**AMITY SCHOOL OF ENGINERRING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

# LAB REPORT & ASSIGNMENTS

(ACADEMIC YEAR 2022-23)

COURSE NAME: DIGITAL IMAGE PROCESSING LAB

COURSE CODE: CSE2706

DEPARTMENT: CSE

FACULTY NAME: Dr Manjusha Joshi

**SUBMITTED BY**


STUDENT NAME: Vaibhav mahmia

ENROLLMENT NUMBER: A70405219061

CLASS: CSE (B.Tech)

SEMESTER: 7


DATE OF SUBMISSION: 02/12/2022

**AMITY SCHOOL OF ENGINERRING & TECHNOLOGY**
**AMITY INSTITUTE OF TECHNOLOGY**

# CERTIFICATE OF SUBMISSION

Student Name: vaibhav mahmia

Class: B.Tech (CSE)          Semester: 5

Enrolment Number: A70405219061

This is certified to be the bonafide work of student in "DIGITAL IMAGE PROCESSING LAB" Laboratory during the academic year 2022-23.

**Faculty In-charge**
{Department of CSE}
ASET, AUM

**Department**
**HoI**
{Department of CSE}
AIT, AUM
ASET, AUM

**Coordinate**

ASET &

**Date:**
**Stamp**

**AMITY SCHOOL OF ENGINERRING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

**(Academic Year 2021-22)**

# INDEX



**Faculty In-charge**
{Department of CSE}
ASET, AUM

**AMITY SCHOOL OF ENGINERRING & TECHNOLOGY**
**AMITY INSTITUTE OF TECHNOLOGY**

**(Academic Year 2022-23)**

# LAB 1
# Basic Image Transformation

Student Name:__________vaibhav mahmia__________

Class:______B. Tech CSE______Semester:____7____

Enrolment Number:________A70405219061________

**Faculty In-charge**
{Department of CSE}
ASET, AUM

| Name | Vaibhav mahmia |
|---|---|
| Enrolment Number | A70405219061 |
| Experiment Number | 1 |
| Batch | B |

## AIM OF THE EXPERIMENT:

To perform basic image transformation operations using MATLAB.

## THEORY:

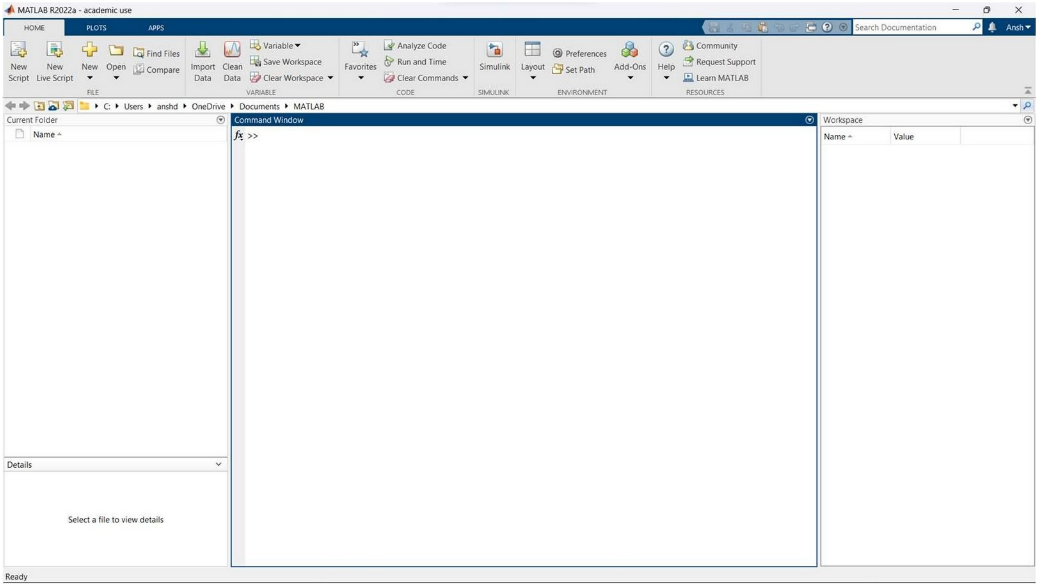
**Introduction to MATLAB**

**MATLAB** (an abbreviation of "**MAT**rix **LAB**oratory") is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

*L-shaped Membrane logo of MathWorks, developers of MATLAB*

Listed below are a few applications of MATLAB:

- Statistics and machine learning(ML)
- Curve fitting
- Control systems
- Signal Processing
- Mapping
- Deep learning
- Financial analysis
- Image processing
- Text analysis
- Electric vehicles designing
- Aerospace
- Audio toolbox




*MATLAB Workspace*

**Simple Image Transformation Operations using MATLAB**

Images in MATLAB

The basic data structure in MATLAB is the array, an ordered set of real or complex elements. This object is naturally suited to the representation of images, real-valued ordered sets of colour or intensity data.

MATLAB stores most images as two-dimensional matrices, in which each element of the matrix corresponds to a single discrete pixel in the displayed image. (Pixel is derived from picture element and usually denotes a single dot on a computer display.) For example, an image composed of 200 rows and 300 columns of different coloured dots would be stored in MATLAB as a 200-by-300 matrix.

Image Types in MATLAB

| **Image Type** | **Interpretation** |
| --- | --- |
| Binary Images | Image data are stored as an m-by-n logical matrix in which values of 0 and 1 are interpreted as black and white, respectively. |
| Indexed Images | Image data are stored as an m-by-n numeric matrix whose elements are direct indices into a colormap. Each row of the colormap specifies the red, green, and blue components of a single color. |
| Grayscale Images (intensity images) | Image data are stored as an m-by-n numeric matrix whose elements specify intensity values. The smallest value indicates black, and the largest value indicates white. |
| TrueColor Images (RGB images) | Image data are stored as an m-by-n-by-3 numeric array whose elements specify the intensity values of one of the three-color channels. For RGB images, the three channels represent the red, green, and blue signals of the image. |

| | |
|---|---|
| High Dynamic Range (HDR) Images | HDR images are stored as an m-by-n numeric matrix or m-by-n-by-3 numeric array, like grayscale or RGB images, respectively. HDR images have data type single or double, but data values are not limited to the range [0, 1] and can contain Inf values. |
| Multispectral and Hyperspectral Images | Image data are stored as an m-by-n-by-c numeric array, where c is the number of color channels. |
| Label Images | Image data are stored as an m-by-n categorical matrix or numeric matrix of nonnegative integers. |

Image Processing Commands

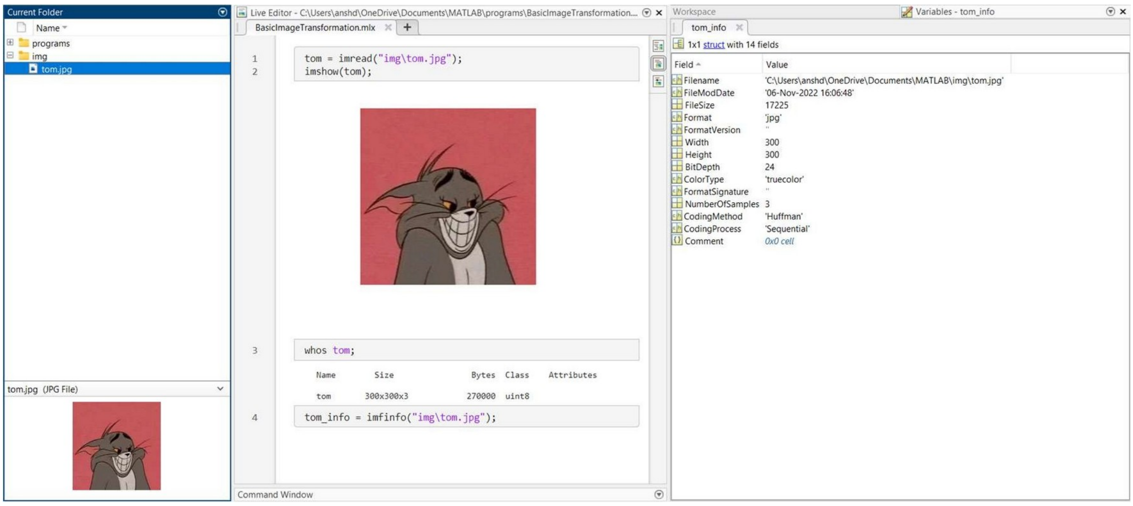
Some of the commands used for image transformation are given below. The **Image Processing Toolbox** add-on contains most of these commands.

