixel. Observe that every pixel's intensity is in [0 -255] as there are 8 bits assigned.

We can access pixel values just the way have access matrix eleements.

```
pix = I(1,1); % pixel values at (1,1)
```

Now,

```
figure; imshow(I); % it will display the image
```

Now let's dig more. Try to understand the following code.

```
I = imread('cameraman.png');

figure; imshow(I);

[row, col] = size(I);

K = uint8(ones(row, col));

for i = 1:row
    for j = 1:col
        K(j,i) = I(i,j);
    end
end

figure; imshow(K);
```

Q.1: By the way, can you do the same without a loop?

Moreover, you can write a matrix as an image formation on to you disk. For example, if we want to save the matrix `K` as an image named modified.jpg, we can use the following command.

```
imwrite(K, 'modified.jpg');
```

The image will be written on to the current directory.

There are other commands that can be helpful.

`imfinfo()` : retrieve the image information
`imtool()` : open GUI to explore the image

R,G,B channels:

Luckily the provide `cameraman.png` is a gray image. But real images are color image. In gray image, every pixel has one value [0 - 255]. In color image, every pixel has 3 values (R, G, B) and every one of these has the range [0 - 255].

Let's work with a color image.

```
I = imread('peppers_color.jpg');
```

Now, in workspace you will see `512 x 512 x 3 unint8`. We know the meaning, only change is – there is a 3rd dimension which stands for R,G,B. In MATLAB, red channel is `1`, green channel is `2`, and blue channel is `3`.

It will be more clear if we access a pixel value. Say, we use the command –

```
r = I(10, 10, 1);
g = I(10, 10, 2);
b = I(10, 10, 3);
```

Here, `r` will have the value at the row # `10`, column # `10`, in color channel # `1` (that means red).
`g` will have the value at the row # `10`, column # `10`, in color channel # `2` (that means green).
`b` will have the value at the row # `10`, column # `10`, in color channel # `3` (that means blue).

Q.2: Now, using three lines of code, can you store all the red values, all the green values and all the blue values in three separate matrices say – R, G and B? And plot display them separately?
Hints- the colon operator can be helpful here.

[END]

Prepared by-
Mohammad Imrul Jubair