Assignment-1

Read the image **'winter.jpg'**, crop it according to a rectangle [100 100 300 300] and show it. Write the cropped image with a recognized file format extension (say .tif) using "**imwrite**" function. Obtain image file details using '**imfinfo**' or '**whos**'.

Matlab Code:

```
%ass-1
x=imread ('F:\robotics ju\image processing\LAB 1ST SEM\dip lab images 1st
sem\winter.jpg');
figure,imshow(x);
size(x);
whos('x');
I=imcrop(x,[100 100 163 163]);
xx=size(I);
imwrite(I, 'F:\robotics ju\image processing\LAB 1ST SEM\dip lab images 1st
sem\winter1.jpg');
figure,imshow(I);title('Cropped Image');
imfinfo('file name with path and extension');
```

Output:

```
whos ('x')
  Name        Size                Bytes  Class    Attributes

  x         676x964x3           1954992  uint8
```

```
>> size(x)

ans =

   676   964     3
```

Original Image

Cropped Image

Assignment-2

Read the colour image 'waterlilies.jpg'. Show the image. Also convert it to grayscale and show the gray image with title "Gray Image". Flip the gray image using array indexing and show the flipped image with title "Flipped image". Obtain a sub-sampled image using array indexing and show it with the title "Sub-sampled Image". Again read the original color image 'waterlilies.jpg' and show it. Then use the command 'impixelinfo'.

Matlab Code:

```
x=imread('F:\robotics ju\image processing\LAB 1ST SEM\dip lab images 1st sem\waterlilies.jpg');
figure,imshow(x);
f=rgb2gray(x);title('Original Image');
figure,imshow(f);title('Gray Image');
fp=f(end:-1:1,:);
figure,imshow(fp);title('Flipped Image');
fs=f(1:2:end,1:2:end);
figure,imshow(fs);title('Subsampled Image');
x=imread('F:\robotics ju\image processing\LAB 1ST SEM\dip lab images 1st sem\waterlilies.jpg');
figure,imshow(x);
```

Output:

