exp : spatial fillter

import numpy as np

import cv2

import matplotlib.pyplot as plt

from scipy import ndimage

f\_img = cv2.imread("/content/cameraman.tif")

cv2\_imshow(f\_img)

**output:**



#Convert RGB image to Gray image

gray\_img = cv2.cvtColor(f\_img, cv2.COLOR\_BGR2GRAY)

cv2\_imshow(gray\_img) #show gray image

**output:**



data\_1 = np.array(gray\_img, dtype=float)

kernel\_1 = np.array([[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]])

highpass\_filter = ndimage.convolve(data\_1, kernel\_1)

cv2\_imshow(highpass\_filter)

**output:**

****

data\_2 = np.array(gray\_img, dtype=float)

kernel\_2 = np.array([[1/9,1/9,1/9], [1/9, 1/9, 1/9], [1/9, 1/9,1/9]])

lowpass\_filter = ndimage.convolve(data\_2, kernel\_2)

cv2\_imshow(lowpass\_filter)

output:



import numpy as np

import cv2

import matplotlib.pyplot as plt

from scipy import ndimage

from scipy.ndimage import gaussian\_filter

f\_img = cv2.imread("/content/cameraman.tif")

cv2\_imshow(f\_img)

g = cv2.GaussianBlur(f\_img,(3,3),0)

cv2\_imshow(g)





img = cv2.imread('/content/cameraman\_salt\_and\_pepper\_noise.tif', cv2.IMREAD\_GRAYSCALE)

img\_out = img.copy()

height = img.shape[0]

width = img.shape[1]

cv2\_imshow(img)

# kernal

for i in np.arange(2, height-3):

for j in np.arange(2, width-3):

neighbors\_med = []

neighbors\_med.clear()

for k in np.arange(-2, 3):

for l in np.arange(-2, 3):

a = img.item(i+k, j+l)

neighbors\_med.append(a)

neighbors\_med.sort()

median = neighbors\_med[4]

b = median

img\_out.itemset((i,j), b)

cv2\_imshow(img\_out)

output:



