Department of Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)

Image Enhancement Neighbourhood Processing Techniques:Smoothing Operators

Experiment : 5

AIM: To perform Image Enhancement Neighbourhood Processing Techniques: Smoothing Operators

PROBLEM STATEMENT:

1. Add Gaussian Noise, Remove using Averaging Filter
2. Add Salt & Pepper Noise, Remove using Median Filter

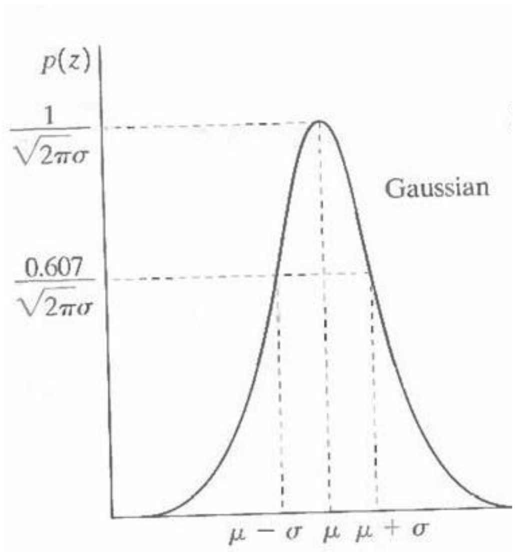
THEORY:

Gaussian Noise:

Gaussian (also called normal) noise models are used frequently in practice. In fact, this tractability is so convenient that it often results in Gaussian models being used in situations in which they are marginally applicable at best. The PDF of a Gaussian random variable, z, is given by

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2/2\sigma^2}$$

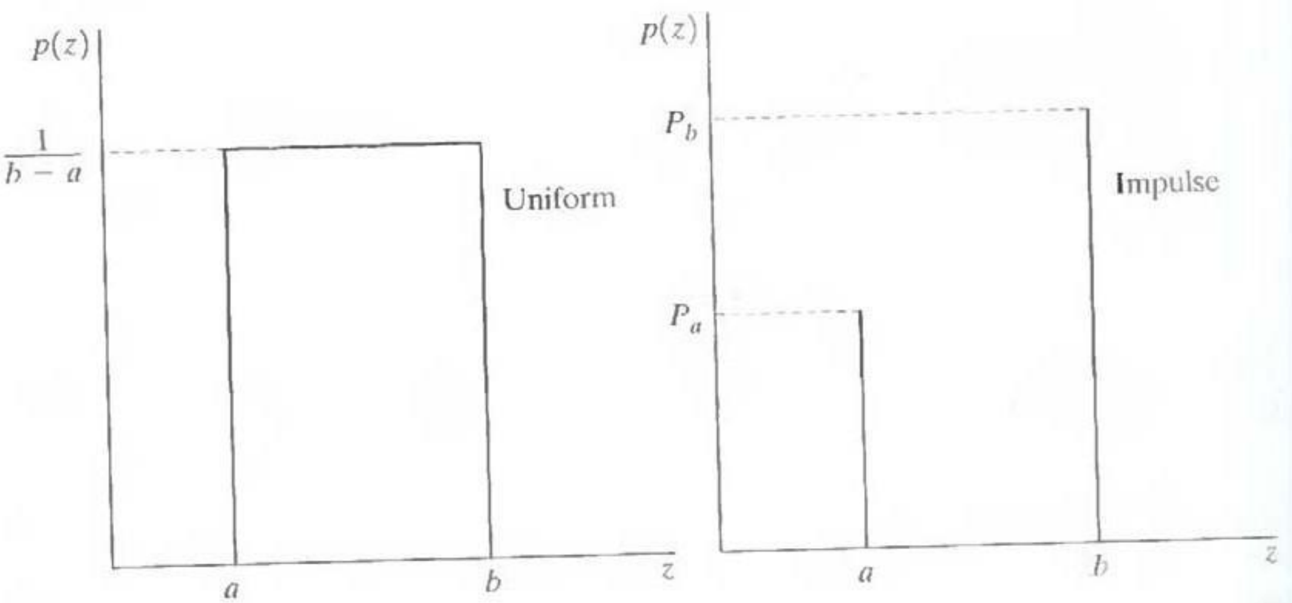
where z represents gray level, μ is the mean of average value of z, and a σ is its standard deviation. The standard deviation squared, σ2, is called the variance of z. A plot of this function is shown in Fig.



When z is described by Eq. (1), approximately 70% of its values will be in the range [(μ - σ), (μ +σ)], and about 95% will be in the range [(μ - 2σ), (μ + 2σ)].