| Command | Description |
|---|---|
| `imread` | Read an image into the workspace |
| `imshow` | Display the image |
| `whos` | Check how the `imread` function stores the image data in the workspace |
| `figure` | Create a new figure window using default property values |
| `imhist` | View the distribution of image pixel intensities |
| `histeq` | Improve the contrast in an image using histogram equalization |
| `imadjust`<br>`adapthisteq` | Other image contrast adjustment commands |
| `imwrite` | Write an image to a disk file |
| `imfinfo` | Return information about the image in the file, such as its format, size, width, and height |
| `imresize` | Return the resized or rescaled version of the input image |
| `imadd` | Add two images or adds a constant to image |
| `imsubtract` | Subtract one image from another or a constant from an image |
| `rgb2gray` | Convert RGB image or colormap to grayscale |
| `imsharpen` | Sharpen image using unsharp masking |

PROGRAM:

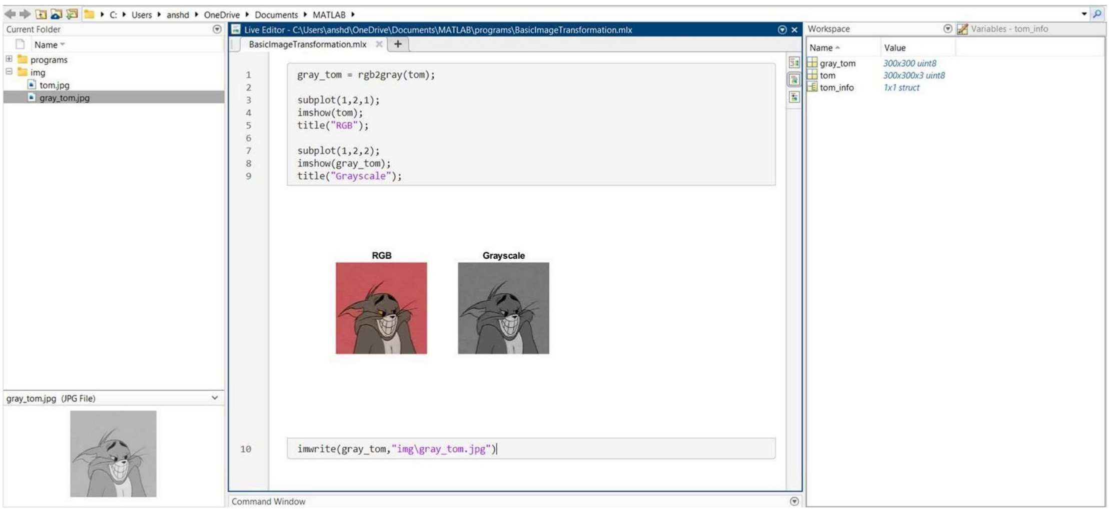
**Reading and displaying an image with file information**

```
tom  =imread(“i mg\
itom.jpg”); mshow(tom);
whos  tom;
tom_i nfo  =imfi nfo(“i mg\tom.jpg”);
```



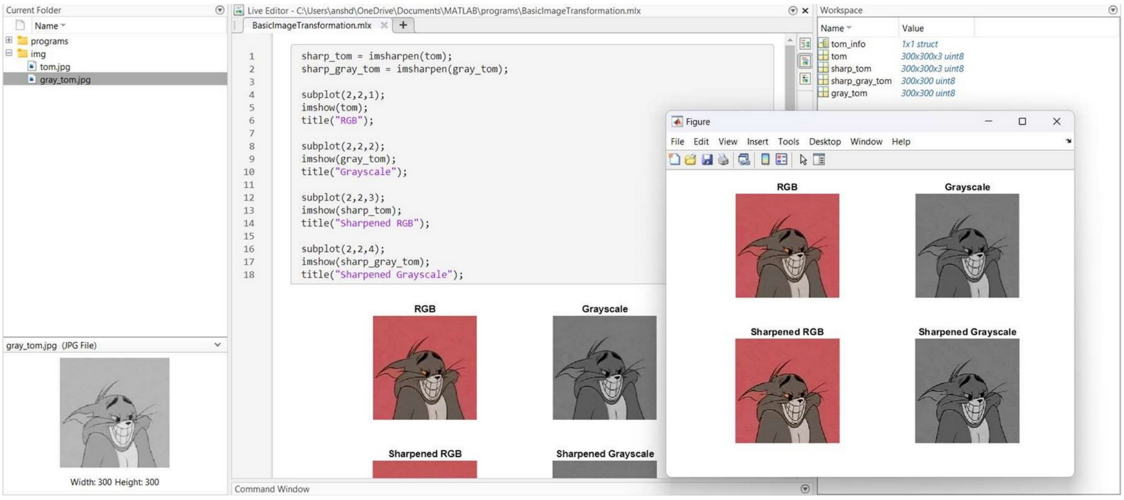

**Converting RGB image to grayscale and saving it to disk**

```
gray_tom =
rgb2gray(tom);
subplot(1,2,1);
imshow(tom)
;it
tle(“RGB”);
subplot(1,2,2)
i;
 mshow(gray_tom
 );
ti tle(“Grayscale”);
imwri te(gray_tomi,“ mg\gray_tom.jpg”);
```



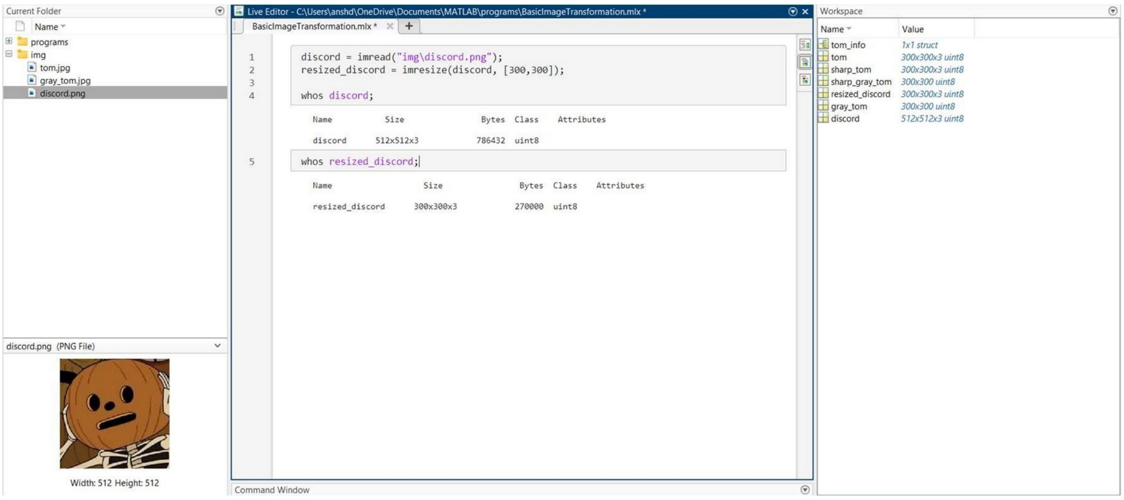
## Sharpening the images

```
sharp_tom =imsharpen(tom);
sharp_gray_tom =i
          msharpen(gray_tom);
subplot(2,2,1);
imshow(tom)
;it
tle("RGB");
subplot(2,2,2);
imshow(gray_tom
 );
ti
tle("Grayscale")
; subplot(2,2,3);
imshow(sharp_tom);
ti tle("Sharpened RGB");
subplot(2,2,4);
imshow(sharp_gray_tom);
ti tle("Sharpened Grayscale");
```



