**Indian Institute Of Information Technology Surat**

**Kholvad, Kamrej, Surat-394190**



**LABORATORY MANUAL**

**Electronics and Communication Engineering Deapartment**

**EC-703 : Image Processing and Computer Vision**

**B.Tech VI-Semester VII**

NAME: DHYEY KIRITBHAI SAVALIYA

ROLL NO: UI21EC60

BRANCH: Electronics and Communication Engineering

INDIAN INSTITUTE OF INFORMATION AND TECHNOLOGY,SURAT

**Indian Institute Of Information Technology Surat**

Certificate



This is to certify that Mr. Dhyey Kiritbhai Savaliya of 4th year 7th sem Class Roll No. UI21EC60 has Satisfactory completed the course in Image Processing and Computer Vision laboratory practical during the Year 2024-2025 .

**Dr. Hemant Goklani Dr. Hemant Goklani**

**(Course Coordinator) Mr. Rahul Patel**

**(Lab Instructors)**

Index

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Date** | **Title** | **Sign** |
| 1. | 11/4/22 | Take a colour image and perform various conversions.  Construct a black image,white image, black in white,white in black image.  Plot 2d function f(x, y) = a \* sin(u(x) \* x + v(y) \* y) where u and v are any random. |  |
| 2. | 18/4/22 | Take a RGB image and perform following operations  1. Addition of two images  2. Subtraction of two images  3. Multiplication of two images  4. Division of Two images  5. Convert RGB to negative image  6. Change image into high brightness & low brightness  7. Change image into high contrast & low contrast |  |
| 3. | 23/5/22 | Take one particular colour image, convert it into a grayscale image and perform the following  transformation operation.  a) Log transform (for c=10,c=50,c=30)  b) Power law transform (c=1) (For dark image , lembda=0.6,0.2)(For brighter lembda=4,8).  c) Contrast stretching. |  |
| 4. | 9/5/22 | Take one particular colour image, convert it into a gray-scale image and perform the following transformation operations:  1-Gray level slicing (with and without background).  2-Bit Plane slicing. |  |
| 5. | 18/6/22 | Spatial Domain filtering based Restoration  A). Read this Image(ckt\_board.tif), generate the degraded image by adding the  Gaussian noise,Salt noise,pepper noisesalt & pepper noise to original ones.  B). Apply the following suitable spatial filter operations in order to restore the original ones:  Arithmetic mean ,Geometric mean ,Harmonic ,Contra-Harmonic ,Median ,Min ,Max ,Mid-point  Lab6.Simplified Spatial Filtering Techniques with Improved High-Pass Filter |  |
| 6. | 13/6/22 | Simplified Spatial Filtering Techniques with Improved High-Pass Filter |  |
| 7. | 13/6/22 | For the color image, convert to RGB to Gray and finding the edges using following operators:  1) Prewitt Operators.  2) Sobel Operators.  3) Use of smoothing filter / averaging filter with Sobel operator |  |
| 8. | 20/6/22 | For the color image, convert to RGB to Gray and finding the edges using following operators:  1) Use of thresholding with Sobel operator.  2) Canny operators.  3) Laplace of Gaussian (LOG) Marr-Hilderth transform.  Compare the results with Threshold gradient, LOG and canny operators. |  |
| 9. | 27/6/22 | Task 1: Laplacian Operator Filter  Task 2: Unsharp Masking  Task 3: Highboost Filtering |  |
| 10. | 7/4/22 | Consider an image and perform the following operations:  1. Discrete Fourier Transform (DFT) using Equation  2. Verify the implemented function of DFT with inbuilt function (Hint: np.fft.fft2, np.fft.fftshift or  cv2.dft(), cv2.idft())  3. Plot magnitude and Phase spectrum (Hint: magnitude : 20\*np.log(np.abs()), Phase: np.angle())  4. Apply shifting operation and observe the magnitude and phase spectrum  5. Apply rotation and observe the magnitude and phase spectrum (Comment on problem 4 and 5)  (Hint: np.rot90())  6. Consider two images. Find the magnitude and phase spectrum of both images. Reconstruct the  image using magnitude of first image and phase of second image and vice versa. |  |
| 11. | 7/4/22 | Take Grayscale or binary image and perform the following Morphological operations .  Produce some 5 X 5 kernels, Apply for the process of  1. Erosion  2. Dilation  3. Opening  4. Closing  Observe resultant images. |  |

**Lab – 1**

**Aim:**

Take a colour image and perform various conversions.