Original Image

Gray Image

Flipped Image

Subsampled Image

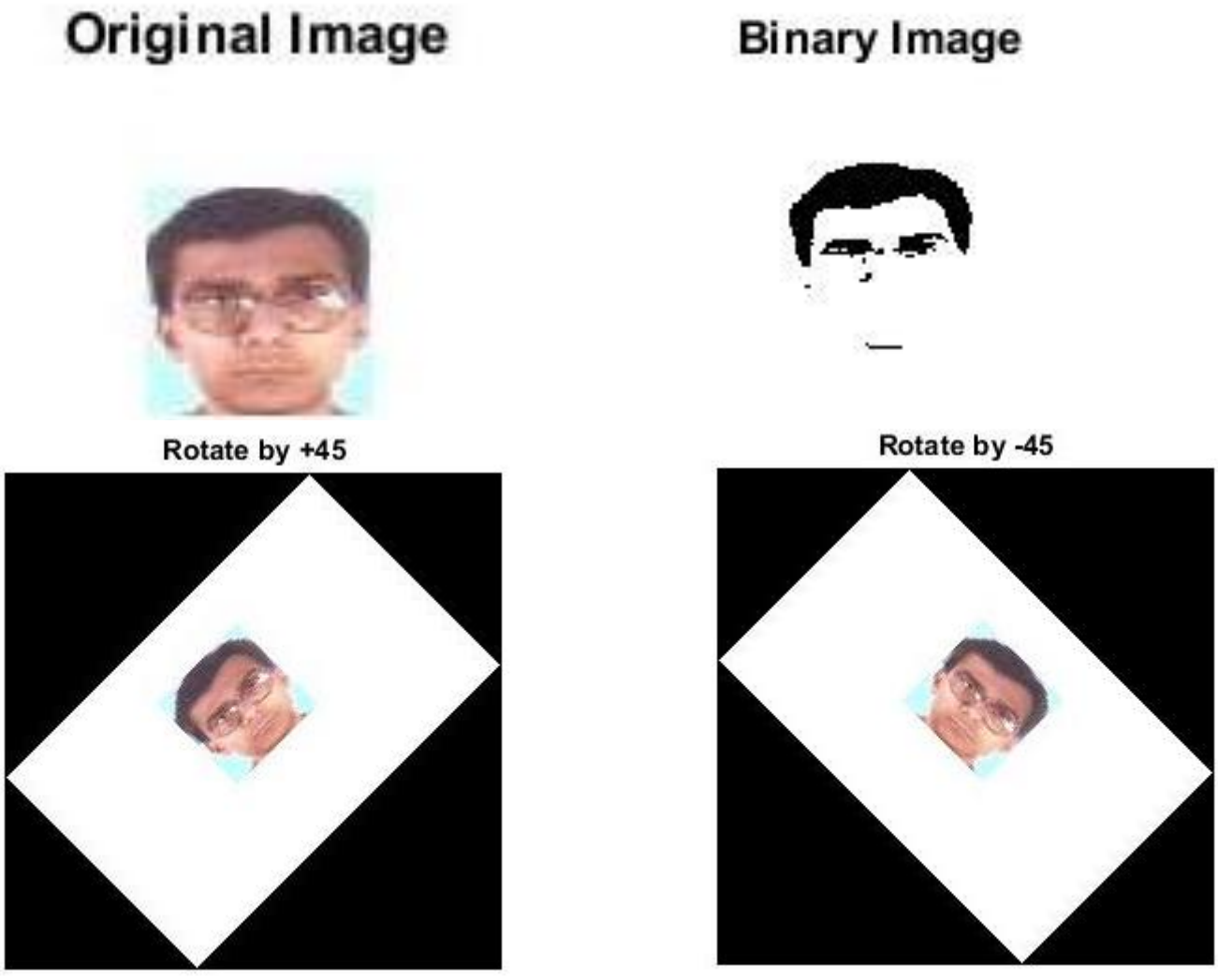
Assignment-3

Read the image ‘p0002.jpg’ and rotate it through a specified angle(say 45°, -45°) and show the images. Check whether image intensity values are of uint8 class. Convert the image intensity values to class double. Also convert the above image to binary image and show it.

Matlab Code:

```
J=imread('F:\robotics ju\image processing\LAB 1ST SEM\dip lab images 1st sem\p0002.jpg');
J1=imrotate(J,+45);
J2=imrotate(J,-45);
figure,imshow(J);title('Original Image');
figure,imshow(J1);title('Rotate by +45');
figure,imshow(J2);title('Rotate by -45');
whos('J');
Jd=im2double(J);
Jb=im2bw(J);
figure,imshow(Jb);title('Binary Image');
```

Output:




```
>> whos('J')
  Name        Size                Bytes  Class    Attributes

  J         144x228x3             98496  uint8
```

Demonstration Program:

a) M-function for generation of a sinusoidal image:

Write an M-function to implement the following 2-D image function

$$g(x,y)=A\sin(u_0x+v_0y)$$

for x=0,1,2,…….,M-1 and y=0,1,2,……..,N-1

Also introduce the timing function 'tic' and 'toe'.

The function inputs are A, u_o, v_o, M, N. The desired output is a sinusoidal image.