Department of Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)
Image Enhancement Neighbourhood Processing Techniques:Smoothing Operators

Impulse (salt-and-pepper) noise :
The PDF of (bipolar) impulse noise is given by If b > a, gray-level b will appear as a light dot in the image. Conversely, level a will appear like a dark dot. If either Pa or Pb is zero, the impulse noise is called unipolar. If neither probability is zero, and especially if they are approximately equal, impulse noise values will resemble salt-and-pepper granules randomly distributed over the image. For this reason, bipolar impulse noise also is called salt-and-pepper noise. Shot and spike noise also are terms used to refer to this type of noise.

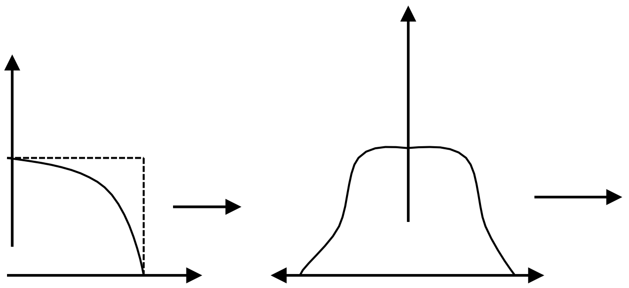


Smoothing

Low pass filtering removes the high frequency content from the image. It is used to remove thenoise present in the image which is generally a high frequency signal.

Averaging Filter

If an image has Gaussian noise, a low pass averaging filter is used to remove the noise. The frequency response and spatial response is show below.



Frequency Spatial Domain

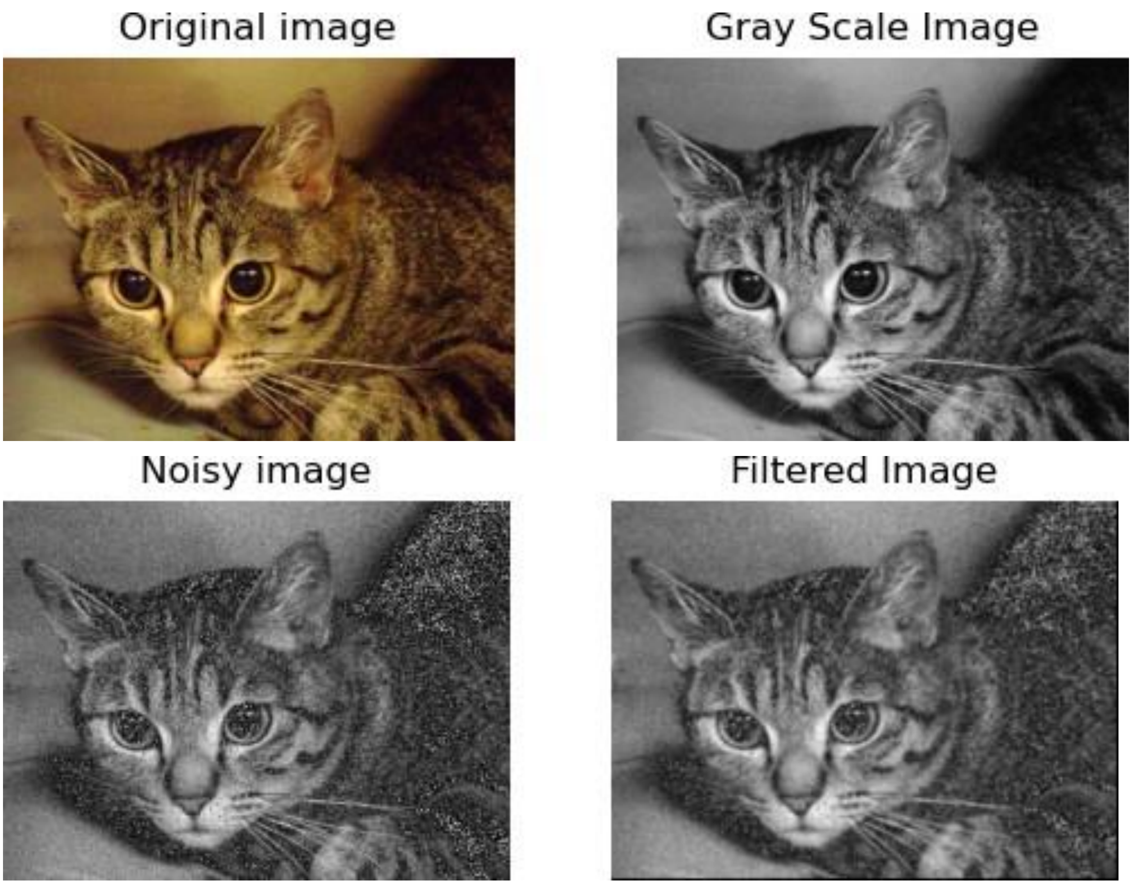
1/9

1	1	1
1	1	1
1	1	1

Averaging Mask

Department of Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)

Image Enhancement Neighbourhood Processing Techniques:Smoothing Operators
From spatial response we generate the mask that would give us the low pass filtering operation.Here all the coefficients are positive and the multiplying factor for a M X N matrix is 1 / (M X N).



