Experiment no.- 3

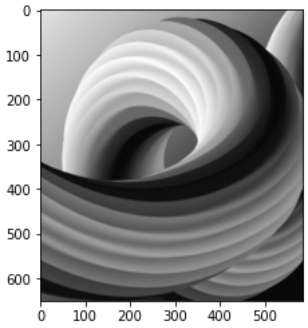
- Title- Image smoothing using spatial domain filters
- Name- Palash Khandelwal
- Roll no.- PB 21
- PRN no.- 1032201050
- Date of performance- 01/02/23

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

```
img = cv2.imread('img1.png')
img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
```

```
plt.imshow(img,cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7f1384950a00>
```

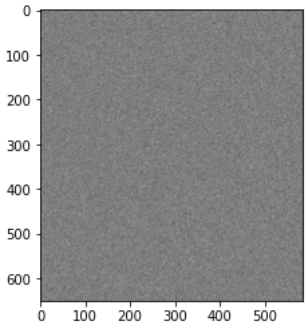



```
print(img.shape)
```

```
(650, 586)
```

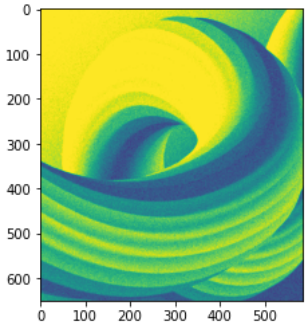
```
gauss_noise = np.zeros((650,586),dtype = np.uint8)
cv2.randn(gauss_noise,128,20)
gauss_noise=(gauss_noise*0.5).astype(np.uint8)
plt.imshow(gauss_noise, cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7f1384bd1700>
```



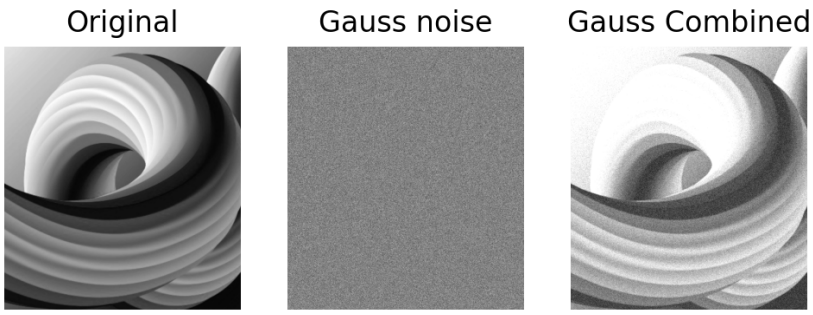

```
gn_img= cv2.add(img, gauss_noise)
plt.imshow(gn_img)
```

```
<matplotlib.image.AxesImage at 0x7f1384e136a0>
```



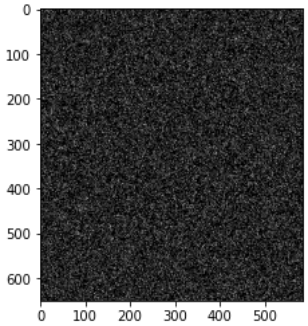
```
fig = plt.figure(dpi = 200)

fig.add_subplot(1,3,1)
plt.imshow(img,cmap = 'gray')
plt.axis("off")
plt.title("Original")


fig.add_subplot(1,3,2)
plt.imshow(gauss_noise,cmap = 'gray')
plt.axis("off")
plt.title("Gauss noise")


fig.add_subplot(1,3,3)
plt.imshow(gn_img,cmap = 'gray')
plt.axis("off")
plt.title("Gauss Combined")
```

```
Text(0.5, 1.0, 'Gauss Combined')
```



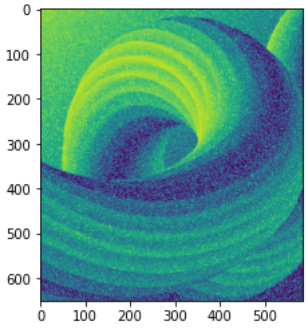

```
 imp_noise = np.zeros((650,586),dtype = np.uint8)
 cv2.randu(imp_noise, 0,255)
 imp_noise=cv2.threshold(imp_noise, 215,255,cv2.THRESH_BINARY)[1]
 plt.imshow(imp_noise, cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7f1384a65af0>
```



