1. Create a program to demonstrate Geometric transformations- Image rotation, scaling, and translation.
2. Display of FFT (1-D & 2-D) of an image and apply Two-dimensional Fourier transform to represent the content of an image using the discrete Fourier transform (DFT) and masking with DFT.
3. Write a Program of Contrast stretching of a low contrast image, Histogram, and Histogram Equalization and Display of bit planes of an Image.
4. Computation of Mean, Standard Deviation, Correlation coefficient of the given Image
5. Implementation of Image Smoothening Filters (Mean and Median filtering of an Image)
6. Implementation of image sharpening filters and Edge Detection using Gradient Filters.
7. Implementation of Image Compression by DCT, DPCM, HUFFMAN coding
8. Implementation of image restoring techniques.
9. Implementation of Image Intensity slicing technique for image enhancement.
10. Study and implement Canny edge detection Algorithm to images and compare it with the existing edge detection algorithms.

Exp 1:

```
%Geometric Transformations - Image Rotation, Scaling, and Translation

% Load an example image
originalImage = imread('example_image.jpg');

% Display the original image
figure;
subplot(2, 2, 1);
imshow(originalImage);
title('Original Image');

% 1. Image Rotation
angle = 30; % Rotation angle in degrees
```

```
rotatedImage = imrotate(originalImage, angle, 'bilinear', 'crop');

% Display the rotated image
subplot(2, 2, 2);
imshow(rotatedImage);
title('Rotated Image');

% 2. Image Scaling
scaleFactor = 1.5; % Scaling factor
scaledImage = imresize(originalImage, scaleFactor);

% Display the scaled image
subplot(2, 2, 3);
imshow(scaledImage);
title('Scaled Image');

% 3. Image Translation
tx = 50; % Translation along x-axis
ty = 20; % Translation along y-axis
translatedImage = imtranslate(originalImage, [tx, ty]);

% Display the translated image
subplot(2, 2, 4);
imshow(translatedImage);
title('Translated Image');

% Adjust the layout for better visualization
sgtitle('Geometric Transformations');

% Save the figure (optional)
% saveas(gcf, 'geometric_transformations_example.png');
```

Exp 2

```
% Read an image
originalImage = imread('your_image.jpg'); % replace 'your_image.jpg' with the path to your image
originalImage = rgb2gray(originalImage); % Convert to grayscale if the image is RGB

% Display the original image
figure;
subplot(2, 2, 1);
imshow(originalImage);
title('Original Image');

% 2D Fourier Transform
fftImage = fft2(double(originalImage));

% Display the magnitude spectrum of the Fourier Transform
magnitudeSpectrum = log(1 + abs(fftshift(fftImage)));
subplot(2, 2, 2);
imshow(magnitudeSpectrum, []);
title('Magnitude Spectrum');

% Apply a simple mask (e.g., a high-pass filter)
mask = ones(size(fftImage));
maskSize = 30;
mask(end/2-maskSize:end/2+maskSize, end/2-maskSize:end/2+maskSize) = 0;
filteredImage = fftImage .* mask;

% Display the masked magnitude spectrum
filteredMagnitude = log(1 + abs(fftshift(filteredImage)));
subplot(2, 2, 3);
```

Exp 3

```
% Image Enhancement
I=imread('cancercell.jpg');
subplot(4,2,1); imshow(I); title('Original Image');

g=rgb2gray(I);
subplot(4,2,5); imshow(g); title('Gray Image');

J=imadjust(g,[0.3 0.7],[]);
subplot(4,2,3); imshow(J); title('Enhanced Image');

D= imadjust(I,[0.2 0.3 0; 0.6 0.7 1],[]);
subplot(4,2,4);imshow(D);title('Enhanced Image 2');
% Histogram and Histogram Equalization
subplot(4,2,7); imhist(g); title('Histogram of Gray Image');
m=histeq(g);
subplot(4,2,6); imshow(m); title('Equalized Image');
subplot(4,2,8); imhist(m); title('Histogram of Equalized Image');
```

Exp4

```
i=imread('1.jfif');
subplot(2,2,1); imshow(i);title('Original Image');

g=rgb2gray(i);
subplot(2,2,2); imshow(g);title('Gray Image');
c = imcrop(g,[60 40 100 90]);
subplot(2,2,3); imshow(c);title('Cropped Image');
m=mean2(i);disp('m'); disp(m);
s=std2(i); disp('s'); disp(s);
```

```
figure,
k=(checkerboard>0.8);
subplot(2,1,1); imshow(k); title('Image1');

k1=(checkerboard>0.5);
subplot(2,1,2); imshow(k1); title('Image2');
r=corr2(k,k1);
disp('r');disp(r);
```