1. Median Filtering

The averaging filter removes the noise by filtering it till it is no longer seen. But in the process italso blurs the edges. If we use averaging filter to remove salt and pepper noise from an image it will blur the noise but will also ruin the edges. Hence when we need to remove salt and pepper

noise we use non-linear filter known as Median Filter. They are also called order statistical filtersbecause their response is based on the ordering and ranking of the pixels contained within the mask.

Department of Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)
Image Enhancement Neighbourhood Processing Techniques:Smoothing Operators

Salt and Pepper Noise

Filtered Image

When an Averaging Filter is used to a picture with Gaussian Noise, the outcome is a smoothed image with reduced noise. Unfortunately, the image may lose some of its clarity and features as a result, particularly in areas containing high-frequency components. The averaging filter is appropriate for Gaussian noise because it smoothes the image by averaging the pixels in a region. As a result, applying the Averaging Filter on a picture with Gaussian Noise blurs the image.
Using a Median Filter on a picture with Salt and Pepper Noise can efficiently eliminate the noise while keeping the image's borders and features. The Median filter replaces the pixel value of a neighbourhood with the neighborhood's median value, making it especially useful for impulsive noise such as salt and pepper noise

Lab Assignments to complete in this session
Problem Statement: Develop a Python program utilizing the OpenCV library to enhance the images in spatial domain. The program should address the following tasks:

Steps to perform median filtering:

1. Assume 3 X 3 empty mask.
2. Place it on left hand side corner.
3. Arrange the 9 pixels in ascending or descending order
4. Choose the median from these nine values.
5. Place this median at the centre.
6. Move the mask in similar fashion as the averaging mask

In median filtering the grey level of the center pixel is replaced by the median filter value of the neighborhood.

Department of Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)

Image Enhancement Neighbourhood Processing Techniques:Smoothing Operators

The solution to the operations performed must be produced by scratch coding without the use of built in OpenCV methods.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Function to display images for comparison
def display_images(original, noisy, denoised, title1, title2, title3):
    plt.figure(figsize=(12, 4))


    plt.subplot(1, 3, 1)
    plt.imshow(original, cmap='gray')
    plt.title(title1)
    plt.axis('off')


    plt.subplot(1, 3, 2)
    plt.imshow(noisy, cmap='gray')
    plt.title(title2)
    plt.axis('off')


    plt.subplot(1, 3, 3)
    plt.imshow(denoised, cmap='gray')
    plt.title(title3)
    plt.axis('off')

    plt.show()


# Function to add Gaussian noise to the image
def add_gaussian_noise(image):
    row, col = image.shape
    mean = 0
    std_dev = 20
    gaussian_noise = np.random.normal(mean, std_dev, (row,
col)).astype('uint8')

    noisy_image = cv2.add(image, gaussian_noise)  # Adding the noise
to the original image
    return noisy_image

# Function to remove Gaussian noise using averaging filter
def remove_gaussian_noise(image):
    denoised_image = cv2.blur(image, (5, 5))  # Averaging filter with
a 5x5 kernel
    return denoised_image


# Function to remove salt and pepper noise using median filter
def remove_salt_and_pepper_noise(image):
    denoised_image = cv2.medianBlur(image, 5)  # Median filter with a
```

```
kernel size of 5
    return denoised_image


# Function to add salt and pepper noise to the image
def add_salt_and_pepper_noise(image, prob):
    noisy_image = np.copy(image)
    row, col = image.shape

    # Salt noise (white pixels)
    num_salt = np.ceil(prob * row * col).astype(int)
    coords = [np.random.randint(0, i - 1, num_salt) for i in
image.shape]
    noisy_image[coords[0], coords[1]] = 255

    # Pepper noise (black pixels)
    num_pepper = np.ceil(prob * row * col).astype(int)
    coords = [np.random.randint(0, i - 1, num_pepper) for i in
image.shape]
    noisy_image[coords[0], coords[1]] = 0

    return noisy_image

# Main processing function
def process_image(image_path):

    original_image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # Gaussian Noise and its removal
    noisy_gaussian = add_gaussian_noise(original_image)
    denoised_gaussian = remove_gaussian_noise(noisy_gaussian)


    display_images(original_image, noisy_gaussian, denoised_gaussian,
                   'Original Image', 'Gaussian Noise', 'Averaging
Filter')

    # Salt and Pepper Noise and its removal
    noisy_salt_pepper = add_salt_and_pepper_noise(original_image,
0.02)
    denoised_salt_pepper =
remove_salt_and_pepper_noise(noisy_salt_pepper)


    display_images(original_image, noisy_salt_pepper,
denoised_salt_pepper,
                   'Original Image', 'Salt & Pepper Noise', 'Median
Filter')


image_path = "/content/download.jpeg"
```

```
process_image(image_path)
```