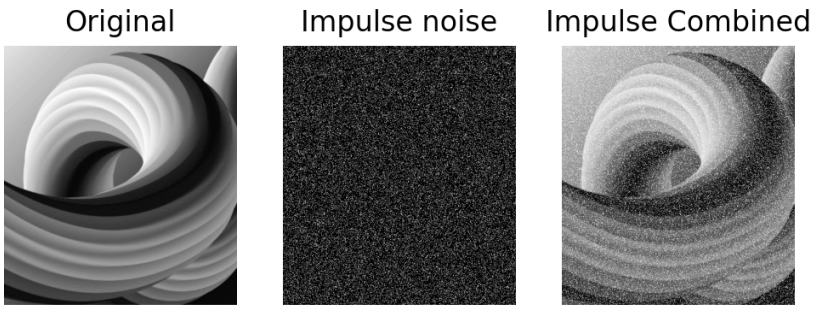

```
im_img= cv2.add(img, imp_noise)
plt.imshow(im_img)
```

```
<matplotlib.image.AxesImage at 0x7f1384a3e8e0>
```



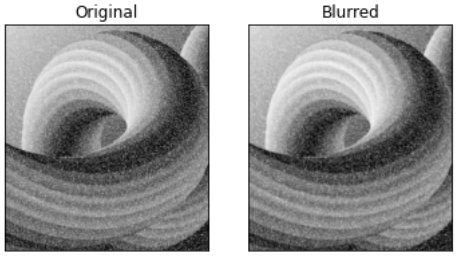
```
fig = plt.figure(dpi = 200)

fig.add_subplot(1,3,1)
plt.imshow(img,cmap = 'gray')
plt.axis("off")
plt.title("Original")


fig.add_subplot(1,3,2)
plt.imshow(imp_noise,cmap = 'gray')
plt.axis("off")
plt.title("Impulse noise")


fig.add_subplot(1,3,3)
plt.imshow(im_img,cmap = 'gray')
plt.axis("off")
plt.title("Impulse Combined")
```

```
Text(0.5, 1.0, 'Impulse Combined')
```




```
# Low pass filter
blur1 = cv2.blur(im_img, (3,3))

plt.subplot(121), plt.imshow(im_img, cmap = 'gray'), plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(blur1, cmap = 'gray'), plt.title('Blurred')
plt.xticks([]), plt.yticks([])
```

```
(([], <a list of 0 Text major ticklabel objects>),
 ([], <a list of 0 Text major ticklabel objects>))
```

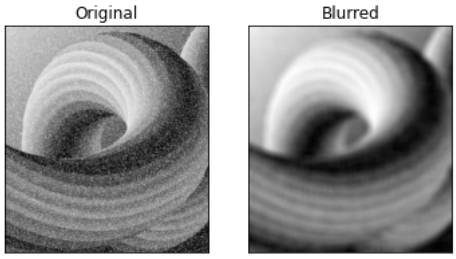



```
blur2 = cv2.blur(im_img, (9,9))

plt.subplot(121), plt.imshow(im_img, cmap = 'gray'), plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(blur2, cmap = 'gray'), plt.title('Blurred')
plt.xticks([]), plt.yticks([])
```

```
(([], <a list of 0 Text major ticklabel objects>),
 ([], <a list of 0 Text major ticklabel objects>))
```

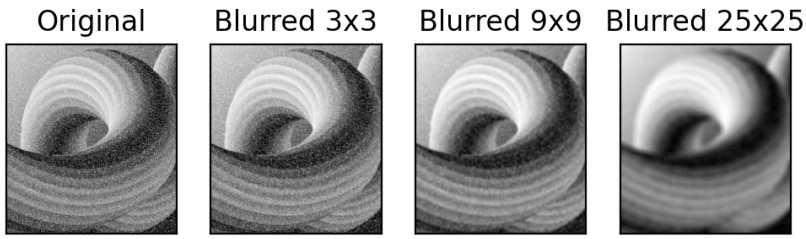
```
blur3 = cv2.blur(im_img, (25,25))

plt.subplot(121), plt.imshow(im_img, cmap = 'gray'), plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(blur3, cmap = 'gray'), plt.title('Blurred')
plt.xticks([]), plt.yticks([])
```

```
(([], <a list of 0 Text major ticklabel objects>),
 ([], <a list of 0 Text major ticklabel objects>))
```