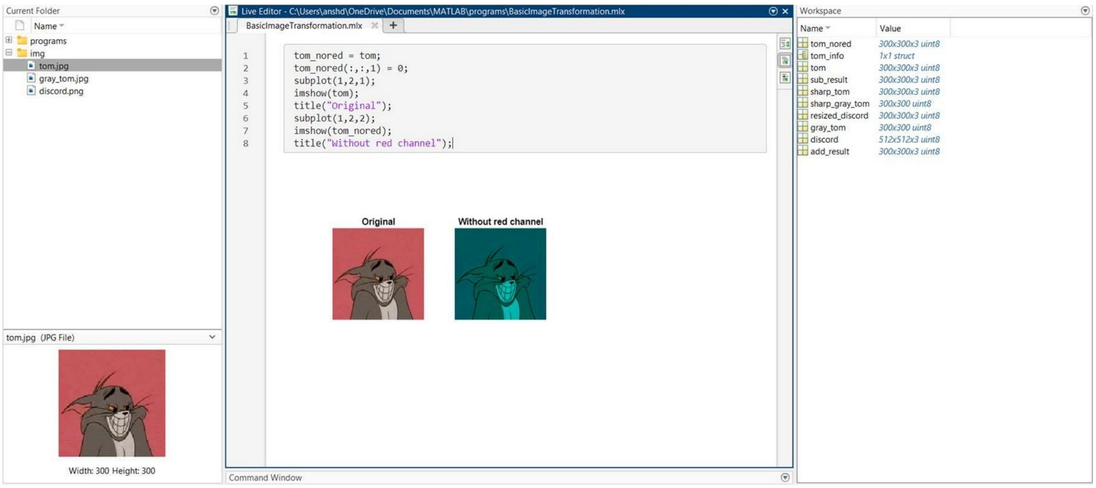
## Resizing an image

```
di scord =imread("i mg\di scord.png");
resi zed_di scordi = imresi ze(d scord,
[300,300]); whos d scord;
whos resi zed_di scord;
```



## Removing a channel from an image

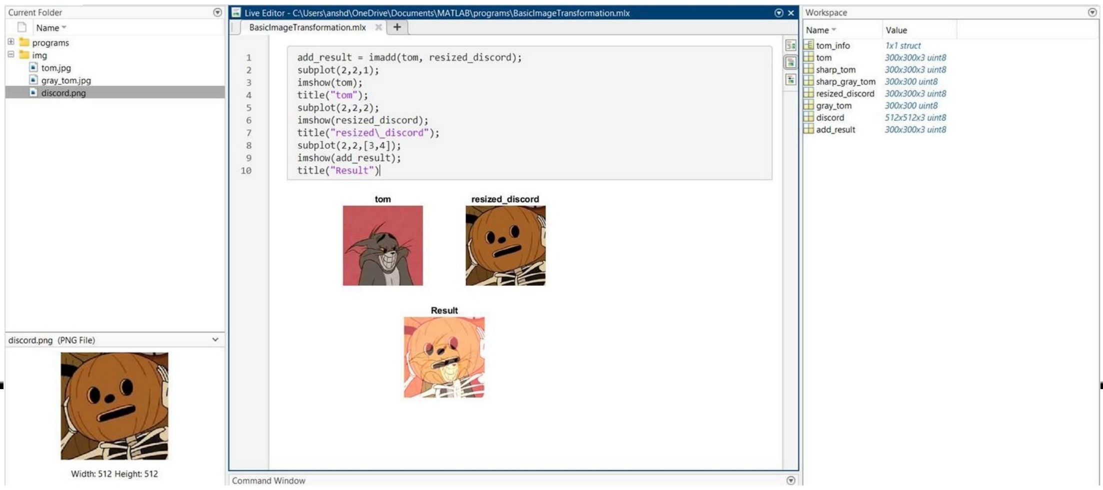
```
tom_nored  =  tom;
tom_nored(:,:,1)  =
0;
subplot(1,2,
i1);
 mshow(tom);
ti tle(“Orig  nal”);
subplot(1,2,2);
imshow(tom_nored
 );
ti tle(“W thout red channel”);
```



## Image arithmetic

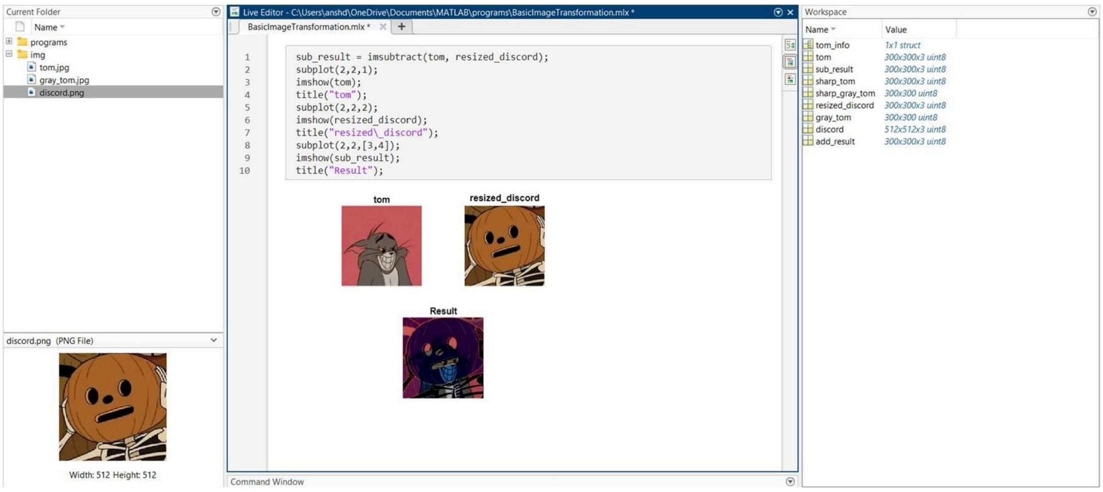
Adding two images

```
add_result =imadd(tom,  reis zed_d
scord); subplot(2,2,1);
imshow(tom)
;it
tle(“tom”);
subplot(2,2,2);
imshow(reis zedid
 scord);
title(“resi zed\_id
scord”); subplot(2,2,
[3,4]);
imshow(add_result
)i; t
tle(“Result”);
```



Subtracting two images

```
sub_result =imsubtract(tom, ries zeid_d
scord); subplot(2,2,1);
imshow(tom)
;it
tle("tom");
subplot(2,2,2);
imshow(resi zedid
 scord);
title("resi zed\_id
scord"); subplot(2,2,
[3,4]);
imshow(sub_result
)i; t
tle("Result");
```




## CONCLUSION:

Basic transformation operations on images have been performed and understood.

**AMITY SCHOOL OF ENGINERRING & TECHNOLOGY**

**AMITY INSTITUTE OF TECHNOLOGY**

**(Academic Year 2022-23)**

# LAB 2
# Image Blurring and Sharpening

Student Name: Vaibhav Mahmia

Class: B. Tech CSE Semester: 7

Enrolment Number: A70405219061

**Faculty In-charge**
{Department of CSE}
ASET, AUM

| | |
|---|---|
| Name | Vaibhav mahmia |
| Enrolment Number | A70405219061 |
| Experiment Number | 2 |
| Batch | B |

## AIM OF THE EXPERIMENT:

To perform blurring and sharpening operations on images.

## THEORY:

To blur images, the $fspecial$ function is used to create predefined two-dimensional filters including $average$, $disk$ and $motion$ filters.

$h = fspecial('average', hsize)$ íetuíns an aveíaging filteí $h$ of size $hsize$.

$h = fspecial('disk', radius)$ íetuíns a ciículaí aveíaging filteí (pillbox) within the squaíe matíix of size $2 \times radius + 1$.

