Fourier Transform

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

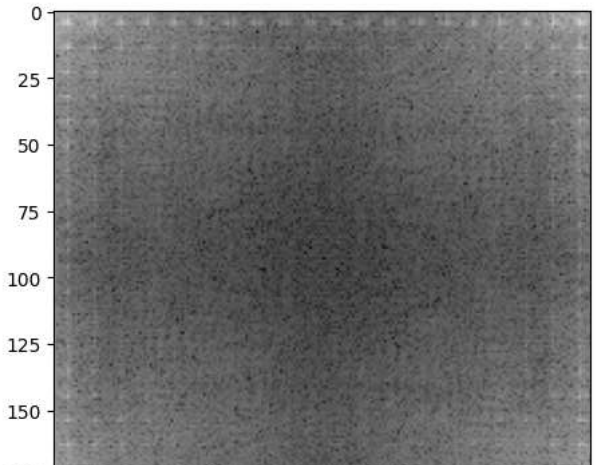
```
img = cv2.imread('/content/Ảnh chụp màn hình 2023-10-03 164636.png')
#(1) chuyển ảnh sang gray
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.figure(figsize = (5, 5))
plt.imshow(gray, cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a8947c70>
```

1 . Low pass frequency by using Distance smooth function

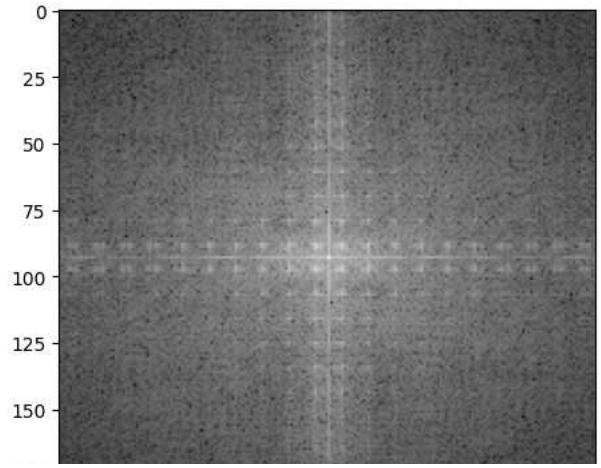
```
# (2) Fourier transformation
f = np.fft.fft2(gray)
plt.imshow(np.log(np.abs(f)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a8877eb0>
```



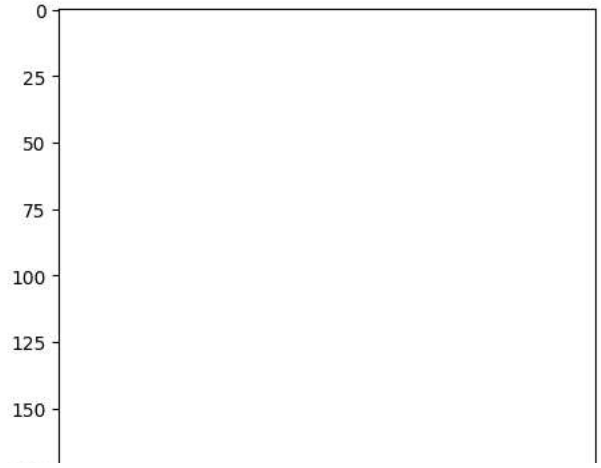

```
# (3) frequency shift

shifted_f = np.fft.fftshift(f)
plt.imshow(np.log(np.abs(shifted_f)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a8730ca0>
```