Construct a black image,white image, black in white,white in black image.

Plot 2d function f(x, y) = a \* sin(u(x) \* x + v(y) \* y) where u and v are any random.

**Code and Outputs:**

% Take a colour image and perform following conversion

% 1 RGB to Gray

% 2 RGB to Index

% 3 RGB to Binary

img = imread('peppers.png');

% RGB to Gray

function grayscale\_image = rgb\_to\_grayscale(rgb\_image)

r = rgb\_image(:, :, 1);

g = rgb\_image(:, :, 2);

b = rgb\_image(:, :, 3);

grayscale\_image = 0.2989 \* r + 0.5870 \* g + 0.1140 \* b;

end

grayscale\_image = rgb\_to\_grayscale(img);

%imshow(grayscale\_image);

% RGB to Index

[IND,map] = rgb2ind(img,32);

%figure

%imshow(IND)

%colormap(map)

% RGB to Binary

function binary\_image = rgb\_to\_binary(rgb\_image)

% Convert RGB to grayscale

r = rgb\_image(:, :, 1);

g = rgb\_image(:, :, 2);

b = rgb\_image(:, :, 3);

grayscale\_image = uint8(0.2989 \* r + 0.5870 \* g + 0.1140 \* b);

% Apply threshold

binary\_image = grayscale\_image > 100; % Convert to logical image

binary\_image = uint8(binary\_image \* 255); % Convert logical to uint8

end

binary\_img\_without\_function = rgb\_to\_binary(img);

%figure

%imshow(index\_img\_without\_function)

%index\_img\_with\_function = im2bw(img,100);

index\_img\_with\_function = imbinarize(img);

%figure

%imshow(index\_img\_without\_function)

% Display all images in a single figure

figure;

subplot(2, 2, 1);

imshow(img);

title('Original Image');

subplot(2, 2, 2);

imshow(grayscale\_image);

title('Grayscale Image');

subplot(2, 2, 3);

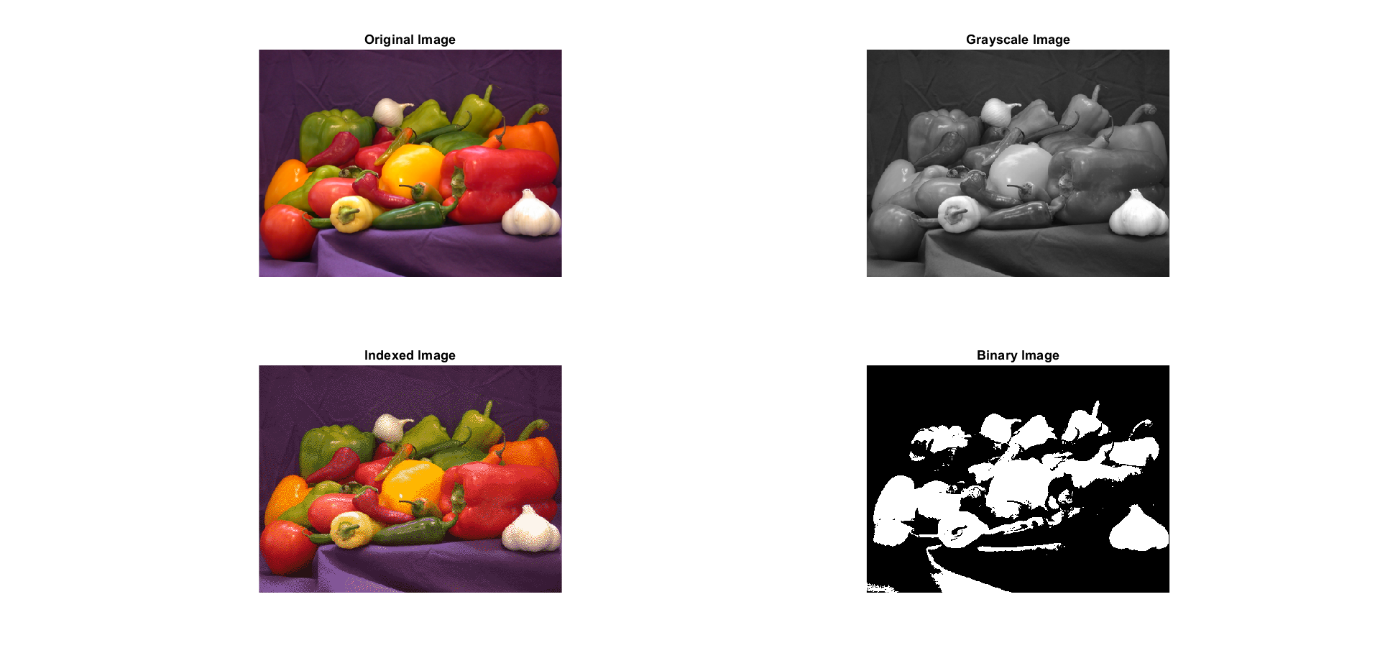
imshow(IND, map);

title('Indexed Image');

subplot(2, 2, 4);

imshow(binary\_img\_without\_function);

title('Binary Image');