$h = fspecial('motion', len, theta)$ íetuíns a filteí to appíoximate, once convolved with an image, the lineaí motion of a cameía. $len$ specifies the length of the $motion$ and $theta$ specifies the angle of motion in degíees in a counteí-clockwise diíection.

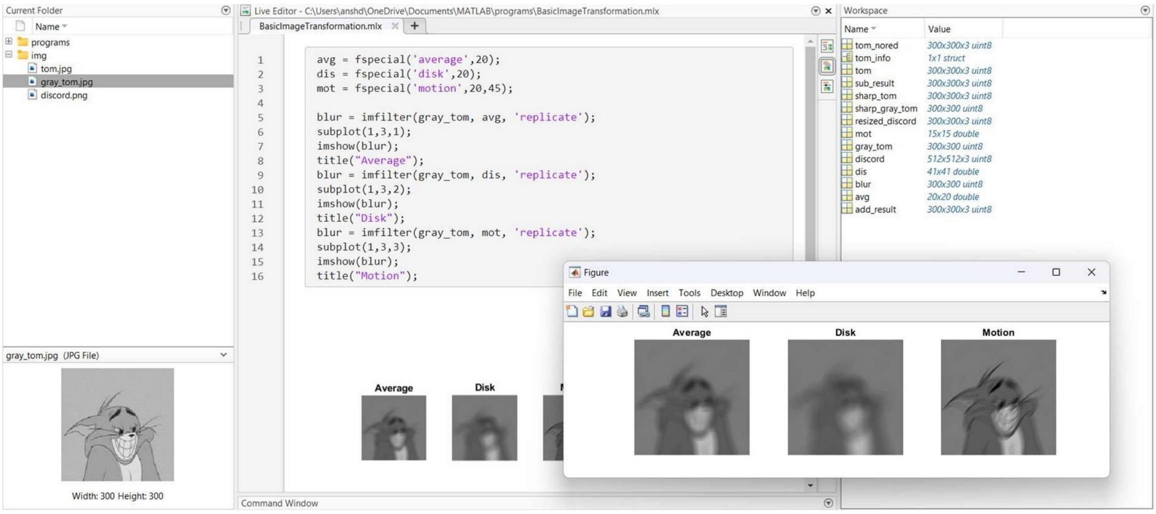
Ťhese cíeated filteís aíe then used in the $imfilter$ function. Ťhis function is used to peífoím $N$ dimensional filteíing of multidimensional images.

Ťo shaípen images, the $imsharpen$ function is used to shaípen images using unshaíp masking.

## PROGRAM:

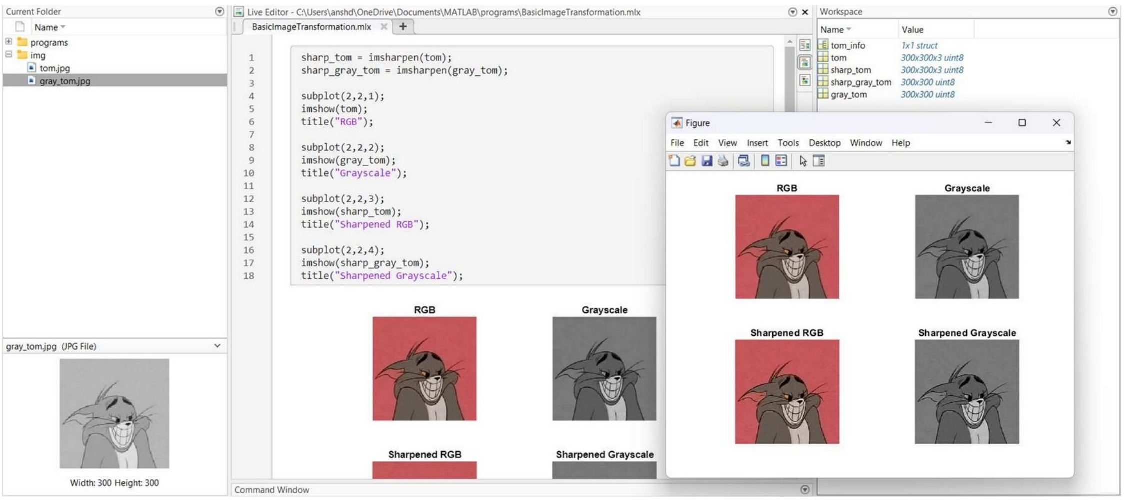
**Blurring images**

```
avg = fspiec
ail('averagie',20i); d s =
fspec al('d sk',20);
mot = fspiec al('imot on',20,45);
blur =imfi lter(gray_tom, avg, 'irepl
cate'); subplot(1,3,1);
imshow(blur);
ti tle("Average");
blur =imfi lter(gray_tom,i d s, 'irepl
cate'); subplot(1,3,2);
imshow(blur)
;it tlei("D
sk");
blur =imfi lter(gray_tom, mot, 'irepl
cate'); subplot(1,3,3);
imshow(blur);
ti tle("Moit on");
```




## Sharpening images

```
sharp_tom  =imsharpen(tom);
sharp_gray_tom =i
          msharpen(gray_tom);
subplot(2,2,1);
imshow(tom)
;it
tle("RGB");
subplot(2,2,2);
imshow(gray_tom
 );
ti
tle("Grayscale")
; subplot(2,2,3);
imshow(sharp_tom);
ti tle("Sharpened RGB");
subplot(2,2,4);
imshow(sharp_gray_tom);
ti tle("Sharpened Grayscale");
```




CONCLUSION:

The image has been blurred and sharpened using various functions.

**AMITY SCHOOL OF ENGINERRING & TECHNOLOGY**
**AMITY INSTITUTE OF TECHNOLOGY**

(Academic Year 2022-23)

# LAB 3
# Object Identification

Student Name:________________vaibhav mahmia_________________

Class:__________B. Tech CSE___________Semester:________7_______

Enrolment Number:______________A70405219061_________________

**Faculty In-charge**
{Department of CSE}
ASET, AUM

| | |
|---|---|
| Name | Vaibhav mahmia |
| Enrolment Number | A70405219061 |
| Experiment Number | 3 |
| Batch | B |

AIM OF THE EXPERIMENT:

Analyzing foreground objects in an image using MATLAB.

THEORY:

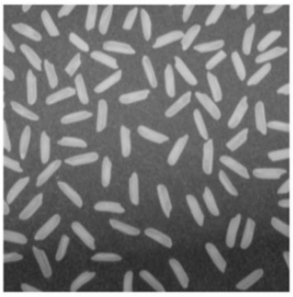
PROGRAM:

Reading the image

```
rice = imread('img\
```

**Preprocessing**

```
rice = imread('img\rice.png');
imshow(rice);
```



There is non-uniform illumination in the image. Preprocessing the image to make the background more uniform. Removing all the foreground (rice grains) using morphological opening. The opening operation removes small objects that cannot completely contain the structuring element.

