**EXPERIMENT- 2**

**OBJECT:** MATLAB Program to Perform basic operations on Images

**Software/Hardware used:**

* MATLAB 9.0 R2016a
* Personal Computer (64 bits)

**Steps to RUN Program:**

**Step-1** Go to Documents, then select MATLAB root directory

**Step-2** Create New FOLDER in MATLAB directory

**Step-3**  Open MATLAB

**Step-4** Go to New or Open any pre-existing Script or import that folder by opening with MATLAB

**Step-5** Write required code in EDITOR window.

**Step-6**  Click on RUN to run the command

**Step-7**  You will get the output result in Command window

[NOTE:- Image must be stored in the same root directory folder of MATLAB which you have imported.]

**Basic Operations on Images:-**

### Operation-1: Reading images

**THEORY:** Images are read in MATLAB environment using the function **‘imread.’**

Syntax of imread is:

imread(‘filename’);

where ‘filename’ is a string having the complete name of the image, including its extension.

**CODE:**

r = imread('penguin.jpg');

Please note that when no path information is included in ‘filename,’ ‘imread’ reads the file from the current directory. When an image from another directory has to be read, the path of the image has to be specified.

### Operation-2 Image Display

Images are displayed on the MATLAB desktop using the function imshow, which has the basic syntax:

imshow(f)

**CODE:**

r = imread('penguin.jpg');

imshow(r)

**Operation-3** Two Image in one Frame

**THEORY:** To display the images penguin.jpg & penguin.jpg together in a single figure, you need to give the following commands:

**CODE:**

A = imread('penguin.jpg');

B = imread('penguin.jpg');

imshow(r)

figure

subplot(1,2,1),imshow(A)

subplot(1,2,2),imshow(B)

### Operation-5 Writing images

**THEORY:** Images are written to the current directory using the imwrite function, which has the following syntax:

imwrite(f, ‘filename’);

This command writes image data ‘f’ to the file specified by ‘filename’ in your current folder. The imwrite function supports most of the popular graphic file formats including GIF, HDF, JPEG or JPG, PBM, BMP, PGM, PNG, PNM, PPM and TIFF and so on.

The following example writes a 100×100 array of grayscale values to a PNG file named random.png in the current folder:

**CODE:**

F = rand(100);

imwrite(F, 'aa.jpg');

### Operation-6: Image information

File details including file name, data, size, format, height and width can be obtained using the command:

imfinfo('aa.jpg');

**Operation-7-** Image quality- To reduce the resolution of the photo

**THEORY:** Image quality is defined in terms of spatial resolution and quantisation.

Spatial resolution is the pixel density over the image. The greater the spatial resolution, the more are the pixels used to display the image. Spatial resolution is expressed qualitatively as dots per inch (dpi).

The following set of commands reduces the resolution of image ‘Penguins\_grey.jpg’ by half

**CODE:**

A = imread('penguin.jpg');

A1 = imresize((imresize(A,1/2)),2);

imshow(A1)

A = imread('penguin.jpg');

A2 = imresize((imresize(A,1/4)),4);

imshow(A2)

The following set of commands reduces the quantisation levels to 64 and displays the image:

A = imread('Penguin.jpg');

B=grayslice(A,64);

imshow(B,gray(64))

**Operation-8:** Histogram

**THEORY:** Histogram of a greyscale image represents the frequency of its grey levels occurrence. It is a graph indicating the number of times each grey level occurs in the image.

In MATLAB, the histogram can be viewed using the imhist function. As an example, the commands that follow can be used to display the histogram of image Penguins\_grey.jpg:

**CODE:**

A = imread('penguin.jpg');

figure(1), imhist(A);

**Operation-9:** Image sharpening

**THEORY:** Image sharpening is a powerful tool for emphasising texture and drawing viewer focus. It can improve image quality, even more than what is achieved through upgrading to a high-end camera lens.

The command used for sharpening an image in MATLAB is:  
B = imsharpen(A)

Following commands sharpen the image using the simple imsharpen command:

**CODE:**

a=imread('penguin.jpg');

imshow(a)

b=imsharpen(a);

figure,imshow(b)