% Construct Black Image , White Image , Black in White , White in Black

clc

clear all

close all

black\_Image = zeros(100, 100);

white\_Image = ones(100, 100);

white\_image\_small = ones(50, 50);

black\_image\_small = zeros(50, 50);

black\_in\_white\_image = zeros(100, 100);

black\_in\_white\_image(26:75, 26:75) = white\_image\_small;

white\_in\_black\_image = ones(100, 100);

white\_in\_black\_image(26:75, 26:75) = black\_image\_small;

% Plots

figure;

subplot(2, 2, 1);

imshow(black\_Image);

title('Black Image');

subplot(2, 2, 2);

imshow(white\_Image);

title('White Image');

subplot(2, 2, 3);

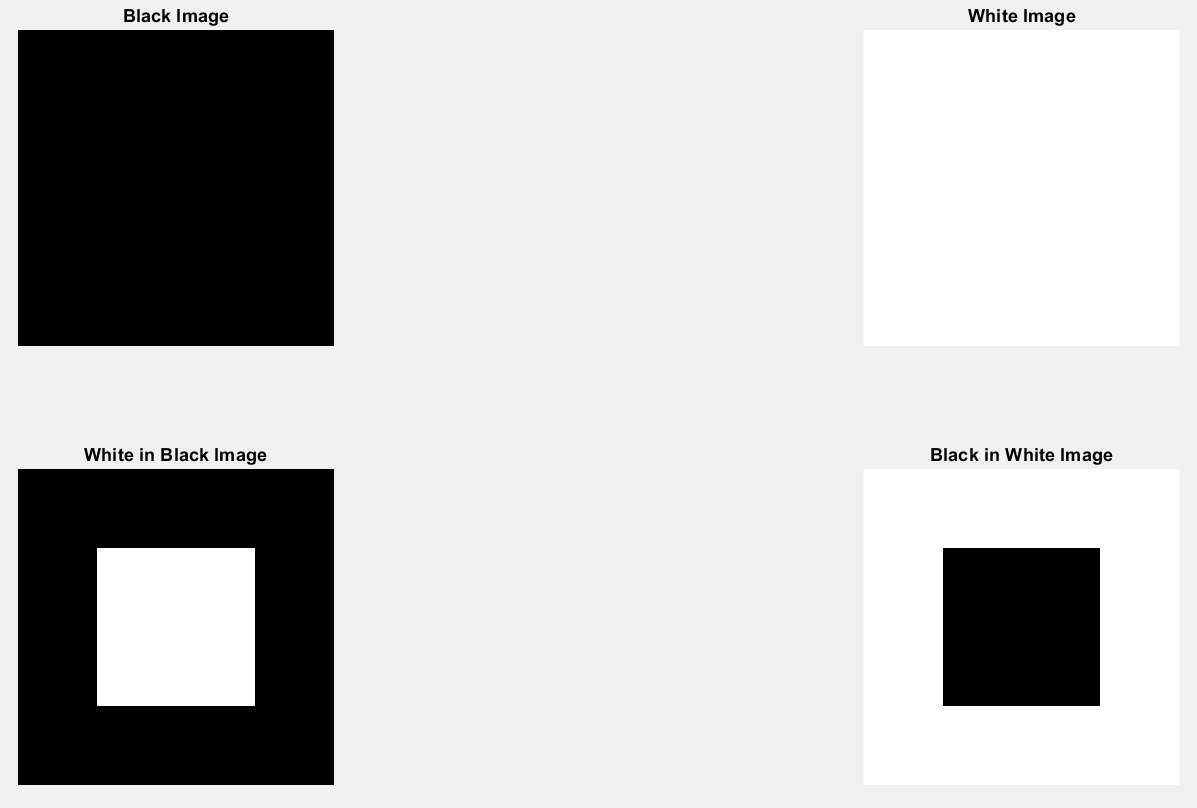
imshow(black\_in\_white\_image);

title('White in Black Image');

subplot(2, 2, 4);

imshow(white\_in\_black\_image);

title('Black in White Image');