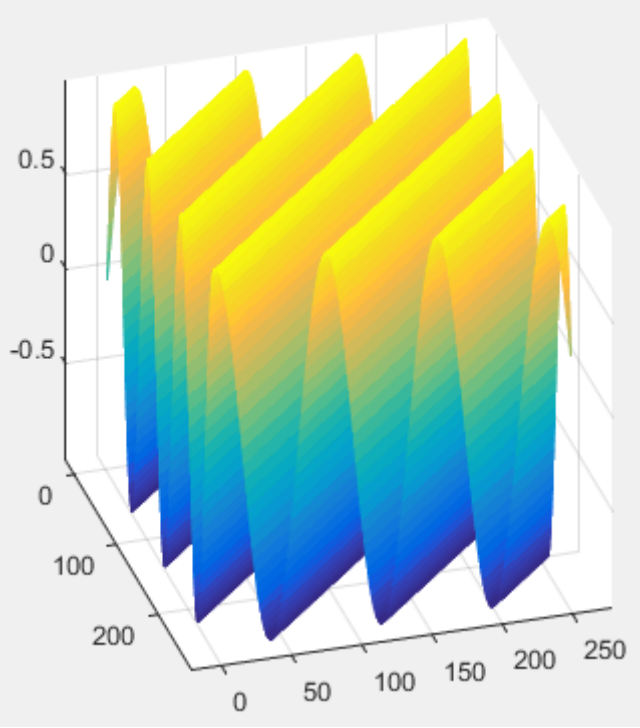
Also show the output image in 3D using 'mesh' function.

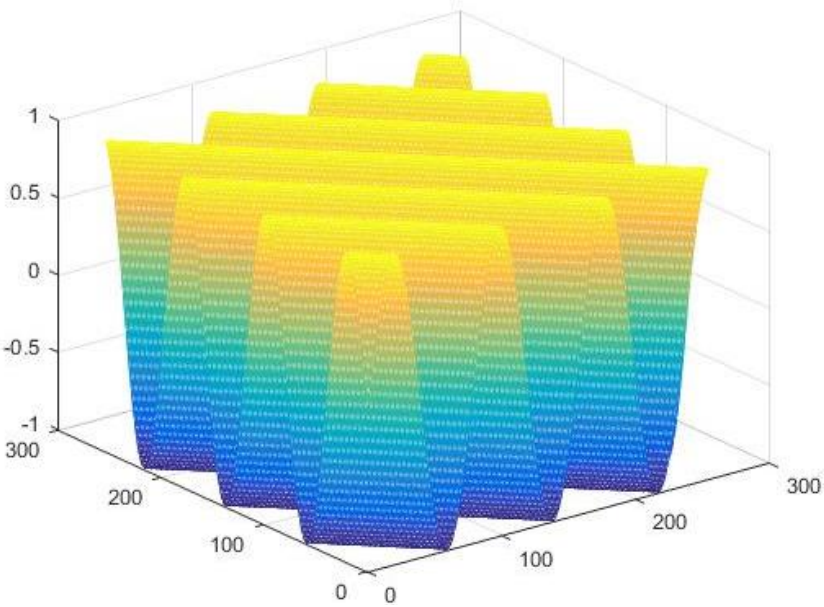
Input Data: A=1.0 u_o=1/(4*pi), v_o=1/(4*pi), M=256, N=256;

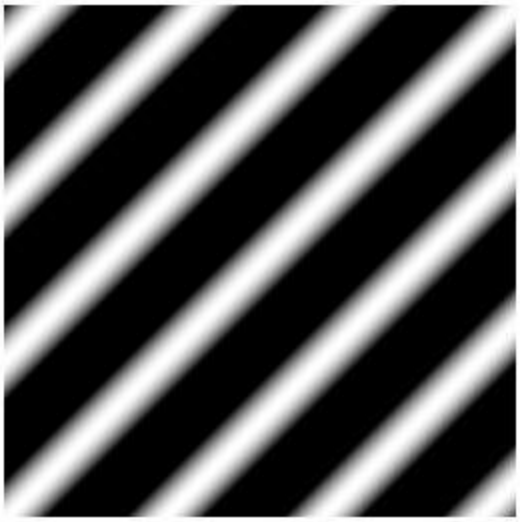
Matlab Code:

```
function[t1,g]=twodsin(A,u0,v0,M,N)
A=1.0; u0=1/(4*pi); v0=1/(4*pi); M=256; N=256;
tic
  r=0:M-1
  c=0:N-1
  [C,R]=meshgrid(c,r);
 g=A*sin(u0*R+v0*C);
 t1=toc
 imshow(g);
 figure, mesh(g)
```

Output:







b) Interactive I/O:s

Interactive M-function displays information and instructions to users and accept inputs from the keyboard. Using function 'input' and 'ginput', run the following program

