**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Department of Electronics and Telecommunication Engineering**

**Subject: Image and Video Processing Program: B.Tech/BTI/MBA**

**Sem: VII/IX/V ACAY:2020-21**

**EXPERIMENT NO. 4**

**Aim:**

1. To write a program in PYTHON to perform low pass averaging filtering in spatial domain on an image with Gaussian noise.
2. To write a program in PYTHON to perform median filtering in spatial domain on an image with salt and pepper noise.

**Software:**  PYTHON

**Prerequisite:**

|  |  |
| --- | --- |
| Sr. No | Concepts |
| 1. | Image Enhancement using spatial domain filters (neighborhood processing) |

**Outcome:**

After successful completion of this experiment students will be able to:

Concept of image enhancement in spatial domain using neighborhood processing methods

**Theory:**

**Neighborhood processing in spatial domain:**

Here, to modify one pixel, we consider values of the immediate neighboring pixels also. For this purpose, 3X3, 5X5 or 7X7 neighborhood mask can be considered. Example of 3X3 mask is shown below.

|  |  |  |
| --- | --- | --- |
| f(x-1,y-1) | f(x-1,y) | f(x-1,y+1) |
| f(x,y-1) | f(x,y) | f(x,y+1) |
| f(x+1,y-1) | f(x+1,y) | f(x+1,y+1) |

**Low Pass filtering**

It is also known as smoothing filter. It removes the high frequency content from the image. Low pass averaging filter mask is as shown.

|  |  |  |
| --- | --- | --- |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

**Median Filtering**

It is also known as nonlinear filtering. It is used to eliminate salt and pepper noise. Here the pixel value is replaced by median value of the neighbouring pixel.

|  |
| --- |
| Name of the Experiment: To apply spatial filtering on the given image |
| Roll No. C018 Name: Dhruvit Jain |
| Program: B.Tech ExTC Semester : VII |
| Date of Performance:31/07/2020 Date of Submission: 31/07/2020 |

**Code for averaging filtering:**

import matplotlib.pyplot as plt

from skimage import io

from skimage.color import rgb2gray

import numpy as np

from scipy import signal

image\_ori = io.imread ('C:/Users/dhruv/Desktop/College/NOTES/IVP/Labs/baboon.png')

io.imshow(image\_ori)

#reduce noise by avergaing filter

image = image\_ori

sh= image.shape

#add gaussian noise

mean = 0

sigma = 10

gaussian\_noise = np.random.normal (mean,sigma,(sh[0],sh[1])) #size should be tuple

image\_gn = image.copy ()

image\_gn[:,:,0] = image [:,:,0]+gaussian\_noise

image\_gn[:,:,1] = image [:,:,1]+gaussian\_noise

image\_gn[:,:,2] = image [:,:,2]+gaussian\_noise

io.imshow(image\_gn)

#constructing avg filter

size = 3

avg\_filter = np.ones((size,size))/(size\*size) #size should be in tuple

print (avg\_filter)

image\_avg = image\_gn.copy()

image\_avg [:,:,0] = signal.convolve2d(image\_gn[:,:,0],avg\_filter,mode='same')

image\_avg [:,:,1] = signal.convolve2d(image\_gn[:,:,1],avg\_filter,mode='same')

image\_avg [:,:,2] = signal.convolve2d(image\_gn[:,:,2],avg\_filter,mode='same')

io.imshow (image\_avg)

plt.figure()

plt.figure(figsize=(10,10))

plt.subplot(1,3,1)

io.imshow (image)

plt.title ('Original Image')

plt.subplot(1,3,2)

io.imshow (image\_gn)

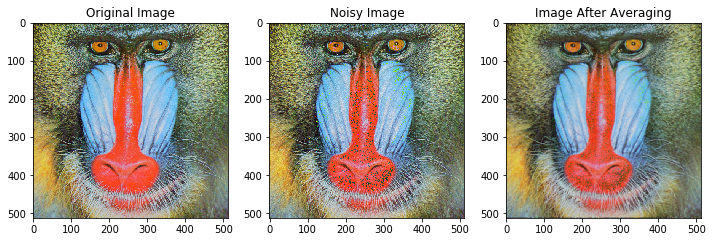
plt.title ('Noisy Image')

plt.subplot(1,3,3)

io.imshow (image\_avg)

plt.title ('Image After Averaging')

**Outputs for averaging filter:**



**Code for median filter:**

#median filter to remove salt pepper noise

from random import randint,seed

image = rgb2gray (image)

io.imshow (image)

sh=image.shape

image\_sp = image.copy ()

#portion of image corrupted by noise, defined by a

a=0.1 #10% of image

sp\_numb = int (sh[0]\*sh[1]\*a) #number of pixels to be corrupted

sp\_val=1 #value of noise

seed (1)

for i in range (0,sp\_numb):

temp1=randint (0,sh[0]-1)

temp2= randint (0,sh[1]-1)

image\_sp [temp1][temp2] = sp\_val

io.imshow (image\_sp)

img\_spfilt = image\_sp.copy()

size = 3

#centre of mask

cent = int ((size-1)/2)

#median

med = int (((size\*size)-1)/2)

for row in range (0,sh[0]-size):

for col in range (0,sh[1]-size):

temp1 = image\_sp [row:row+size,col:col+size]

temp2 = np.reshape (temp1,(1,size\*size))

temp3 = np.sort (temp2)

img\_spfilt[row+cent][col+cent]=temp3[0][med]

plt.figure()

plt.figure(figsize=(10,10))

plt.subplot(1,3,1)

io.imshow (image)

plt.title ('Original Image')

plt.subplot(1,3,2)

io.imshow (image\_sp)

plt.title ('Salt Pepper Noise Image')

plt.subplot(1,3,3)

io.imshow (img\_spfilt)

plt.title ('Image After median Filter')

**Outputs for median filter:**



**Conclusion:**

Average filter reduces gaussian noise, hence can be used to enhance the quality of a noisy image. Average filter blurs/smoothens the image. As size of average filter increases, blurring increases.

Median filter removes salt (white) as well as pepper (black) noise. As we increase the number of pixels affected by salt and pepper noise, the quality of image detoriates. However, it can be removed by using median filters of various sizes.