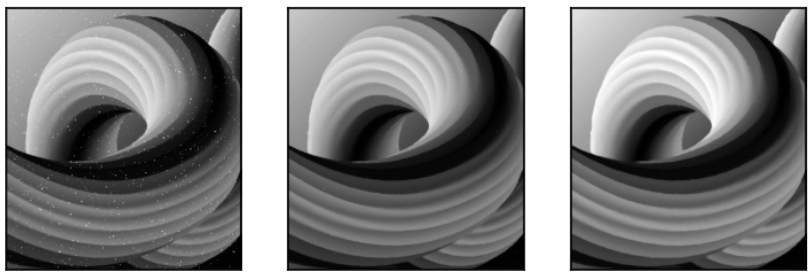

```
fig= plt.figure(dpi=200)
plt.subplot(141), plt.imshow(im_img, cmap = 'gray'), plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(142), plt.imshow(blur1, cmap = 'gray'), plt.title('Blurred 3x3')
plt.xticks([]), plt.yticks([])
plt.subplot(143), plt.imshow(blur2, cmap = 'gray'), plt.title('Blurred 9x9')
plt.xticks([]), plt.yticks([])
plt.subplot(144), plt.imshow(blur3, cmap = 'gray'), plt.title('Blurred 25x25')
plt.xticks([]), plt.yticks([])
```

```
(([], <a list of 0 Text major ticklabel objects>),
 ([], <a list of 0 Text major ticklabel objects>))
```



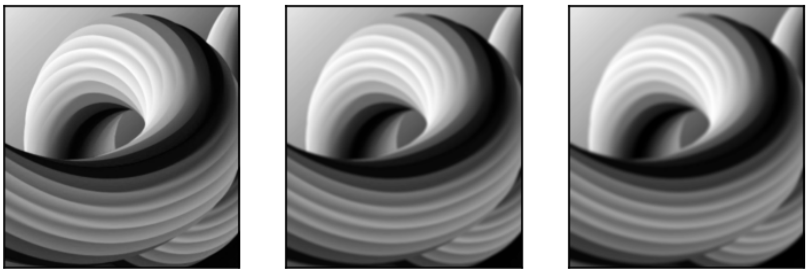

```
median1 = cv2.medianBlur(im_img, 3)
median2 = cv2.medianBlur(im_img, 5)
median3 = cv2.medianBlur(im_img, 9)
fig= plt.figure(dpi=150)
plt.subplot(131),plt.imshow(median1, cmap = 'gray')
plt.xticks([]), plt.yticks([])
plt.subplot(132),plt.imshow(median2, cmap = 'gray')
plt.xticks([]), plt.yticks([])
plt.subplot(133),plt.imshow(median3, cmap = 'gray')
plt.xticks([]), plt.yticks([])
```

```
(([], <a list of 0 Text major ticklabel objects>),
 ([], <a list of 0 Text major ticklabel objects>))
```



```
img_gaussian_blur1= cv2.GaussianBlur(img,(3,3),cv2.BORDER_DEFAULT)
img_gaussian_blur2= cv2.GaussianBlur(img,(9,9),cv2.BORDER_DEFAULT)
img_gaussian_blur3= cv2.GaussianBlur(img,(15,15),cv2.BORDER_DEFAULT)
fig= plt.figure(dpi=150)
plt.subplot(131),plt.imshow(img_gaussian_blur1, cmap = 'gray')
plt.xticks([]), plt.yticks([])
plt.subplot(132),plt.imshow(img_gaussian_blur2, cmap = 'gray')
plt.xticks([]), plt.yticks([])
plt.subplot(133),plt.imshow(img_gaussian_blur3, cmap = 'gray')
plt.xticks([]), plt.yticks([])
```

```
(([], <a list of 0 Text major ticklabel objects>),
 ([], <a list of 0 Text major ticklabel objects>))
```