Matlab Code:

```
X=imread('F:\robotics ju\image processing\LAB 1ST SEM\dip lab images 1st sem\p0002.jpg');
figure, imshow(X);
N=str2num(input('How many points you want to click:','s'));
[a,b]=ginput(N);
```

Output:

a [109;88;102;125;138]

b [35.0000;63.0000;67.0000;66.0000;53.0000]

Assignment-5

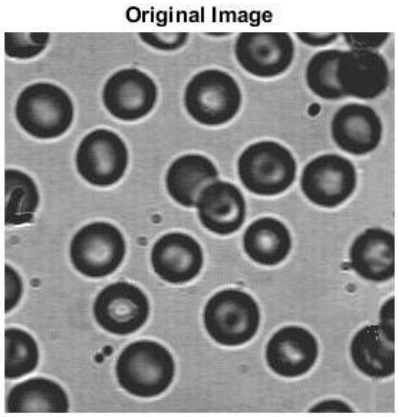
Generating and plotting Grayscale image histograms:

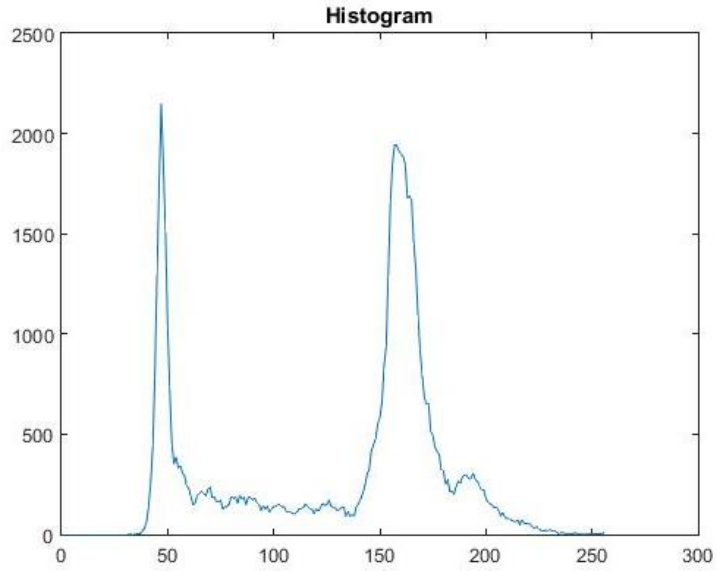
Read the image 'blood1.tif', assign it to f and show it. Find the gostogram of the gray image and assign the output as h and plot h in a separate figure window. Obtain the range of the horizontal and vertical axis of the histogram plot. Using 'axis' function, set the actual range of data for plotting. Set the axis of the currently shown figure manually with axis limits and tick marks. Use 'bar' function to plot the histogram. For this take histogram points to be plotted along vertical axis (length(h)=256) and intensity levels along horizontal axis.

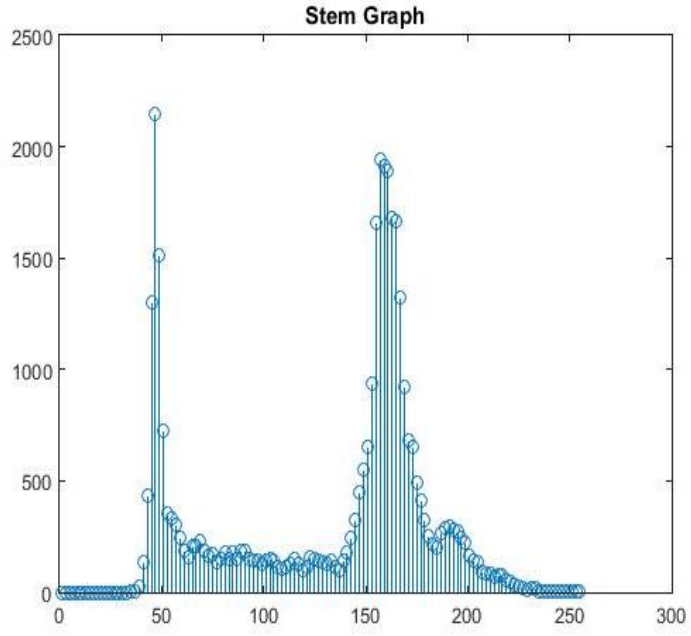
Matlab Code:

