**Assignment -3**

**1. Import an image, convert image types from 'jpg' to 'png', and save it in a different location**

% Import the image

img = imread('input.jpg');

% Convert and save the image as 'png'

imwrite(img, 'output.png');

**2. Perform arithmetic operations on two different images and compare the images**

% Import the images

img1 = imread('image1.jpg');

img2 = imread('image2.jpg');

% Ensure the images are of the same size

img2 = imresize(img2, size(img1(:,:,1)));

% Perform arithmetic operations

sum\_img = img1 + img2;

diff\_img = img1 - img2;

mul\_img = immultiply(img1, img2);

div\_img = imdivide(img1, img2);

% Display the results

figure;

subplot(2,2,1), imshow(sum\_img), title('Sum of Images');

subplot(2,2,2), imshow(diff\_img), title('Difference of Images');

subplot(2,2,3), imshow(mul\_img), title('Product of Images');

subplot(2,2,4), imshow(div\_img), title('Quotient of Images');

**3. Import an image, change the colour of the image to grey, smooth with a Gaussian filter, and show all images**

% Import the image

img = imread('input.jpg');

% Convert to grayscale

gray\_img = rgb2gray(img);

% Apply Gaussian filter

smoothed\_img = imgaussfilt(gray\_img, 2);

% Display all images

figure;

subplot(1,3,1), imshow(img), title('Original Image');

subplot(1,3,2), imshow(gray\_img), title('Grayscale Image');

subplot(1,3,3), imshow(smoothed\_img), title('Smoothed Image');

**4. Import an image, change the image size, rotate it by 30˚, and show all images**

% Import the image

img = imread('input.jpg');

% Change the size of the image

resized\_img = imresize(img, [300 300]);

% Rotate the image by 30 degrees

rotated\_img = imrotate(img, 30);

% Display all images

figure;

subplot(1,3,1), imshow(img), title('Original Image');

subplot(1,3,2), imshow(resized\_img), title('Resized Image');

subplot(1,3,3), imshow(rotated\_img), title('Rotated Image');

**5. Create a histogram of the grayscale image, apply histogram equalization, and show all images**

% Import the image

img = imread('input.jpg');

% Convert to grayscale

gray\_img = rgb2gray(img);

% Create histogram

figure;

subplot(2,2,1), imshow(gray\_img), title('Grayscale Image');

subplot(2,2,2), imhist(gray\_img), title('Histogram');

% Apply histogram equalization

equalized\_img = histeq(gray\_img);

% Display equalized image

subplot(2,2,3), imshow(equalized\_img), title('Equalized Image');

subplot(2,2,4), imhist(equalized\_img), title('Equalized Histogram');

**6. Add noise to the grayscale image, remove it using a median filter, and show all images**

% Import the image

img = imread('input.jpg');

% Convert to grayscale

gray\_img = rgb2gray(img);

% Add noise to the image

noisy\_img = imnoise(gray\_img, 'salt & pepper', 0.02);

% Apply median filter to remove noise

denoised\_img = medfilt2(noisy\_img);

% Display original, noisy, and denoised images

figure;

subplot(1,3,1), imshow(gray\_img), title('Original Image');

subplot(1,3,2), imshow(noisy\_img), title('Noisy Image');

subplot(1,3,3), imshow(denoised\_img), title('Denoised Image');