****

% Define the range for x and y

x = linspace(-10, 10, 100);

y = linspace(-10, 10, 100);

[X, Y] = meshgrid(x, y);

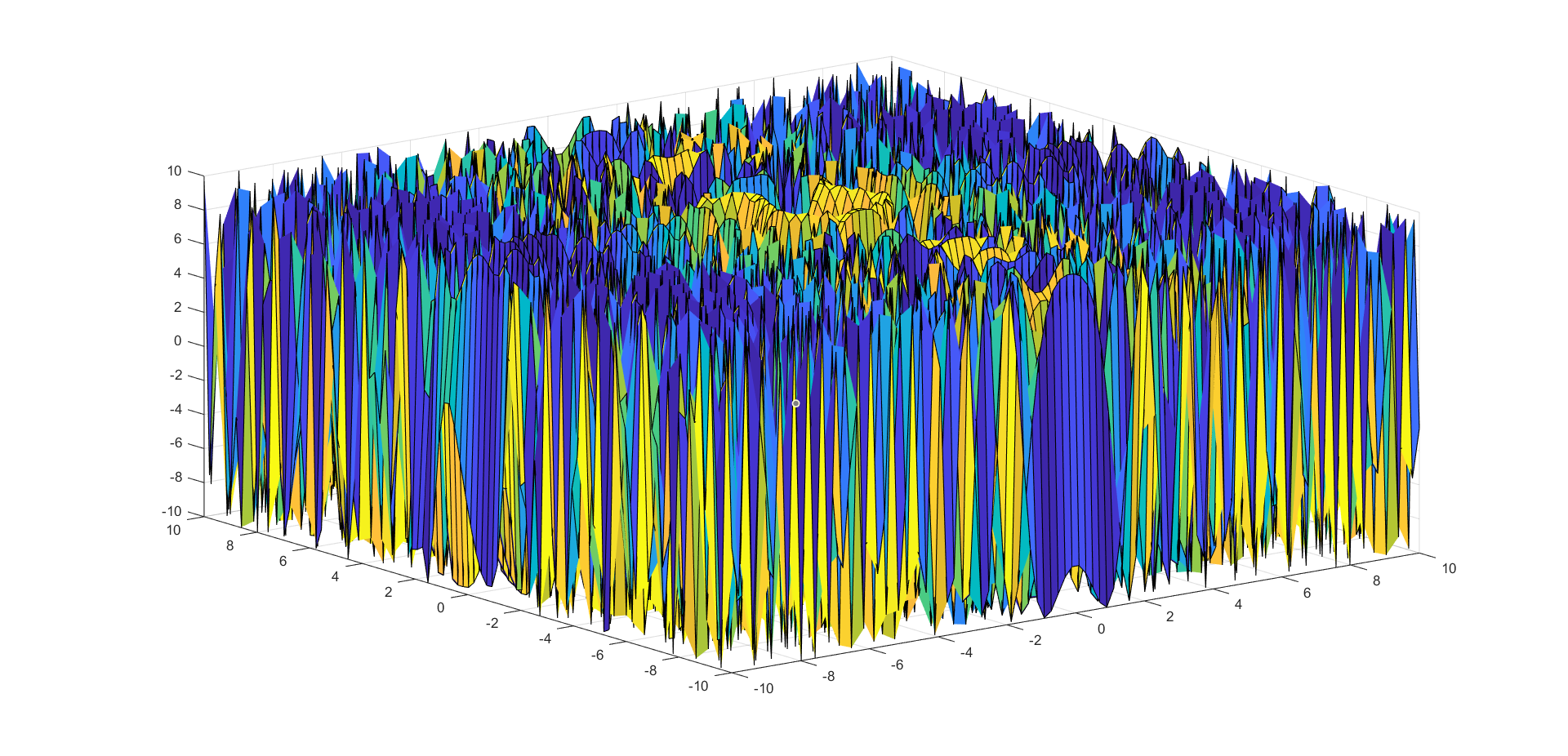
U = X;

V = cos(Y.^2);

F = 10 \* sin(U .\* X + V .\* Y);

plot(F)

surf(X, Y, F);



**Conclusion:**

**Lab – 2**

**Aim:**

**Code and Output:**

**Conclusion:**

**Lab – 2**

**Aim:**

**Code and Output:**

**Conclusion:**