Name: Vighnesh Vikas Salgaonkar
Div/Rollno: B-21

PAGE No.
DATE / /

Experiment 1: Arithmetic & logical operations of on an image

Aim:- To perform arithmetic, logical and basic operations on an image.

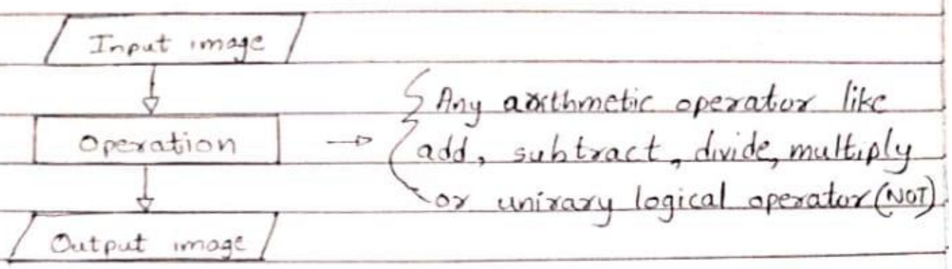
Apparatus:- PC/Laptop, MATLAB software and few images.

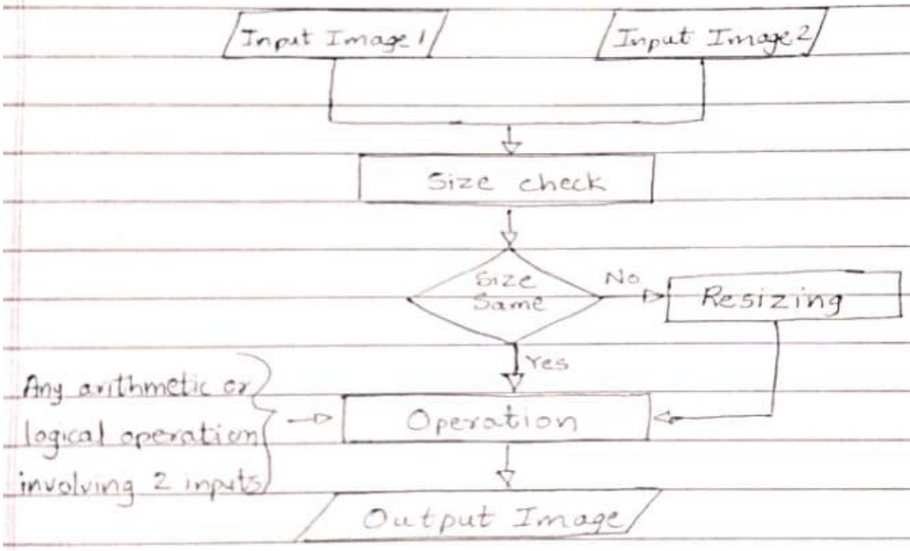
Theory:- MATLAB is a programming platform designed specifically for engineers and scientists to analyze and design systems and products. It is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. It allows matrix multiplication, matrix manipulation, plotting of functions and implementation of algorithms, creation of user interfaces and interfacing with programs written in another languages

Image arithmetic operations or a logical operator to two or more images; when applied produces some worthy results. The operators are applied in a pixel by pixel fashion, which means that the value of a pixel in the image depends only on the values of corresponding pixels in the input images. Hence the images must be of same size. Although image arithmetic is the simplest form of image processing, there is a wide range of applications. A main advantage of arithmetic operators or logical operators is that the process is very simple and therefore fast. Logical operators are often used to combine two (mostly binary) images. In case of integer images, the logical operator is normally applied in a bitwise way. To use all these operations on images, we should be familiar with few basic operations like format conversion, image resizing, and many more

Name: Vighnesh Vikas Salgaonkar
Div/Roll no: B-21

Algorithm & Flowchart:


Input image
Operation
Output image
Any arithmetic operator like add, subtract, divide, multiply or unirary logical operator (NOT)



Each time when we input an image in MATLAB we will use a variable to store the image matrix. If the operation can be performed on one image itself then we will use the pre existing functions in MATLAB and again store output in new variable. Once this is done we will use subplot to display the image. This process is same even if we have two or more images but we need to make sure that the images are of same size.