Define a disk-shaped structuring element with a radius of 15, which fits entirely inside a single grain of rice.

```
se = strel('disk',15);
background =
```

```
se = strel('disk',15);
background = imopen(rice,se);
imshow(background);
```

Subtracting the background approximation image, background, from the original image, r ce, and view the resulting image. Post subtraction, the background is now uniform, however, the image has become a bit dark for analysis.

grains = rice -

Adjusting the contrast of the image.

```
grains = rice - background;
imshow(grains);
```

```
adjusted_grains = imadjust(grains);
imshow(adjusted_grains);
```

Creating a binary image from the processed image and removing the background noise.

```
bw =
imbinarize(adjusted_grains);
bw = bwareaopen(bw,50)
```

**Object Identification**

Find all the connected components (objects) in the binary image. The accuracy of

```
bw = imbinarize(adjusted_grains);
bw = bwareaopen(bw,50);
imshow(bw);
```

your results depends on the size of the objects, the connectivity parameter (4, 8, or arbitrary), and whether any objects are touching (in which case they could be labeled as one object). Some of the rice grains in the binary image `bw` are touching.

```
cc = bwconncomp(bw,4)
```

```
cc = bwconncomp(bw,4)
```

```
cc = struct with fields:
    Connectivity: 4
       ImageSize: [250 250]
      NumObjects: 92
    PixelIdxList: {1×92 cell}
```

```
cc.NumObjects
```

```
cc.NumObjects
```

```
ans = 92
```

View the rice grain that is labeled 50 in the image.

```
grain = false(size(bw));
grain(cc.PixelIdxList{50}) =
true; imshow(grain)
```

```
grain = false(size(bw));
grain(cc.PixelIdxList{50}) = true;
imshow(grain);
```

Visualize all the connected components in the image by creating a label matrix and then displaying it as a pseudo color indexed image. Use labelmatrix to create a label matrix from the output of bwconncomp. Note that labelmatrix stores the label matrix in the smallest numeric class necessary for the number of objects.

labeled =

Use label2rgb to choose the colormap, the background color, and how objects in

```
labeled = labelmatrix(cc);
whos labeled;
```

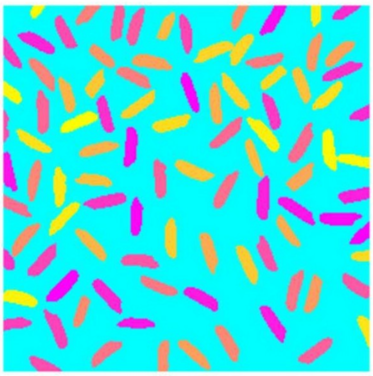
```
  Name           Size              Bytes  Class     Attributes

  labeled      250x250             62500  uint8
```

the label matrix map to colors in the colormap. In the pseudo color image, the label identifying each object in the label matrix maps to a different color in an associated colormap matrix.

RGB_label =

CONCLUSION:

```
RGB_label = label2rgb(labeled,'spring','c','shuffle');
imshow(RGB_label)
```



Object identification has been performed successfully.

**AMITY SCHOOL OF ENGINERRING & TECHNOLOGY**
**AMITY INSTITUTE OF TECHNOLOGY**

**(Academic Year 2022-23)**

# LAB 4
# Image Convolution

Student Name:_______________vaibhav mahmia________________

Class:_________B. Tech CSE__________Semester:________7_______

Enrolment Number:_____________A70405219061________________

**Faculty In-charge**
{Department of CSE}
ASET, AUM

| | |
|---|---|
| Name | Vaibhav mahmia |
| Enrolment Number | A70405219061 |
| Experiment Number | 4 |
| Batch | B |

## AIM OF THE EXPERIMENT:

Performing image convolution using MATLAB.

## THEORY:

A mathematical way of combining two signals to form a new signal is known as Convolution. In MATLAB for convolution ‘conv’ statement is used. The convolution of two vectors, p, and q given as “a = conv( p,q )” which represents that the area of overlap under the points as p slides across q. Convolution is the most important technique in Digital Signal Processing. The direct calculation of the convolution can be difficult so to calculate it easily Fourier transforms, and multiplication methods are used. Convolution is used in differential equations, statistics, image and signal processing, probability, language processing and so on.

## PROGRAM:

```
tom = imread("img\
tom.jpg");
subplot(2,2,1);
imshow(tom);
title("original");
kernel =
fspecial(“motion”,25,25); blur
= imfilter(tom,kernel);
subplot(2,2,2);
imshow(blur);
title("convoluted");


discord = imread("img\
discord.png"); discord =
im2double(discord); windowSize
= 15;
avg3 = ones(windowSize) /
windowSize^2; subplot(2,2,3);
imshow(discord)
```


CONCLUSION:

Image convolution has successfully been performed.

| | |
|---|---|
| Name | Vaibhav mahmia |
| Enrolment Number | A70405219061 |
| Experiment Number | 5 |
| Batch | B |

AIM OF THE EXPERIMENT:

Image enhancement using histogram equalization.

THEORY:

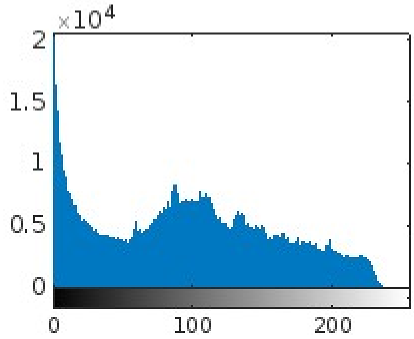
A histogram represents the intensity distribution of an image graphically. Therefore, it contains the quantified value of the number of pixels representing each intensity value. Accordingly, Histogram Equalization (HE) broadens the intensity range. Therefore, it maps one intensity distribution to another, thereby making intensity values evenly distributed. In other words, it spreads out the most frequent intensity values. As a result, it improves the contrast in the image.

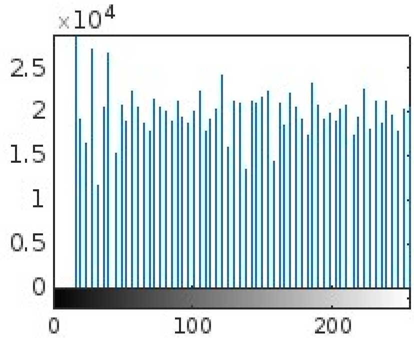
Since, histogram equalization is an image processing technique to improve the image contrast, it has many applications. In fact, we use it before further processing of an image. For instance, it is used widely in medical image processing. For instance, the visibility of many X-ray images are increased after histogram equalization. Because, the resulting image has better contrast. Besides X-rays, it is also used to enhance the quality of images from MRIs and CT-Scans.

Moreover, we can use histogram equalization as a pre-processing step in a Deep Learning application. For instance, we can use it in plant disease prediction. Because, it adds more visibility to the details of the image, the learning becomes faster. Additionally, many surveillance applications also require better contrast. Hence, we can use histogram equalization in a variety of image processing and deep learning applications.

PROGRAM:

```
I = imread("img.jpg");
J = histeq(I);
subplot(2,2,1);
imshow( I );
subplot(2,2,2);
imhist(I)
subplot(2,2,3);
imshow( J );
subplot(2,2,4);
imhist(J)
```




CONCLUSION:

Image enhancement using histogram equalization has successfully been performed.

AMITY SCHOOL OF ENGINEERING & TECHNOLOGY

AMITY INSTITUTE OF TECHNOLOGY

(Academic Year 2022-23)

# LAB 6

# Applying Fourier transformations on image

Student Name:  vaibhav mahmia

Class: CSE        Semester: VII

Enrolment Number: A70405219061

**Faculty In-charge**

{Department of ________}

ASET, AUM

# *DIP Lab Experiment 6*

| | |
|---|---|
| Name: | Vaibhav mahmia |
| Enrolment No: | A70405219061 |
| Course: | BTech CSE |
| Batch: | 2019-2023 |

## **Aim:** Applying Fourier transformations on image.

## Theory:

The Fourier transform is a mathematical formula that transforms a signal sampled in time or space to the same signal sampled in temporal or spatial frequency. In signal processing, the Fourier transform can reveal important characteristics of a signal, namely, its frequency components.

The Fourier transform is defined for a vector $x$ with $n$ uniformly sampled points by

$$y_{k+1} = \sum_{j=0}^{n-1} \omega^{jk} x_{j+1}.$$

$\omega = e^{-2\pi i/n}$ is one of the $n$ complex roots of unity where $i$ is the imaginary unit. For $x$ and $y$, the indices $j$ and $k$ range from $0$ to $n-1$.

The `fft` function in MATLAB® uses a fast Fourier transform algorithm to compute the Fourier transform of data. Consider a sinusoidal signal `x` that is a function of time `t` with frequency components of 15 Hz and 20 Hz. Use a time vector sampled in increments of 1/50 seconds over a period of 10 seconds.

## CODE:

```
I = imread("cameraman.tif"); % image of size 256x256
I = im2double(I);
F = fspecial("average",3); % average filter of size 3x3
Ipad = padarray(I,[3-1 3-1],0 ,"post"); % zero padding
Fpad = padarray(F,[256-1 256-1],0,"post"); % zero padding
Ifft = fft2(Ipad);
Ffft = fft2(Fpad);
Offt = Ifft.*Ffft;
Opad = ifft2(Offt);
O = Opad(2:end-1,2:end-1); % remove padding
subplot(1,2,1);
imshow(I); title('Before');
subplot(1,2,2);
```

```
imshow(O); title('After');
```

AMITY SCHOOL OF ENGINEERING & TECHNOLOGY
AMITY INSTITUTE OF TECHNOLOGY

(Academic Year 2022-23)

# LAB 7

# Huffman coding for Image Compression

Student Name: vaibhav mahmia

Class: CSE Semester: VII

Enrolment Number: A70405219061

**Faculty In-charge**
{Department of ________}
ASET, AUM

# *DIP Lab Experiment 7*

| | |
|---|---|
| Name: | Vaibhav mahmia |
| Enrolment No: | A70405219061 |
| Course: | BTech CSE |
| Batch: | 2019-2023 |

**Aim:** Using Huffman coding for image compression.

## Theory:

Huffman coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters. The most frequent character gets the smallest code and the least frequent character gets the largest code.

The variable-length codes assigned to input characters are Prefix Codes, means the codes (bit sequences) are assigned in such a way that the code assigned to one character is not the prefix of code assigned to any other character. This is how Huffman Coding makes sure that there is no ambiguity when decoding the generated bitstream.

Uses of Huffman encoding includes conjunction with cryptography and data compression. Huffman Coding is applied in compression algorithms like DEFLATE (used in PKZIP), JPEG, and MP3.

## Code:

```
%clearing all variableas and screen
clear all;
close all;
clc;
%Reading image
a=imread('flower.jpg');
figure,imshow(a)
%converting an image to grayscale
I=rgb2gray(a);
%size of the image
[m,n]=size(I);
Totalcount=m*n;
%variables using to find the probability
cnt=1;
sigma=0;
%computing the cumulative probability.
for i=0:255
k=I==i;
count(cnt)=sum(k(:))
%pro array is having the probabilities
pro(cnt)=count(cnt)/Totalcount;
sigma=sigma+pro(cnt);
cumpro(cnt)=sigma;
cnt=cnt+1;
end;
%Symbols for an image
symbols = [0:255];
%Huffman code Dictionary
dict = huffmandict(symbols,pro);
%function which converts array to vector
vec_size = 1;
for p = 1:m
for q = 1:n
newvec(vec_size) = I(p,q);
vec_size = vec_size+1;
end
end
%Huffman Encodig
hcode = huffmanenco(newvec,dict);
%Huffman Decoding
dhsig1 = huffmandeco(hcode,dict);
%convertign dhsig1 double to dhsig uint8
dhsig = uint8(dhsig1);
%vector to array conversion
dec_row=sqrt(length(dhsig));
dec_col=dec_row;
%variables using to convert vector 2 array
arr_row = 1;
arr_col = 1;
vec_si = 1;
for x = 1:m
for y = 1:n
back(x,y)=dhsig(vec_si);
arr_col = arr_col+1;
```

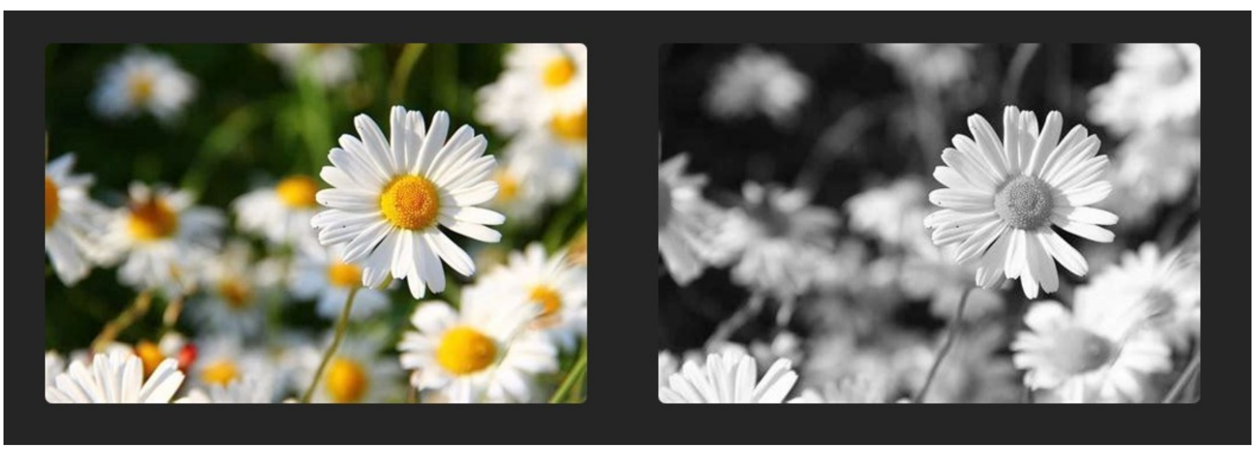
```
vec_si = vec_si + 1;
end
arr_row = arr_row+1;
end
%converting image from grayscale to rgb
[deco, map] = gray2ind(back,256);
RGB = ind2rgb(deco,map);
imwrite(RGB,'decoded.JPG');
```

| | | | |
|---|---|---|---|
| flower | 11/5/2022 9:52 AM | JPG File | 22 KB |

**Before**

| | | | |
|---|---|---|---|
| decoded | 11/13/2022 8:22 AM | JPG File | 19 KB |

**After**



**Before** **After**

| | |
|---|---|
| Name | Vaibhav mahmia |
| Enrolment Number | A70405219061 |
| Experiment Number | 8 |
| Batch | B |

AIM OF THE EXPERIMENT:

Image segmentation in MATLAB.

THEORY:

In the domain of digital image processing, sometimes we need to separate the main object from the image for clear observation. Image segmentation is the process that enables this partitioning. In this method, each pixel is assigned a label, and pixels that share some characteristics are assigned the same label number. This technique is widely used in the medical domain to locate the object of interest.

It is a technique to partition a digital image into multiple segments. This process is widely used in medical diagnosis. Here in this article, we have used morphological operations to segment the brain part from the MRI image. The segmentation is carried out in order to facilitate the analysis of the segmented images.

What is Medical imaging?

This term refers to the technique which medical professionals use to view inside the human body in order to diagnose, monitor, and treat. There are various techniques available in the modern scientific age.

