CS517: Lab Session 1
Introduction to MATLAB for basic DIP

NOTE: Download MATLAB (recommended) or use online MATLAB (https://www.mathworks.com/products/matlab-online.html)

1. Running commands from Command Line

Load images into the workspace.

2) Writing a basic program in MATLAB
clc; Clears Command window
clear all; Clears workspace
close all; Closes all figures
3) Save the program as a .m file, eg. "FirstMatlabCode.m"
4) Run the code by pressing "F5" or alternately clicking on the green pointer button on the programming window.
5) **help <command_name>** in command window to find the documentation of any inbuilt function.
6) To put a comment within a line, type % followed by the comment text; MATLAB treats all the information after the % on a line as a comment. Example:

```
% This is a comment line in Matlab
```

2. Basic MATLAB commands

1) Declaring a numeric variable

```
Var1 = 5;   % The numeric value 5 is stored in Var1.
temp=[1 2 3 4 5];
```

2) Declaring a string variable

```
Str1 = 'string';
```

3) Displaying a string or text

```
disp('The value of the variable is');
```

4) Displaying a variable

```
disp(Var1)
```

Output : 5
5) Size of variable
size(Var1) , or it is only a vector **length(Var1)**
Output : 1
6) Details of a variable

```
whos Str1
```

Output:

Name	Size	Bytes	Class	Attributes
str1	1x6	12	char	

7) Maximum, Minimum, Mean, and Variance of a variable. Important when variable is a vector
Maximum : **max(VariableName);** %// Use temp here as variable name
Minimum : **min(VariableName);**
Mean : **mean(VariableName);**
Variance : **var(VariableName);**

3. Read image from file and save to file

ImageMatrix = imread('lena512.png'); %Reads image.
size(ImageMatrix); %Finds size of the image. # Display the size of rgb and gray image
image(ImageMatrix); %Displays image.
imshow(ImageMatrix); % Displays image, image processing toolbox needed to use this.
imwrite(ImageMatrix, 'image1new.png'); %Writes image into a file in png format.
imwrite(ImageMatrix, 'image1new.tiff'); %Writes image into a file in tiff format.

%%%%%%%%%%% Small Exercise %%%%%%%%%%%%%
%% Execute the following code and explain why B / Tb is showing up as white image

```
clear all;
A = imread('peppers.png');  % Reading a color image
imshow(A)
Ag=rgb2gray(A);
B=double(Ag);
C=B/255;
disp([ max(A(:)) max(Ag(:)) max(B(:)) max(C(:)) ])
figure;subplot(2,2,1);imshow(A);subplot(2,2,2);imshow(Ag);subplot(2,2,3);imshow(B);subplot(2,2,4);imshow(C);
imwrite(B,'temp1.tif');
imwrite(C,'temp2.tif');
Tb=imread('temp1.tif');
Tc=imread('temp2.tif');
figure;subplot(1,3,1);imshow(A);subplot(1,3,2);imshow(Tb);subplot(1,3,3);imshow(Tc);
```

4. Figure and subplots

figure; % will create the canvas
subplot(1,4,1); % 1, 4, 1-> #of rows, #of cols, index
imshow(ImageMatrix);
title('Original Image');

5. Submatrices and colon notation

Vectors and submatrices are often used in MATLAB to achieve fairly complex data manipulation effects. "Colon notation" (which is used both to generate vectors and reference submatrices) and subscripting by vectors are keys to efficient manipulation of these objects. Creative use of these features permits one to minimize the use of loops (which slows MATLAB) and to make code simple and readable. Special effort should be made to become familiar with them.

The expression 1:5 (met earlier in for statements) is actually the row vector [1 2 3 4 5]. The numbers need not be integers nor the increment one. For example,

0.2:0.2:1.2
gives [0.2, 0.4, 0.6, 0.8, 1.0, 1.2], and
5:-1:1
gives [5 4 3 2 1]. The following statements will, for example, generate a table of sines. Try it.
x = [0.0:0.1:2.0]';
y = sin(x);
[x y]
Note that since sin operates entry-wise, it produces a vector y from the vector x.

