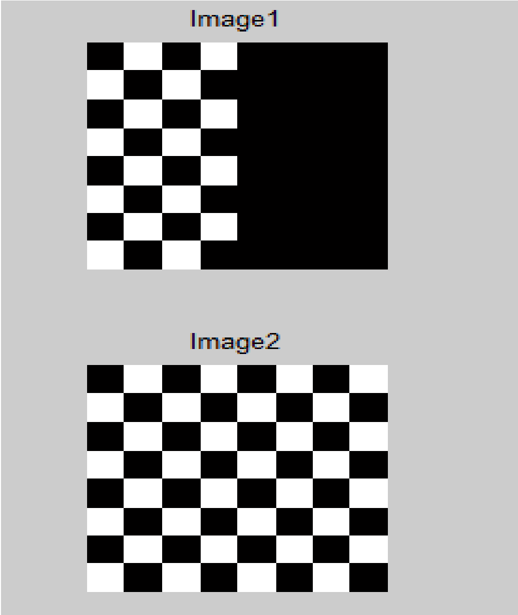



**RESULT:**

**REVIEW QUESTIONS:**

1. How is the mean computed for pixel values in an image and what does it represent?
2. What does the standard deviation reveal about an image's pixel values, and how is it calculated?
3. What is the significance of the correlation coefficient in image analysis, and how is it determined for two images?
4. Describe the concept of correlation coefficient as it pertains to image analysis. How is the correlation coefficient calculated for two images, and what does it indicate about their similarity or dissimilarity?
5. How can mean and standard deviation be used to assess the overall brightness and contrast of an image? Provide a step-by-step explanation with a practical example.

| **Ex. No.** | **IMPLEMENTATION OF IMAGE ENHANCEMENT-SPATIAL FILTERING** | **Date** |
|---|---|---|
| | | |

**AIM:**
The aim of this project is to implement image enhancement using spatial filtering techniques to improve the visual quality of digital images.

**SOFTWARE REQUIRED:**
MATLAB 2013b

**THEORY:**
Image enhancement through spatial filtering is a fundamental image processing technique. It involves the use of convolution operations with specific filter kernels to modify pixel values in an image. A basic spatial filter, such as an averaging filter, is used to reduce noise and enhance

the overall appearance of the image. The process involves convolving the image with the filter kernel, which replaces each pixel's value with a weighted average of its neighboring pixels. This can help to improve image quality, reduce artifacts, and enhance specific features.