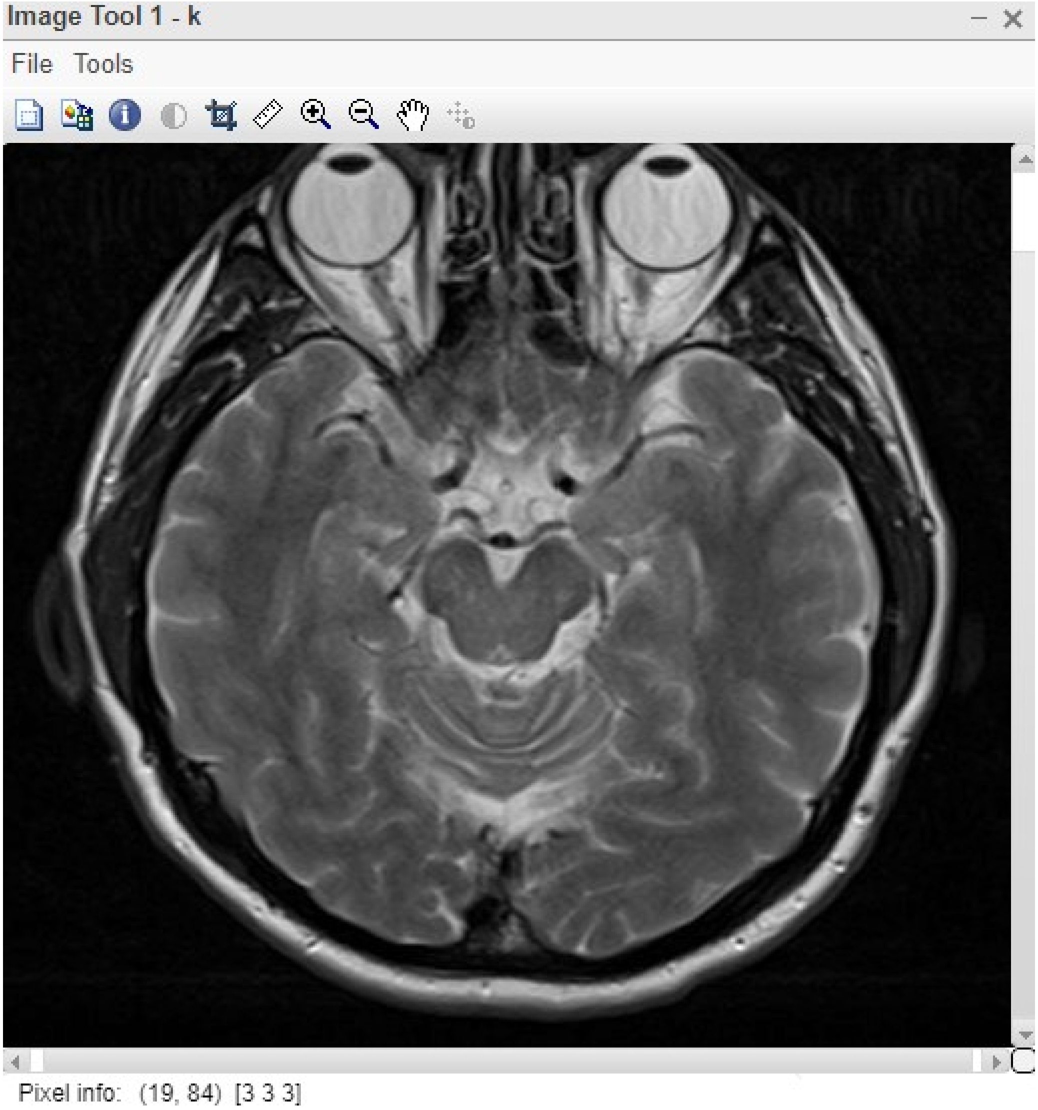
- MRI: Magnetic resonance imaging
- CT Scan: Computed Tomography Scan
- X-Ray: Using electromagnetic waves called X-rays.
- Ultrasound: Uses sound waves to create pictures of inner body tissues. It does not use any radiation.

PROGRAM:

```
% MATLAB code for
% Separate the brain part from MRI image.

% read the mri image.
k=imread("mri.jpg");

% display the image.
imtool(k,[]);

% convert it into binary image.
k1=im2bw(k,graythresh(k));

% display the binary image.
imtool(k1);
```

```
% Make the brain largest connected component.
% We need to apply opening operation.
% define the structuring element.
SE=strel('disk',7,4);

% apply the opening operation.
k2=imopen(k1,SE);

% display the image now.
imtool(k2);

% apply connected component analysis.
b=bwlabel(k2);

% display the colored map image.
imtool(b,[]);

% brain is component labeled as 9.
% set all other component as 0 except brain.
b(b~=9)=0;

% display the brain part.
imtool(b);

% inside the brain part, black portion is there.
% close the black pixels inside brain part.
k3=imclose(b,strel('disk',18));

% display the brain part.
imtool(k3);

% extract the brain from original image.
k4=k3.*double(k);

% display the real brain from original image.
imtool(k4,[]);
```

**Original Image:**




**Output extracted from the original image**


Image Tool 2 - k1
File Tools
Pixel info: (182, 54) 1

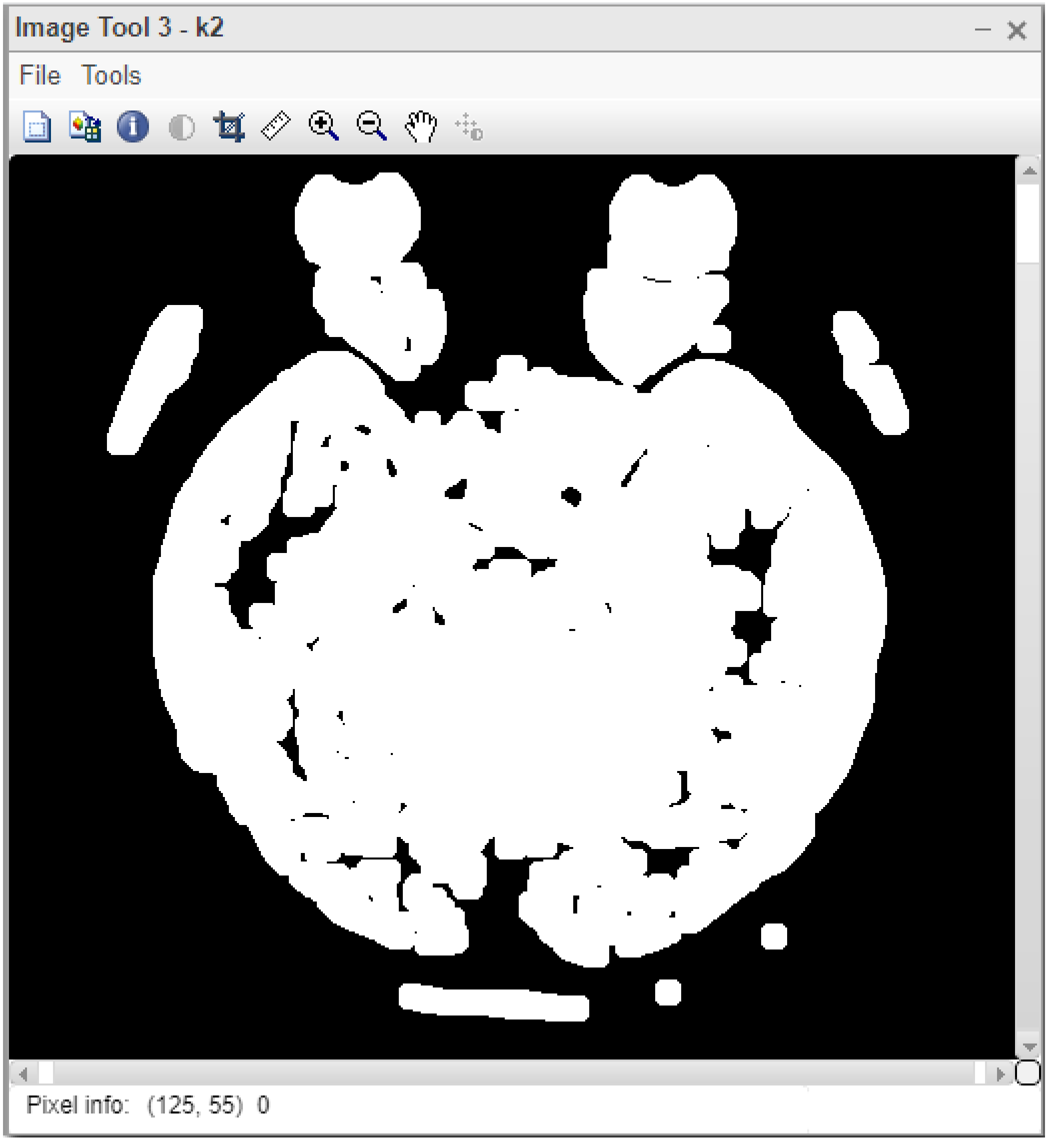
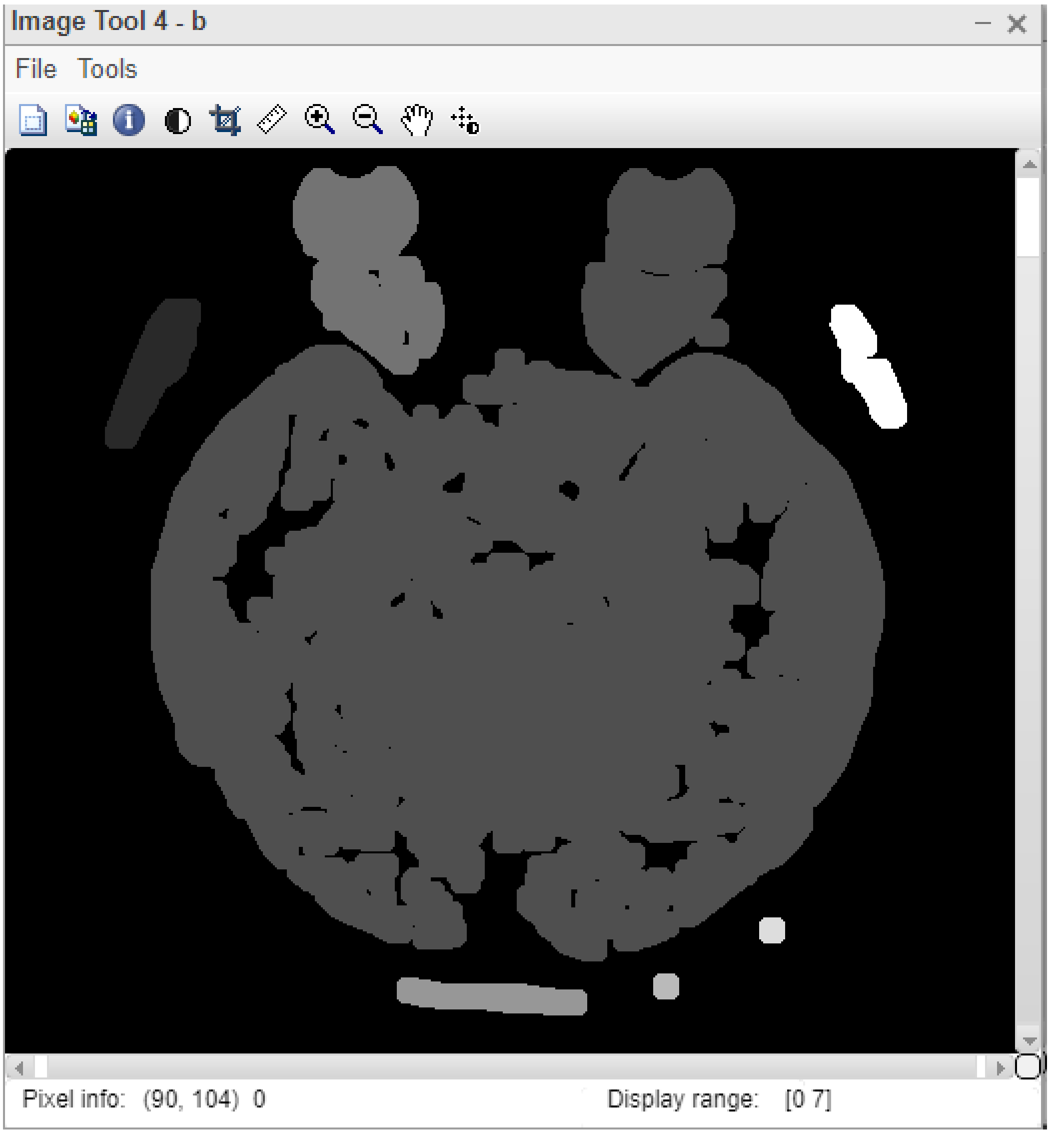