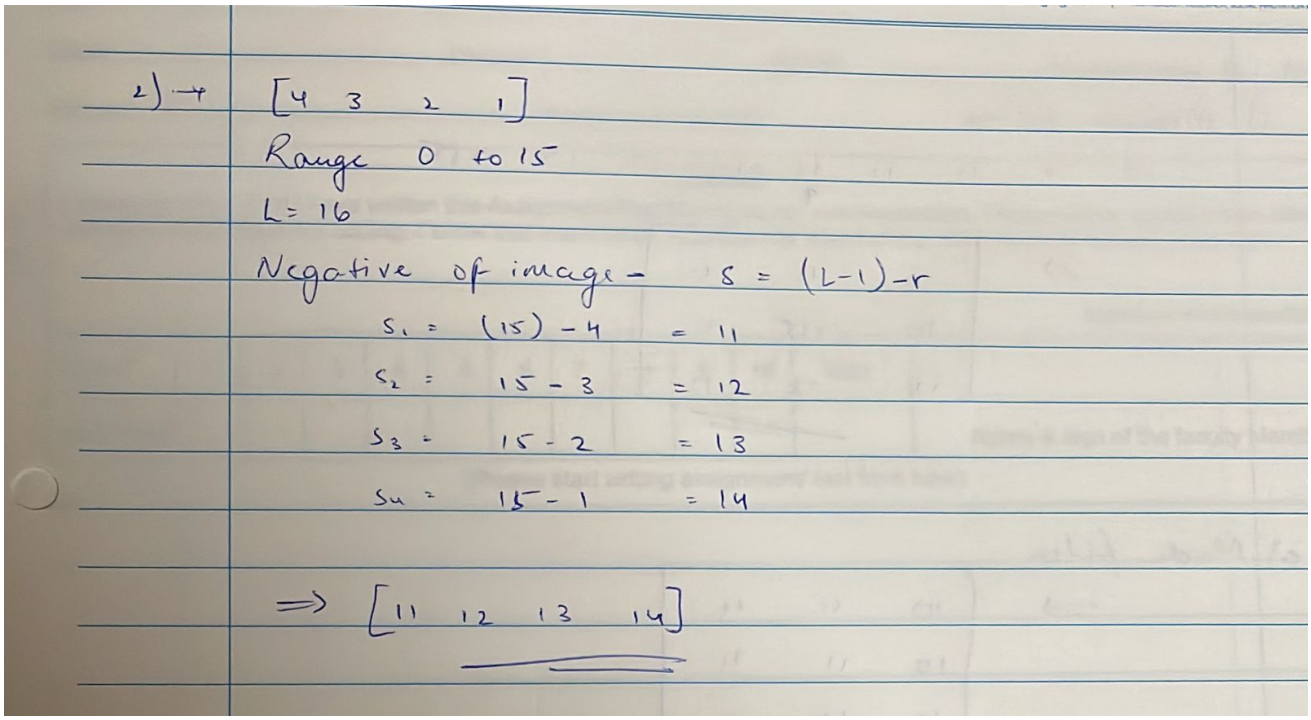
Result and Conclusion- Low pass filter in spatial domain is used as smoothing filter and fill up gap between image.

Post Lab Questions-

1. What is point processing?
   Point processing is now defined as an operation which calculates the new value of a pixel in g(x,y) based on the value of the pixel in the same position in f(x,y) and some operation.

---

2. Consider an input image row [4 3 2 1] with intensity values in the range of 0 to 15. Determine the negative of the image row.



---

3. Determine the new value of the central pixel of the following image by applying 3x3 size: a) Mean filter b) Median filter c) Mode filter 10 11 11 10 255 11 12 12 11

3) → Image-

$$\begin{bmatrix} 10 & 11 & 11 \\ 10 & 255 & 11 \\ 12 & 12 & 11 \end{bmatrix}$$

a) Mean filter (3x3)

$(10+11+11+10+255+11+12+12+11)/9 = 38$

⇒

$$\begin{bmatrix} 10 & 11 & 11 \\ 10 & 38 & 11 \\ 12 & 12 & 11 \end{bmatrix}$$

b) Median filter

10 10 11 11 11 11 12 12 255

↑

→

$$\begin{bmatrix} 10 & 11 & 11 \\ 10 & 255 & 11 \\ 12 & 12 & 11 \end{bmatrix}$$

c) Mode fiter

⇒

$$\begin{bmatrix} 10 & 11 & 11 \\ 10 & 11 & 11 \\ 12 & 12 & 11 \end{bmatrix}$$