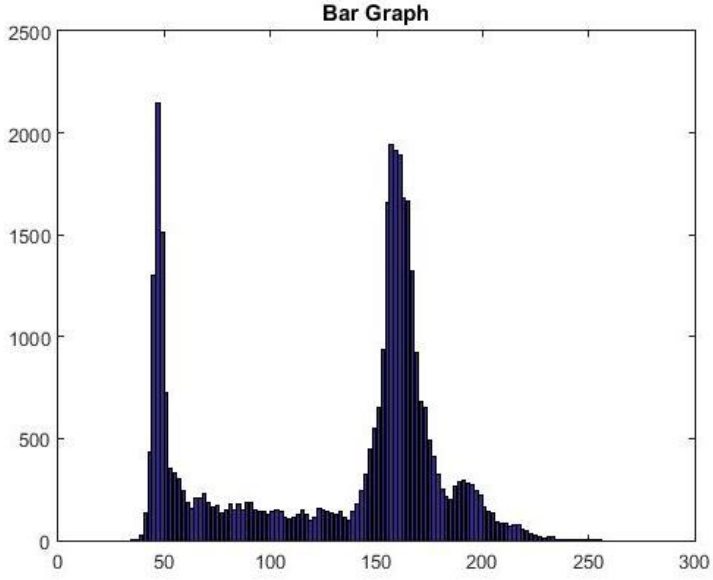
```
ff=imread('F:\robotics ju\image processing\LAB 1ST SEM\dip lab images 1st sem\blood1.jpg');
f=rgb2gray(ff);
figure,imshow(f);title('Original Image')
h=imhist(f);
figure,plot(h);title('Histogram');
axis([0 255 0 2500]);
set(gca,'xtick',0:30:255);
set(gca,'ytick',0:1500:1200);
vert=h(1:2:256);
horz=1:2:256;
figure,bar(horz,vert);title('Bar Graph');
figure,stem(horz,vert);title('Stem Graph');
```

Output:









Assignment-6

a) Histogram equalization is used for image enhancement. Perform the following steps:

i) Read the image 'blood1.tif' and assign it to f and show it.
ii) Plot the histogram of f.
iii) Set axis limits and tick marks automatically.
iv) Apply histogram equalization on f and assign the output as g.
v) Show the output(processed) image 'g'.
vi) Also show the histogramof 'g'.

Note that the increase in contrast is due to the considerable spread of the histogram over the entire intensity scale.

b) Demonstration Program: Enhancement program for histogram equalization.

In histogram equalization process the transformation function is simply the cumulative sum of the normalized histogram values. In the following program cdf is calculated for the above mentioned image and a plot of transformation function is displayed. We can tell visually from this transformation function that a narrow range of input intensity level is transformed into the full intensity scale in the output image.

Assignment-7

Generate a filter mask of type 'average' of size 5 and assign it as h. Read the image 'blood1.tif', and assign it to g and apply h to it and assign the filtered image as g1. Truncate all values of g1 to the nearest integer and then change its class to uint8. Show the result with the title 'Blurred Image'. Also show the original image with the title 'Original Image'.

Assignment-8

Image enhancement can be done by applying Laplacian filter mask. Perform the following steps.

Step-1: Read the image 'blood.tif' and give the title 'Original Image'. Change its class to double assign it as f.
Step-2: Generate a laplacian 3x3 filter with the value of alpha=0 and assign it as w4.
Step-3: Apply w4 to f and subtract the output from f and assign the result as g4.
Step-4: Show the image g4 with the title "Enhanced Image".
Step-5: Generate another form of Laplacian filter w8=[1 1 1;1 -8 1;1 1 1]
Step-6: Repeat Step-3 with w8 instead of w4 and assign the result as g8.
Step-7: Show the image g8 with the title "More Enhanced Image".