Exp 5

```
I=imread('nuron.jpg');
 K = rgb2gray(I);
 J= imnoise(K ,'salt & pepper',0.05);
 f= medfilt2(J,[3,3]);
 f1=medfilt2(J,[10,10]);

 subplot(3,2,1); imshow(I); title('Original Image');  subplot(3,2,2); imshow(K); title('Gray Image');  subplot(3,2,3); imshow(J); title('Noise added Image');  subplot(3,2,4); imshow(f); title('3x3 Image');  subplot(3,2,5); imshow(f1); title('10x10 Image');


 %Mean Filter and Average Filter  figure;  i=imread('nuron.jpg');  g=rgb2gray(i);


g1=fspecial('average',[3 3]);  b1 = imfilter(g,g1);  subplot(2,2,1); imshow(i); title('Original Image');  subplot(2,2,2); imshow(g); title('Gray Image');  subplot(2,2,3); imshow(b1); title('3x3 Image');
g2= fspecial('average',[10 10]);  b2=imfilter(g,g2);
 subplot(2,2,4); imshow(b2); title('10x10 Image');

%Implementation of filter using Convolution figure;
I= imread('earcell.jpg');
I=I(:,:,1); subplot(2,2,1); imshow(I); title('Original Image');

a=[0.001 0.001 0.001; 0.001 0.001 0.001; 0.001 0.001 0.001]; R=conv2(a,I);
subplot(2,2,2); imshow(R); title('Filtered Image');

b=[0.005 0.005 0.005; 0.005 0.005 0.005; 0.005 0.005 0.005]; R1=conv2(b,I);
```

```
subplot(2,2,3); imshow(R1); title('Filtered Image 2');
```

Exp 6

```
i=imread('cancercell.jpg');
subplot(4,2,1); imshow(i); title('Original Image');

g=rgb2gray(i);
subplot(4,2,2); imshow(g); title('Gray Image');

f=fspecial('laplacian',0.05); im=imfilter(g,f);
subplot(4,2,3); imshow(im); title('Laplacian ');

s=edge(g, 'sobel');
subplot(4,2,4); imshow(s); title('Sobel');

p=edge(g, 'prewitt');
subplot(4,2,5); imshow(p); title('Prewitt');

r=edge(g, 'roberts');
subplot(4,2,6); imshow(r); title('Roberts');

[BW,thresh,gv,gh]=edge(g,'sobel',[],'horizontal');
[BW1,thresh1,gv1,gh1]=edge(g,'sobel',[],'vertical');

subplot(4,2,7); imshow(BW); title('Sobel Horizontal'); subplot(4,2,8);
imshow(BW); title('Sobel Vertical');
```

Exp 7

```
I = imread("download.jpg","jpg");
g=rgb2gray(I);

blockSize = 8;
```

```
dctCoefficients = blockproc(g, [blockSize blockSize], @(block) dct2(block.data));

quantizationFactor = 4;
quantizedDCT = round(dctCoefficients / quantizationFactor);

symbols = unique(quantizedDCT(:));
probabilities = histcounts(quantizedDCT(:), numel(symbols)) / numel(quantizedDCT);
huffmanDictionary = huffmandict(symbols, probabilities);

dctHuffmanCodes = huffmanenco(quantizedDCT(:), huffmanDictionary);

decodedQuantizedDCT = huffmandeco(dctHuffmanCodes, huffmanDictionary);
decodedQuantizedDCT = reshape(decodedQuantizedDCT, size(quantizedDCT));

reconstructedImage = blockproc(decodedQuantizedDCT, [blockSize blockSize], @(block) idct2(block.data));
re2 = reconstructedImage * quantizationFactor;

figure;
subplot(3, 1, 1);
imshow(g);
title('Original Image');
subplot(3, 1, 2);
imshow(uint8(reconstructedImage));
title('Reconstructed Image');
subplot(3, 1, 3);
imshow(uint8(re2));
title('Reconstructed Image multiplied by quantization factor');
```

Exp 8

```
% Read input image
inputImage = imread('input_image.jpg');

% Add noise to the image (for demonstration purposes)
noisyImage = imnoise(inputImage, 'salt & pepper', 0.02);
```

```
% Apply median filtering for noise reduction
filteredImage = medfilt2(noisyImage);

% Display original, noisy, and filtered images
figure;
subplot(2, 3, 1), imshow(inputImage), title('Original Image');
subplot(2, 3, 2), imshow(noisyImage), title('Noisy Image');
subplot(2, 3, 3), imshow(filteredImage), title('Filtered Image');

% Read blurred image
blurredImage = imread('blurred_image.jpg');

% Define the point spread function (PSF)
PSF = fspecial('motion', 20, 45); % Example motion blur PSF

% Apply Wiener deconvolution
deblurredImage = deconvwnr(blurredImage, PSF);

% Display original, blurred, and deblurred images
subplot(2, 3, 4), imshow(inputImage), title('Original Image');
subplot(2, 3, 5), imshow(blurredImage), title('Blurred Image');
subplot(2, 3, 6), imshow(deblurredImage), title('Deblurred Image');
```

Exp 9

```
i = imread('earcell.jpg');
subplot(3,2,1); imshow(i); title('Original Image');

I = im2double(i);

level = graythresh(I);
BW = im2bw(I, level);
subplot(3,2,2); imshow(BW); title('Image graythresh');

level1 = 0.2 * BW;
subplot(3,2,3); imshow(level1); title('0.2 Slice');
```

```
level2 = 0.4 * BW;
subplot(3,2,4); imshow(level2); title('0.4 Slice');

level3 = 0.6 * BW;
subplot(3,2,5); imshow(level3); title('0.6 Slice');

level4 = 0.8 * BW;
subplot(3,2,6); imshow(level4); title('0.8 Slice');
```

Exp 10 :

```
i = imread('rosefinch.png');
g = rgb2gray(i);

subplot(2,2,1);
imshow(i);
title('Original Image');

subplot(2,2,2);
imshow(g);
title('Gray Image');

c = edge(g, 'canny');
subplot(2,2,3);
imshow(c);
title('Canny output');
```