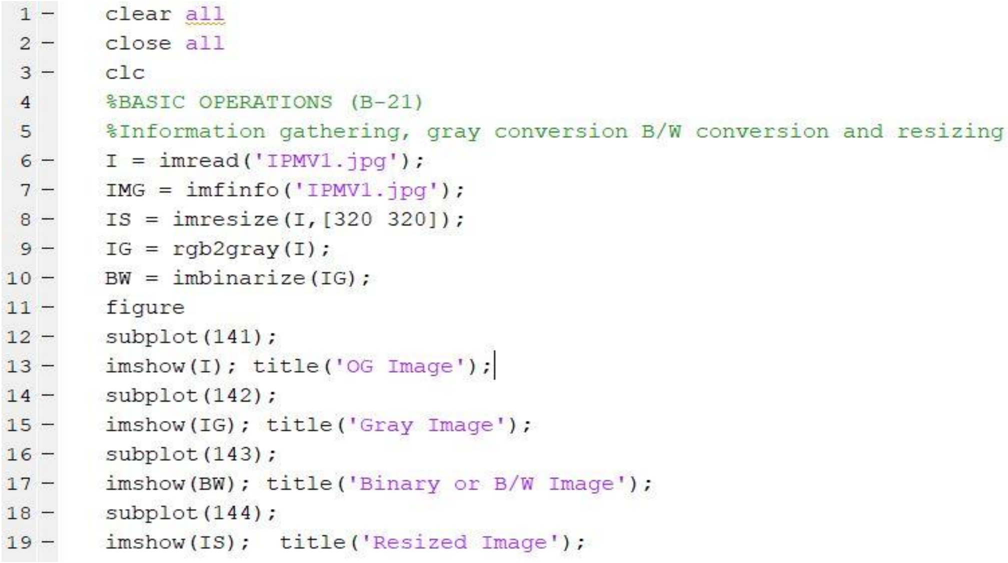
Operations:

- To perform basic operations on an image.
- To perform arithmetic operations on images.
- To perform logical operations on images.

Coding and Output:

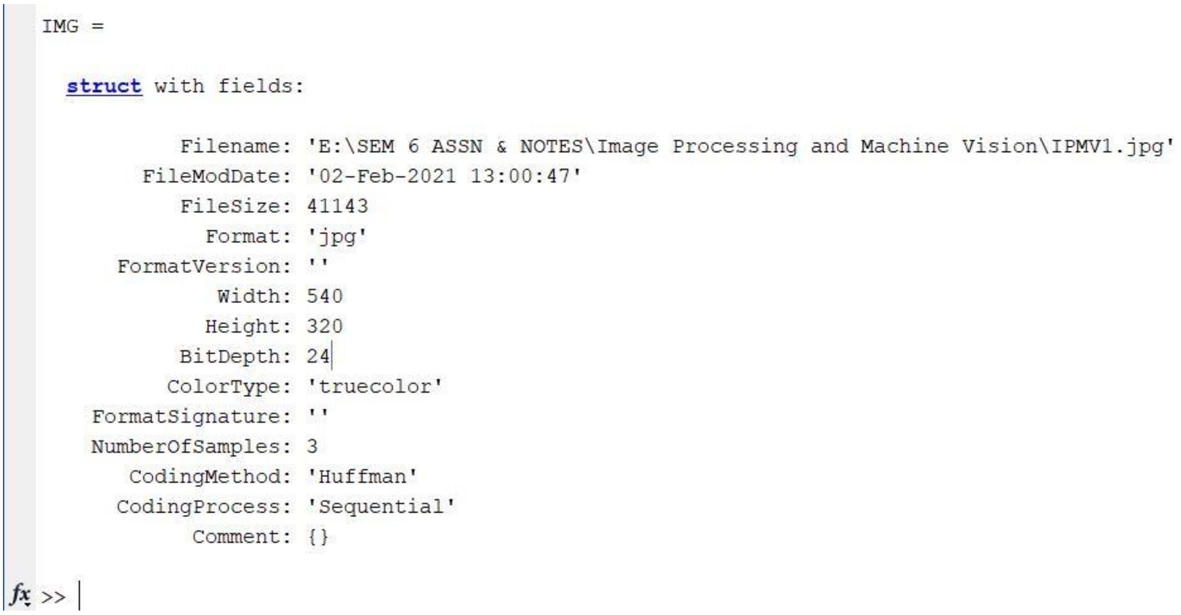
Arithmetic Operations:

- Code for Basic Operations

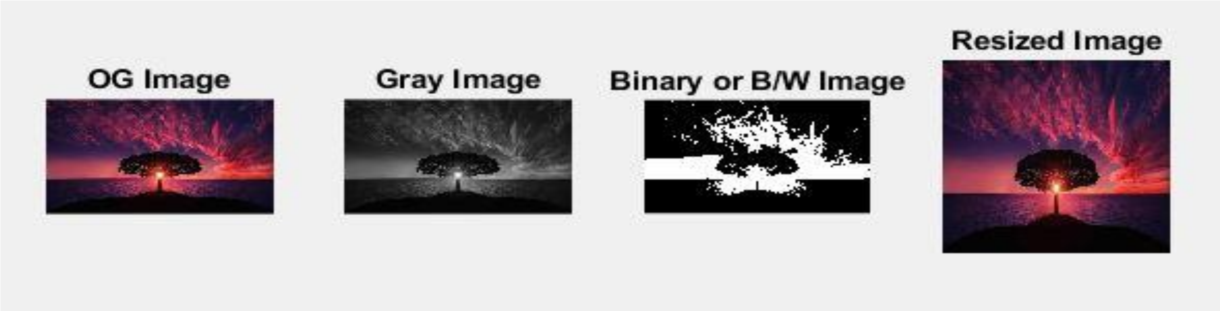


```
clear all
close all
clc
%BASIC OPERATIONS (B-21)
%Information gathering, gray conversion B/W conversion and resizing
I = imread('IPMV1.jpg');
IMG = imfinfo('IPMV1.jpg');
IS = imresize(I,[320 320]);
IG = rgb2gray(I);
BW = imbinarize(IG);
figure
subplot(141);
imshow(I); title('OG Image');
subplot(142);
imshow(IG); title('Gray Image');
subplot(143);
imshow(BW); title('Binary or B/W Image');
subplot(144);
imshow(IS);  title('Resized Image');
```

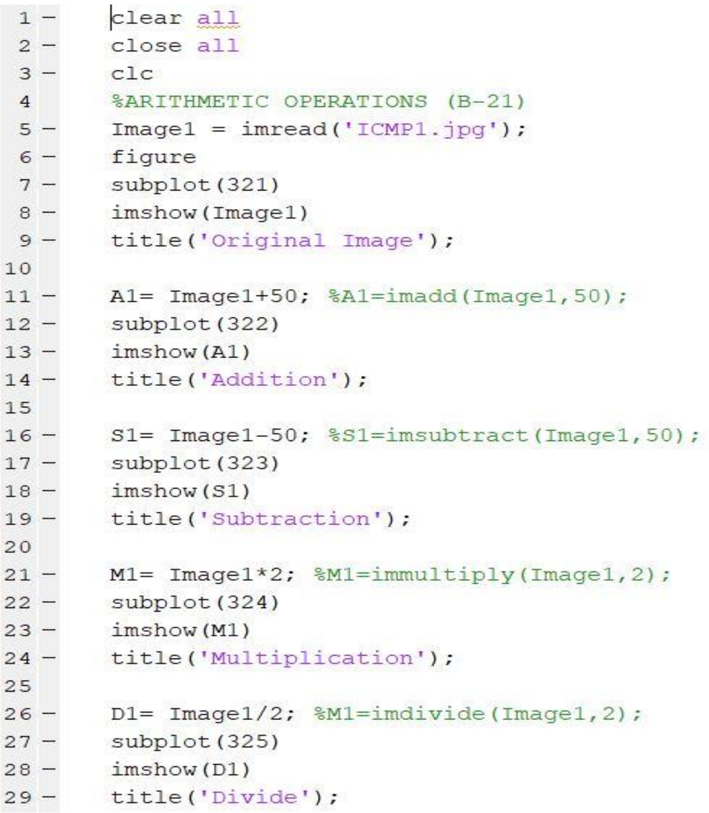
- Output for Basic Operations



```
IMG =

  struct with fields:

           Filename: 'E:\SEM 6 ASSN & NOTES\Image Processing and Machine Vision\IPMV1.jpg'
        FileModDate: '02-Feb-2021 13:00:47'
           FileSize: 41143
             Format: 'jpg'
      FormatVersion: ''
              Width: 540
             Height: 320
           BitDepth: 24
          ColorType: 'truecolor'
    FormatSignature: ''
    NumberOfSamples: 3
       CodingMethod: 'Huffman'
      CodingProcess: 'Sequential'
            Comment: {}

fx >>
```

