**Experiment 4**

**Aim:** Apply spatial filtering

**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)

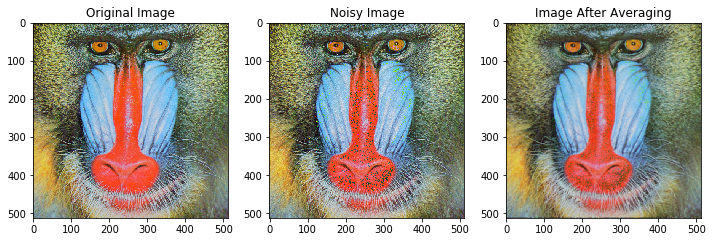
plt.title ('Noisy Image')

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.