Display the two filtered image and notice that the later is more significantly sharpened than the former concluding that the Laplacian filter or second type is more effedcdtive in image enhancement.

Assignment-9

Consider the image 'bldg.jpg'. Corrupt it with salt and pepper noise in which both the black and white points have a probability of occurance of 0.2 using MATLAB function 'imnoise' and show the corrupted image.Then reduce the noise with the help of median filtering using the matlab function 'medifilt2' and show the resulting image.

Assignment-10: Sobel Edge Detector

Edge in an image can be detected in horizontal, vertical or both direction by 'edge' function. For this perform the following steps:

Step-1: Read the image 'bldg.jpg', show it and assign it as f after converting it to double.

Step-2: Apply 'edge' function on f with 'sobel' filter in vertical direction with automatic threshold, assign the output as gv and t.

Step-3: Repeat Step-2 for horizontal direction.

Step-4: Find the horizontal value 't' from the output of the edge function.

Step-5: Find the output gboth with specified threshold T and show gboth.

Assignment-10: Sobel Edge Detector

Edge in an image can be detected in horizontal, vertical or both direction by 'edge' function. For this perform the following steps:

Step-1: Read the image 'bldg.jpg', show it and assign it as f after converting it to double.

Step-2: Apply 'edge' function on f with 'sobel' filter in vertical direction with automatic threshold, assign the output as gv and t.

Step-3: Repeat Step-2 for horizontal direction.

Step-4: Find the horizontal value 't' from the output of the edge function.

Step-5: Find the output gboth with specified threshold T and show gboth.

Demonstration program: sobel edges at +45° and -45°

Matlab function "edge" does not compute sobel edges ±45°.To compute such edges we need to specify the mask and use "imfilter". The sobel mask which can compute edges at 45° direction is w45=[-2 -1 0;-1 0 1;-2 -1 0]. Using "imfilter" function we can get strong edges of "bldg..jpg" image oriented at 45°.

Strong edges of "bldg.jpg" image oriented at -45° can be oriented using the mask.

Wneg=[0 1 2;-1 0 1;-2 -1 0] with same commands.

Matlab code:

Demonstration program: sobel edges at +45° and -45°

Matlab function "edge" does not compute sobel edges ±45°.To compute such edges we need to specify the mask and use "imfilter". The sobel mask which can compute edges at 45° direction is w45=[-2 -1 0;-1 0 1;-2 -1 0]. Using "imfilter" function we can get strong edges of "bldg..jpg" image oriented at 45°.

Strong edges of "bldg.jpg" image oriented at -45° can be oriented using the mask.

Wneg=[0 1 2;-1 0 1;-2 -1 0] with same commands.

Matlab code:

Assign:11 (Log Edge Dtection)

Laplacian of Gaussian(Log) edge detection has two effects.It smoothes the image with Gaussian Filter and it computes the Laplacian which yields a double edge image. Locating edges then consists of finding the zero crossing between the double edges. :-

Step1: Read the image "bldg.jpg" and show it. Assign it to f1 after converting it to double.

Step2: Apply "edge" function on f1 with 'log filter". Find the threshold value "t' from the output of the "edge" function and display it.

Step3: Vary the value of t (a value less than t and another greater than t) and apply again the edge function on f1 with "log" filter providing two specified values of t as output parameter.

Step 4: Show the image.

Assign 12: Canny edge detection

The canny detector is the most powerful edge detector by function edge.

Apply canny filter on the same image "bldg..jpg" to detect its strong as well as following steps 1 to 4 in Assignment 10.

```
X=imread();
F1=double(x);
[gcan,t]=edge(f1,’canny’);
Disp(t);
g=edge(f1, ‘canny’,[0.08 0.23]);
figure,imshow(g);
t1=[0.15 0.3];
gcanny=edge(f1,’canny’,t1);
```