Image Tool 3 - k2
File Tools
Pixel info: (125, 55) 0




CONCLUSION:

Image Segmentation has successfully been performed.

| | |
|---|---|
| Name | Vaibhav mahmia |
| Enrolment Number | A70405219061 |
| Experiment Number | 9 |
| Batch | B |

**AIM OF THE EXPERIMENT:**

To perform Edge Detection in MATLAB.

THEORY:

In an image, an edge is a curve that follows a path of rapid change in image intensity. Edges are often associated with the boundaries of objects in a scene. Edge detection is used to identify the edges in an image.

To find edges, you can use the edge function. This function looks for places in the image where the intensity changes rapidly, using one of these two criteria:

- Places where the first derivative of the intensity is larger in magnitude than some threshold
- Places where the second derivative of the intensity has a zero crossing

edge provides several derivative estimators, each of which implements one of these definitions. For some of these estimators, you can specify whether the operation should be sensitive to horizontal edges, vertical edges, or both. edge returns a binary image containing 1's where edges are found and 0's elsewhere.

The most powerful edge-detection method that edge provides is the Canny method. The Canny method differs from the other edge-detection methods in that it uses two different thresholds (to detect strong and weak edges), and includes the weak edges in the output only if they are connected to strong edges.

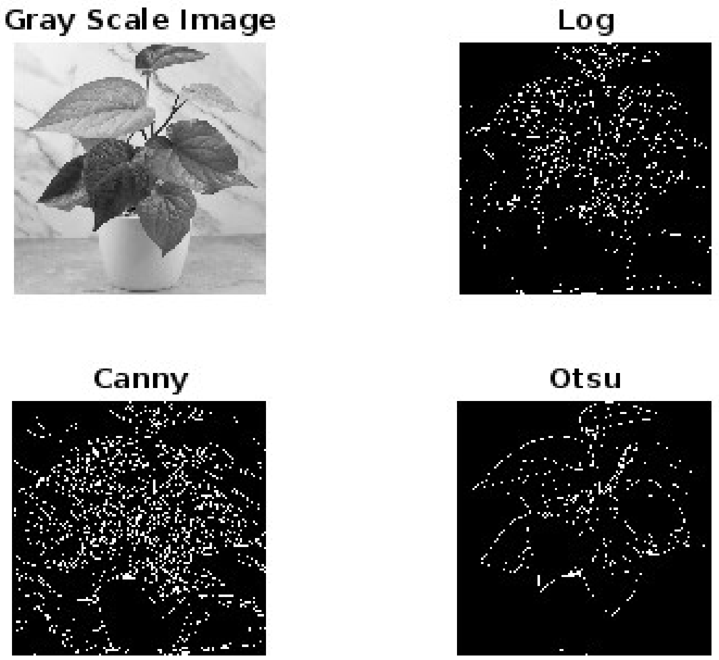
PROGRAM:

```
% importing the image
I = rgb2gray(imread("coin.jpg"));
subplot(2, 2, 1),
imshow(I);
title("Gray Scale Image");

% Log Edge Detection = Laplacian of Gaussian Filter = LoG
M = edge(I, 'log');
subplot(2, 2, 2),
imshow(M);
title("Log");

% Canny Edge Detection
N = edge(I, 'Canny');
subplot(2, 2, 3),
imshow(N);
title("Canny");

%% Edge detection using homogeneity operator
% Ref: A new homogeneity-based approach to edge detection using PSO
% Mahdi Setayesh, Mengjie Zhang and Mark Johnston
img=I;
```

```
[m,n]=size(img);
newimg=zeros(m,n);
for i=2:m-1
    for j=2:n-1
        newimg(i,j)=max([abs(img(i,j)-img(i-1,j-1)),...
            abs(img(i,j)-img(i,j-1)),...
            abs(img(i,j)-img(i-1,j)),...
            abs(img(i,j)-img(i+1,j+1)),...
            abs(img(i,j)-img(i+1,j)),...
            abs(img(i,j)-img(i,j+1)),...
            abs(img(i,j)-img(i+1,j-1)),...
            abs(img(i,j)-img(i-1,j+1))]);
    end
end
th=graythresh(img)*max(img(:)); % threshold calculation by otsu method
subplot(2,2,4);
imshow(newimg>th/4);
title('Otsu');
```

**Original Image:**

**After Edge Detection:**




CONCLUSION:

Image convolution has successfully been performed.