**Aim: Use of Fourier transform for filtering the image.**

% Read an image

originalImage = imread('one.jpeg'); % Replace 'example.jpg' with your image file

% Convert the image to grayscale if it's in color

if size(originalImage, 3) == 3

originalImage = rgb2gray(originalImage);

end

% Display the original image

subplot(2, 3, 1);

imshow(originalImage);

title('Original Image');

% Compute the 2D Fourier Transform of the image

fourierTransform = fft2(double(originalImage));

% Shift the zero frequency components to the center

fourierTransformShifted = fftshift(fourierTransform);

% Create a simple low-pass filter (e.g., a circular mask)

[M, N] = size(originalImage);

radius = 50; % Adjust the radius to control the filter size

[x, y] = meshgrid(1:N, 1:M);

centerX = N/2;

centerY = M/2;

mask = (x - centerX).^2 + (y - centerY).^2 <= radius^2;

% Apply the filter by multiplying it with the Fourier Transform

filteredTransform = fourierTransformShifted .\* mask;

% Shift the frequency components back to their original positions

filteredTransformShifted = ifftshift(filteredTransform);

% Compute the inverse Fourier Transform to get the filtered image

filteredImage = ifft2(filteredTransformShifted);

% Display the filtered image

subplot(2, 3, 2);

imshow(abs(filteredImage), []);

title('Filtered Image');

% Display the magnitude of the Fourier Transform

subplot(2, 3, 3);

imshow(log(1 + abs(fourierTransformShifted)), []);

title('Magnitude of Fourier Transform');

% Display the filter mask

subplot(2, 3, 4);

imshow(mask, []);

title('Filter Mask');

% Display the phase of the Fourier Transform

subplot(2, 3, 5);

imshow(angle(fourierTransformShifted), []);

title('Phase of Fourier Transform');

% Display the phase of the filtered image

subplot(2, 3, 6);

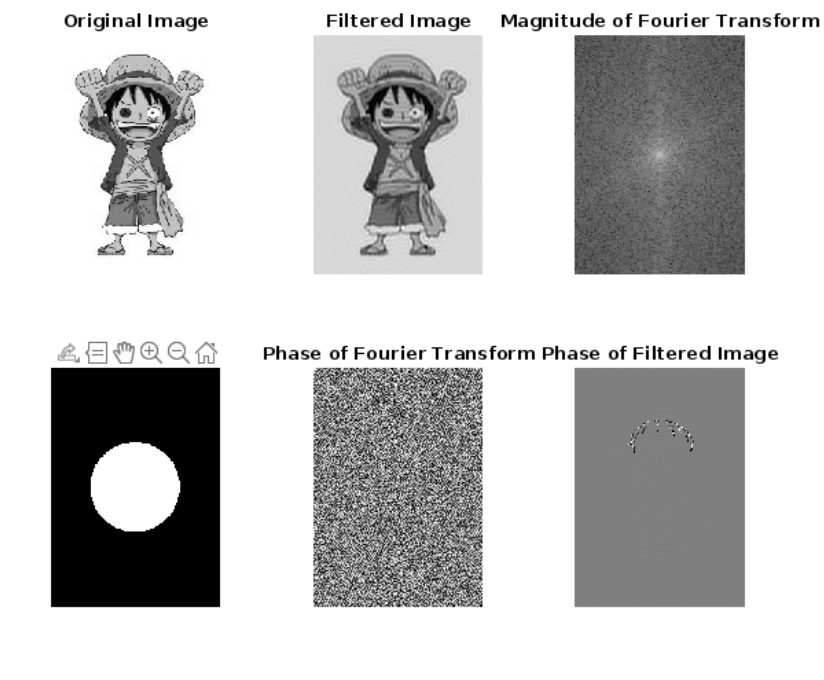
imshow(angle(filteredImage), []);

title('Phase of Filtered Image');

% Wait for user input before closing the figure

pause;

close;

****