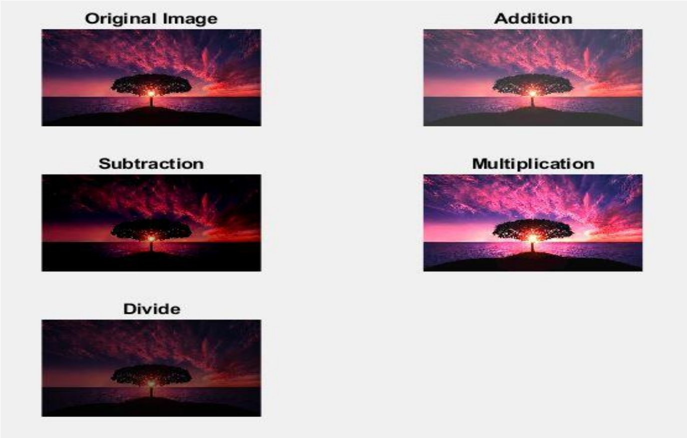


- Code for Arithmetic Operations



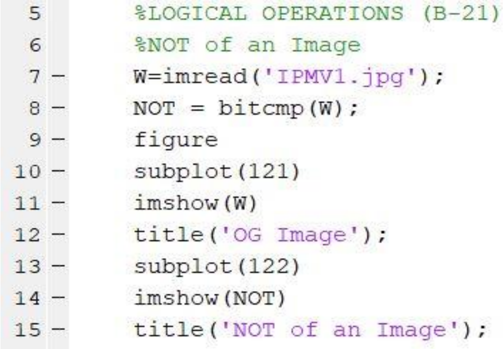
```
clear all
close all
clc
%ARITHMETIC OPERATIONS (B-21)
Image1 = imread('ICMP1.jpg');
figure
subplot(321)
imshow(Image1)
title('Original Image');

A1= Image1+50; %A1=imadd(Image1,50);
subplot(322)
imshow(A1)
title('Addition');

S1= Image1-50; %S1=imsubtract(Image1,50);
subplot(323)
imshow(S1)
title('Subtraction');

M1= Image1*2; %M1=immultiply(Image1,2);
subplot(324)
imshow(M1)
title('Multiplication');

D1= Image1/2; %M1=imdivide(Image1,2);
subplot(325)
imshow(D1)
title('Divide');
```

- Output for Arithmetic Operations



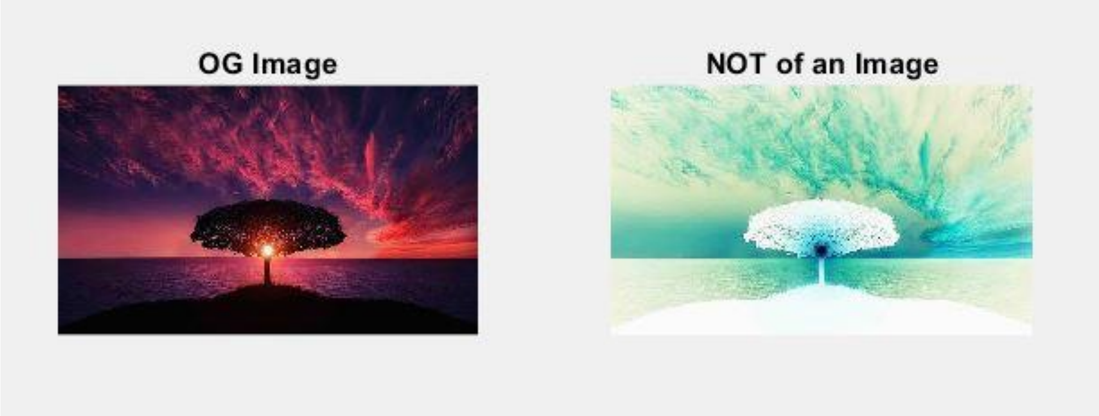
Logical Operations:

- Code for NOT:

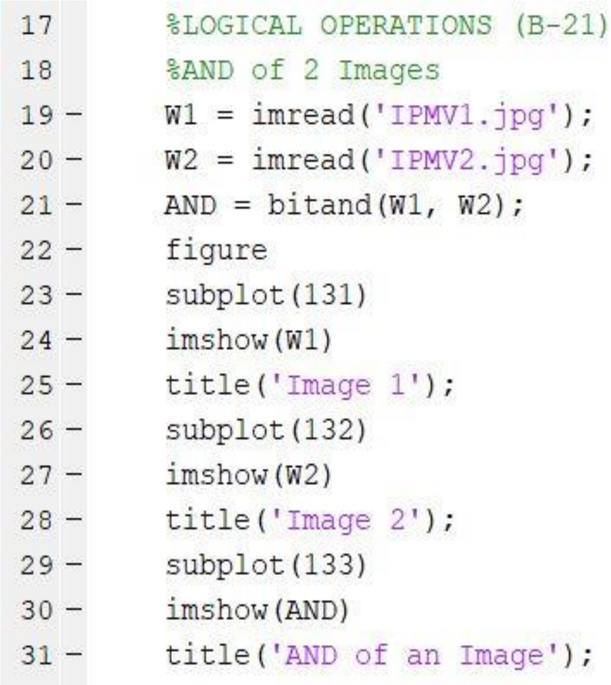


```
%LOGICAL OPERATIONS (B-21)
%NOT of an Image
W=imread('IPMV1.jpg');
NOT = bitcmp(W);
figure
subplot(121)
imshow(W)
title('OG Image');
subplot(122)
imshow(NOT)
title('NOT of an Image');
```

- Output for NOT:

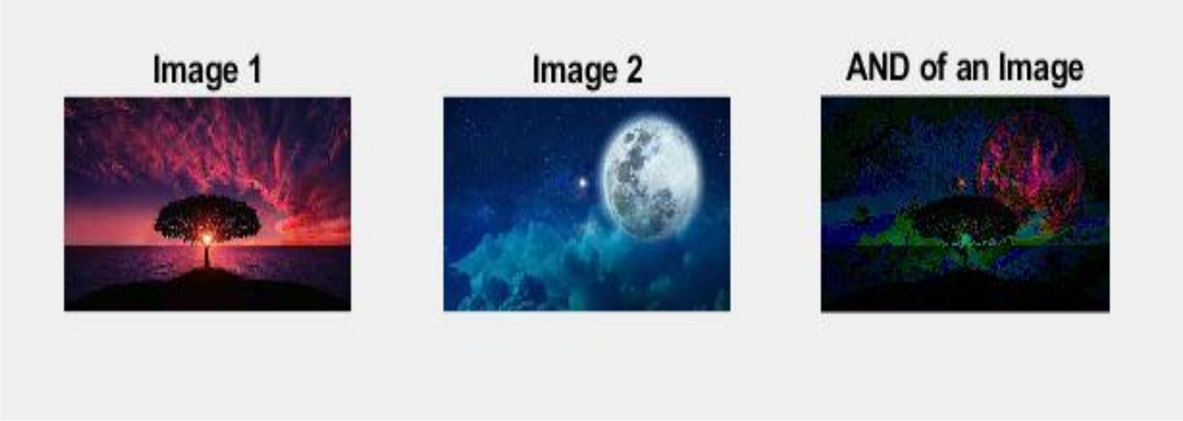


- Code for AND:

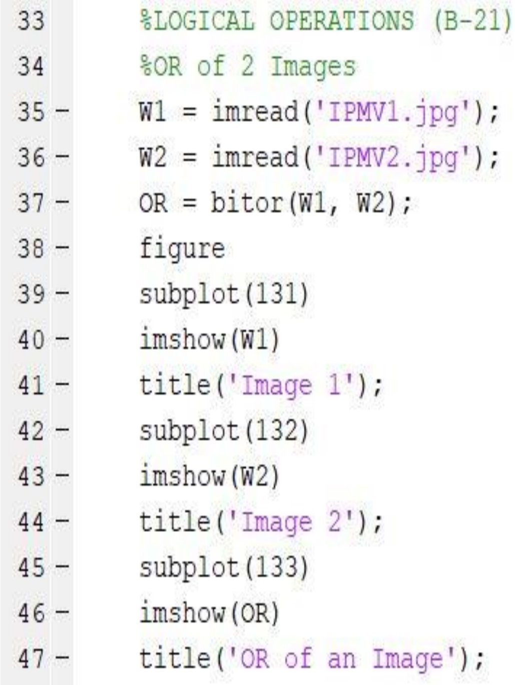


```
%LOGICAL OPERATIONS (B-21)
%AND of 2 Images
W1 = imread('IPMV1.jpg');
W2 = imread('IPMV2.jpg');
AND = bitand(W1, W2);
figure
subplot(131)
imshow(W1)
title('Image 1');
subplot(132)
imshow(W2)
title('Image 2');
subplot(133)
imshow(AND)
title('AND of an Image');
```

- Code for AND:

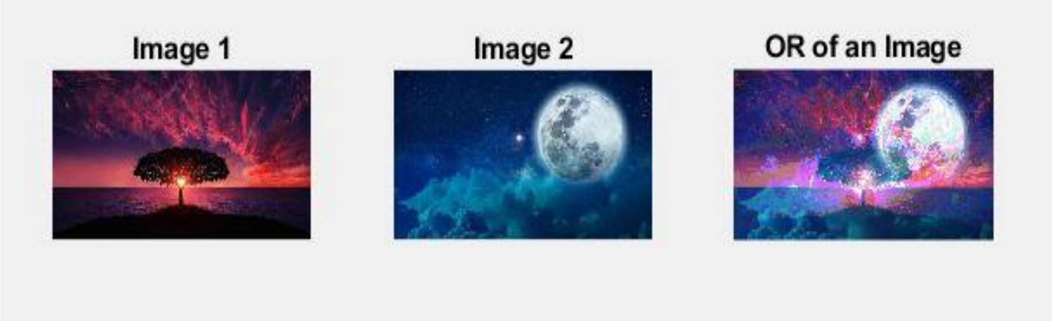


- Code for OR:

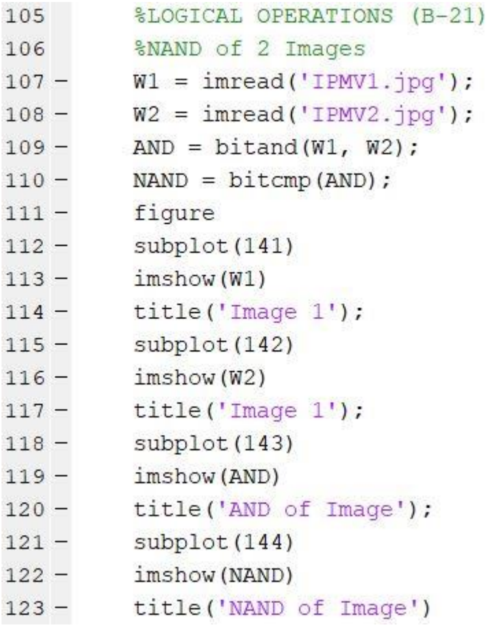


```
%LOGICAL OPERATIONS (B-21)
%OR of 2 Images
W1 = imread('IPMV1.jpg');
W2 = imread('IPMV2.jpg');
OR = bitor(W1, W2);
figure
subplot(131)
imshow(W1)
title('Image 1');
subplot(132)
imshow(W2)
title('Image 2');
subplot(133)
imshow(OR)
title('OR of an Image');
```

- Output for OR:

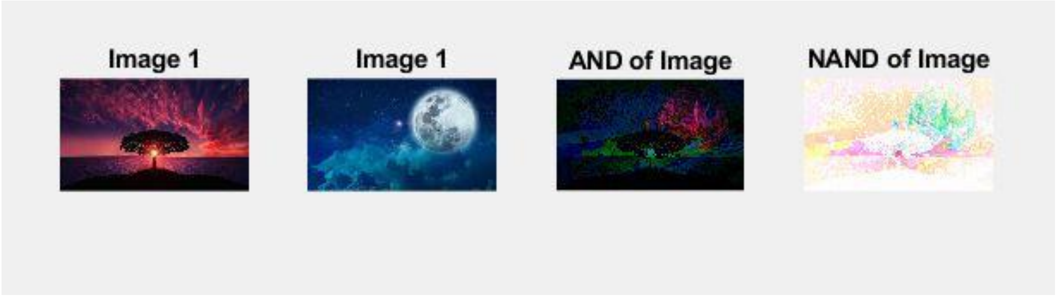


- Code for NAND:

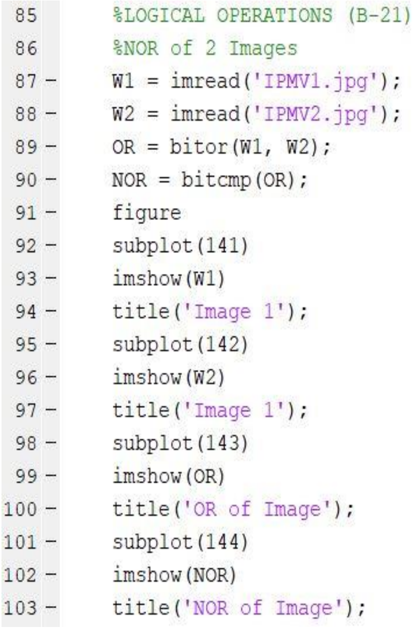


```
%LOGICAL OPERATIONS (B-21)
%NAND of 2 Images
W1 = imread('IPMV1.jpg');
W2 = imread('IPMV2.jpg');
AND = bitand(W1, W2);
NAND = bitcmp(AND);
figure
subplot(141)
imshow(W1)
title('Image 1');
subplot(142)
imshow(W2)
title('Image 1');
subplot(143)
imshow(AND)
title('AND of Image');
subplot(144)
imshow(NAND)
title('NAND of Image')
```

- Output for NAND:

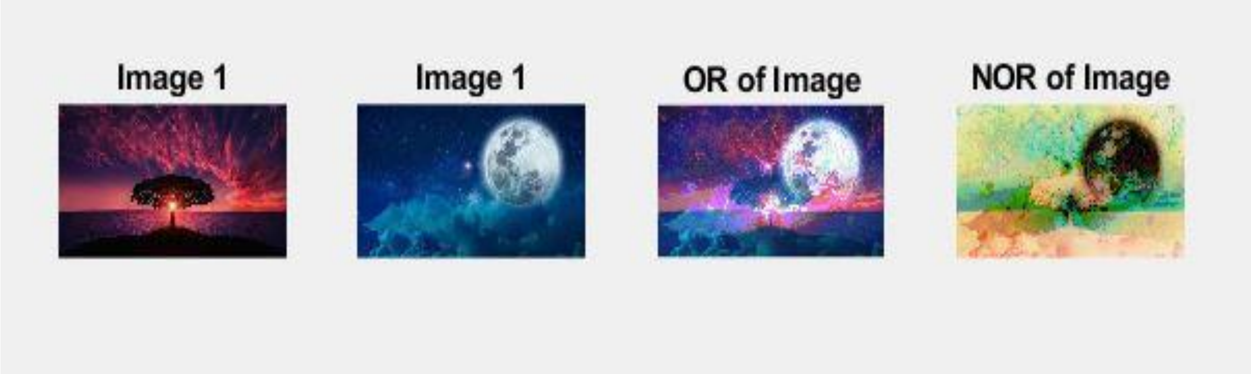


- Code for NOR:

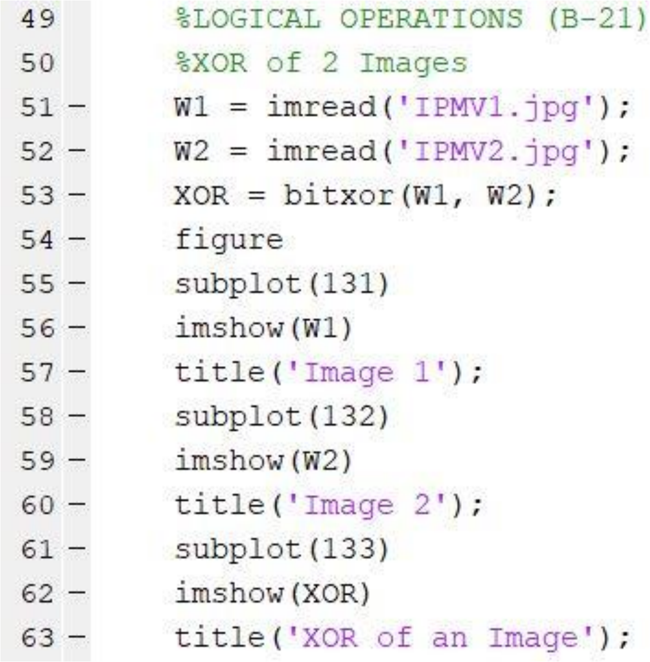


```
%LOGICAL OPERATIONS (B-21)
%NOR of 2 Images
W1 = imread('IPMV1.jpg');
W2 = imread('IPMV2.jpg');
OR = bitor(W1, W2);
NOR = bitcmp(OR);
figure
subplot(141)
imshow(W1)
title('Image 1');
subplot(142)
imshow(W2)
title('Image 1');
subplot(143)
imshow(OR)
title('OR of Image');
subplot(144)
imshow(NOR)
title('NOR of Image');
```

- Output for NOR:

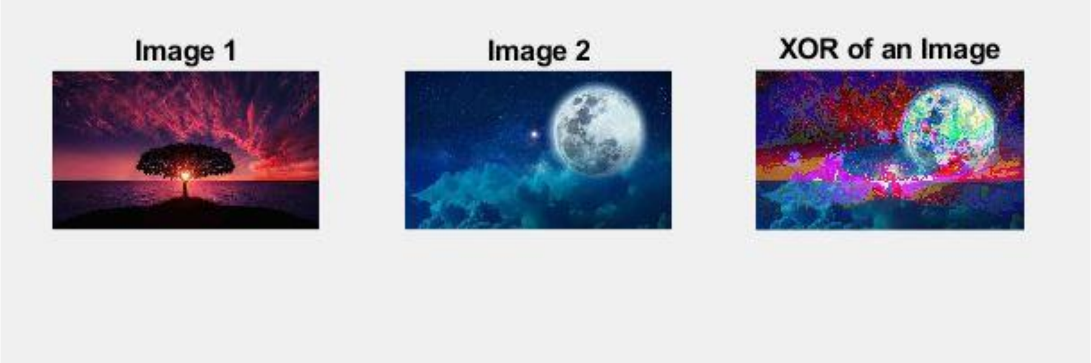


- Code for EXOR:

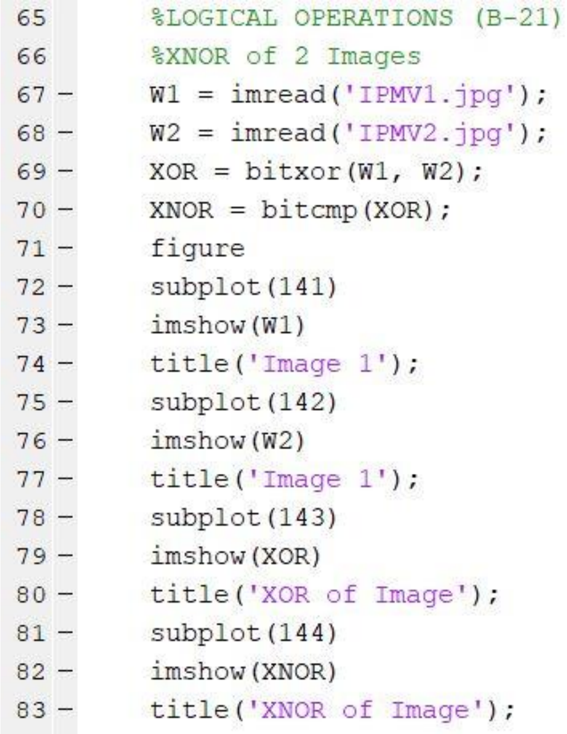


```
%LOGICAL OPERATIONS (B-21)
%XOR of 2 Images
W1 = imread('IPMV1.jpg');
W2 = imread('IPMV2.jpg');
XOR = bitxor(W1, W2);
figure
subplot(131)
imshow(W1)
title('Image 1');
subplot(132)
imshow(W2)
title('Image 2');
subplot(133)
imshow(XOR)
title('XOR of an Image');
```

- Output for EXOR:



- Code for EXNOR:



```
%LOGICAL OPERATIONS (B-21)
%XNOR of 2 Images
W1 = imread('IPMV1.jpg');
W2 = imread('IPMV2.jpg');
XOR = bitxor(W1, W2);
XNOR = bitcmp(XOR);
figure
subplot(141)
imshow(W1)
title('Image 1');
subplot(142)
imshow(W2)
title('Image 1');
subplot(143)
imshow(XOR)
title('XOR of Image');
subplot(144)
imshow(XNOR)
title('XNOR of Image');
```

- Output for EXNOR:

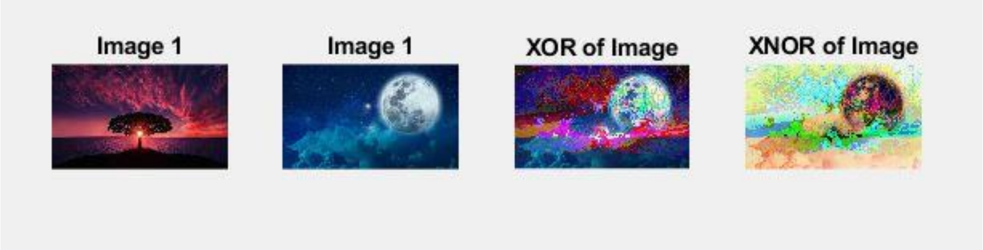
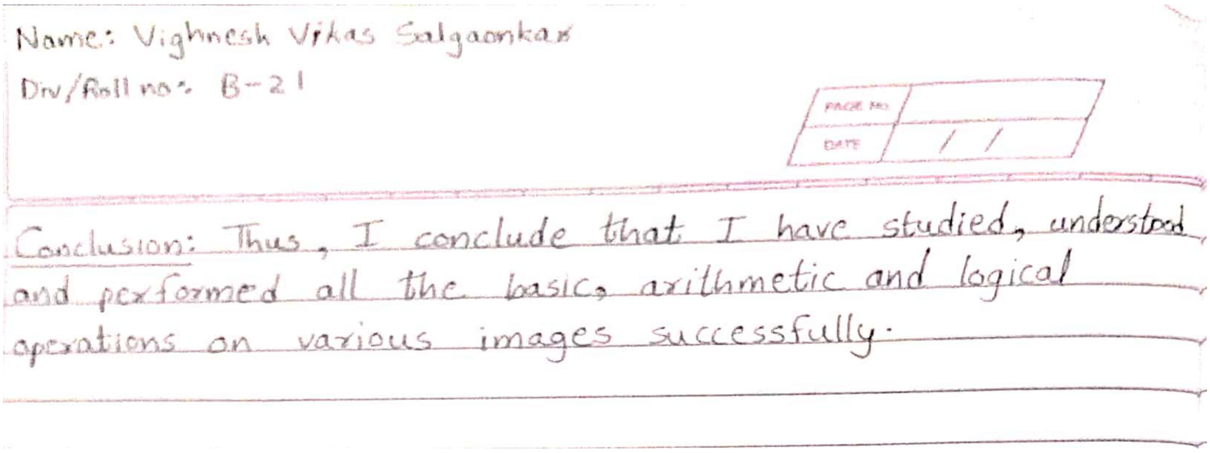


Image 1

Image 1

XOR of Image

XNOR of Image



Name: Vighnesh Vikas Salgaonkar
Div/Roll no: B-21

PAGE NO
DATE / /

Conclusion: Thus, I conclude that I have studied, understood and performed all the basics arithmetic and logical operations on various images successfully.