The colon notation can be used to access submatrices of a matrix. For example,

A(1:4,3)
is the column vector consisting of the first four entries of the third column of A. A colon by itself denotes an entire row or column:
A(:,3)
is the third column of A, and A(1:4,:) is the first four rows. Arbitrary integral vectors can be used as subscripts:
A(:,[2 4])
contains as columns, columns 2 and 4 of A. Such subscripting can be used on both sides of an assignment statement:
A(:,[2 4 5]) = B(:,1:3)
replaces columns 2,4,5 of b with the first three columns of B. Note that the entire altered matrix A is printed and assigned. Try it.
Columns 2 and 4 of A can be multiplied on the right by the 2-by-2 matrix [1 2;3 4]:

A(:,[2,4]) = A(:,[2,4])*[1 2;3 4]
Once again, the entire altered matrix is printed and assigned.
If x is an n-vector, what is the effect of the statement x = x(n:-1:1)? Try it.

6. For, while, if, scalar functions, vector functions

Check out the code snippets here:
http://www.math.ucsd.edu/~bdriver/21d-s99/matlab-primer1.html#s5

7. Function

The following function named **mymax** should be written in a file named mymax.m. It takes five numbers as argument and returns the maximum of the numbers.

Create a function file, named **mymax.m** and type the following code in it −

```
function max = mymax(n1, n2, n3, n4, n5)

%This function calculates the maximum of the
% five numbers given as input
max =  n1;
if(n2 > max)
```

```
  max = n2;
end
if(n3 > max)
  max = n3;
end
if(n4 > max)
  max = n4;
end
if(n5 > max)
  max = n5;
end
```

8. Save and Load workspace variables

Save All Workspace Variables to MAT-File

Save all variables from the workspace in a binary MAT-file, test.mat. If filename is a variable, use function syntax.

```
filename = 'test.mat';
save(filename)
```

Otherwise, you also can use command syntax.

```
save test.mat
```

Remove the variables from the workspace, and then retrieve the data with the load function.

```
clear
load('test.mat')
```

Save Specific Variables to MAT-File

Create and save two variables, p and q, to a file called pqfile.mat.

```
p = rand(1,10);
q = ones(10);
save('pqfile.mat','p','q')
```

MATLAB®saves the variables to the file, pqfile.mat, in the current folder.
You also can use command syntax to save the variables, p and q.

```
save pqfile.mat p q
```

Load All Variables from MAT-File

Load all variables from the example MAT-file, gong.mat. Check the contents of the workspace before and after the load operation.

```
disp('Contents of workspace before loading file:')
whos

disp('Contents of gong.mat:')
whos('-file','gong.mat')

load('gong.mat')
```

disp('Contents of workspace after loading file:')
whos

Load Specific Variable From MAT-File
Load only variable y from example file handel.mat. If the workspace already contains variable y, the load operation overwrites it with data from the file.
load('handel.mat','y')

You also can use command syntax to load some specific variables lets say the variable, y.
load handel.mat y

9. Resize / Varying spatial resolution of an image.

Task1:
a) Resize image **ngc6543a.jpg** in images folder using **imresize** function. NOTE: use help to know about the imresize function.
b) Resize **ngc6543a.jpg** in 4 different sizes, ***original, 256×256, 64×64, 16×16*** and plot those four imges using **subplot** function mentioned above.

10. Questions

1) Task1:

a) Read two images kidred.jpg and lena512.bmp in the images folder using above imread function.

b) Create a function file which will take two matrices (i.e. images) as input and return the Root Mean Square Error (RMSE).

2) Task2: Create m×n image with band of 5 pixel wide at the center (vertically and horizontally) shown below.