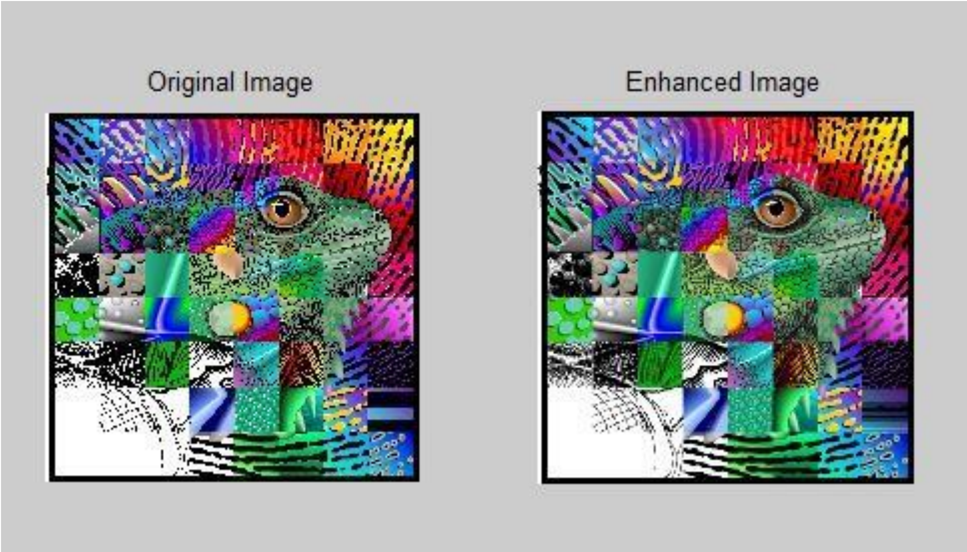
**PROCEDURE:**

1. Load the input image on which image enhancement will be performed.
2. Choose a suitable filter kernel based on the specific image enhancement goals. Common choices include averaging filters, Gaussian filters, or edge-enhancing filters.
3. Convolve the input image with the selected filter kernel using the `conv2` function in MATLAB.
4. Adjust the filter size and values to control the extent of enhancement.
5. Convert the resulting image to an appropriate data type (e.g., `uint8`) to ensure it falls within the valid pixel value range (0-255).
6. Display both the original and enhanced images using MATLAB's `imshow` function.
7. Save the enhanced image to a file for further analysis or use.

**PROGRAM:**

```
% Load the image
originalImage = imread('image2.jpg'); % the image filename

% Convert the image to double precision for better processing
originalImage = im2double(originalImage);

% Define the filter kernel (averaging filter)
filterSize = 3; % Size of the filter kernel (e.g., 3x3)
filter = ones(filterSize) / filterSize^2;

% Apply the filter using convolution
enhancedImage = imfilter(originalImage, filter, 'replicate'); % 'replicate' padding to handle
image borders

% Display the original and enhanced images
subplot(1, 2, 1);
imshow(originalImage);
title('Original Image');
subplot(1, 2, 2);
imshow(enhancedImage);
title('Enhanced Image');

% Save the enhanced image if needed
% imwrite(enhancedImage, 'enhanced_image.jpg');
```

**OUTPUT:**




**RESULT:**

**REVIEW QUESTIONS:**

1. What is the primary goal of image enhancement through spatial filtering?
2. Can you explain the concept of a filter kernel and its role in image enhancement using spatial filtering?
3. What is convolution, and how does it apply to image enhancement in this context?
4. What are some common spatial filter types used for image enhancement, and how do they differ in their effects on an image?
5. What considerations should be taken into account when selecting a filter kernel and its size for image enhancement, and how does this choice affect the output?

| Ex. No. | IMPLEMENTATION OF IMAGE ENHANCEMENT- FILTERING IN FREQUENCY DOMAIN | Date |
|---|---|---|
| | | |

**AIM:**

The aim of this project is to implement image enhancement through filtering in the frequency domain to improve the visual quality and extract specific features from digital images.

**SOFTWARE REQUIRED:**

MATLAB 2013b

**THEORY:**

Image enhancement through filtering in the frequency domain involves transforming an image from the spatial domain to the frequency domain using techniques like the Fast Fourier Transform (FFT). Once in the frequency domain, filtering operations are applied to enhance or suppress specific frequency components. The filtered image is then transformed back to the spatial domain using the Inverse Fast Fourier Transform (IFFT). This process can be used to remove noise, sharpen details, and improve image quality.

**PROCEDURE:**

1. Load the input image for image enhancement.
2. Apply the Fast Fourier Transform (FFT) to convert the image from the spatial domain to the frequency domain.
3. Choose an appropriate filter (e.g., high-pass, low-pass, or band-pass filter) to perform the desired enhancement operation.
4. Apply the selected filter to the transformed image in the frequency domain. This involves point-wise multiplication or convolution with the filter.
5. Use the Inverse Fast Fourier Transform (IFFT) to transform the filtered image back to the spatial domain.
6. Adjust the output image to ensure it falls within the valid pixel value range (0-255).
7. Display both the original and enhanced images.

**PROGRAM:**

```
clc
clear all
close all
a = imresize(imread('image.jpg’), [256, 256]);
m, n] = size(a);
mask = zeros(m, n);
for i=113:143
        for j=113:143
                mask(i, j) = 1;
        end
end
b = fftshift(fft2(a));
d = b .* mask;
e = abs(ifft2(d));
subplot(2, 2, 1)
imshow(a)
title('Orginal image')
subplot(2, 2, 2) imshow(uint8(e))
title('Low-pass filter filtered image')
subplot(2, 2, 3)
imshow(mask)
title('Low-pass filter mask')
```

**OUTPUT:**