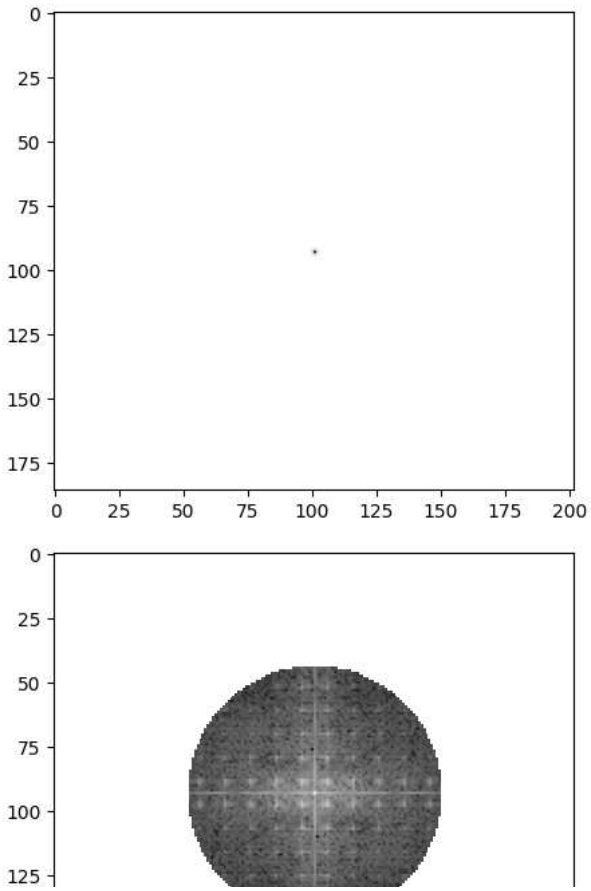

```
#(4a) low pass frequency
low_pass = np.ones(shape = gray.shape)
plt.imshow(low_pass, cmap = 'gray', vmax = 1, vmin = 0)
```

```
<matplotlib.image.AxesImage at 0x7bd4a87c3c40>
```



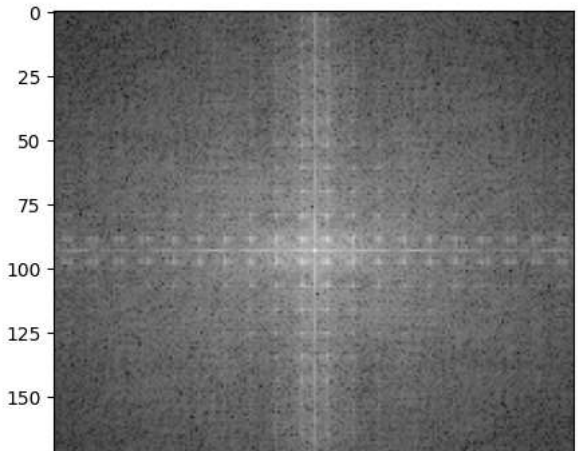

```
xc = gray.shape[1] // 2
yc = gray.shape[0] // 2

Do = 350

def d(x,y):
  return np.sqrt((x-xc)**2 + (y-yc)**2)

for x in range(gray.shape[1]):
  for y in range(gray.shape[0]):
    if d(x,y) > Do:
      low_pass[y,x] = 0

shifted_g = shifted_f * low_pass
plt.figure(figsize = (5,5))
plt.imshow(np.log(np.abs(low_pass)), cmap= 'gray')
plt.figure(figsize = (5,5))
plt.imshow(np.log(np.abs(shifted_g)), cmap = 'gray')
```

```
<ipython-input-33-fd02fb9f22aa>:16: RuntimeWarning: divide by zero encountered in log
  plt.imshow(np.log(np.abs(low_pass)), cmap= 'gray')
<ipython-input-33-fd02fb9f22aa>:18: RuntimeWarning: divide by zero encountered in log
  plt.imshow(np.log(np.abs(shifted_g)), cmap = 'gray')
<matplotlib.image.AxesImage at 0x7bd49c7f20b0>
```




```
# (6) invert shift

f = np.fft.ifftshift(f)
plt.figure(figsize=(5,5))
plt.imshow(np.log(np.abs(f)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a857bb50>
```




```
new_img = np.abs(np.fft.ifft2(f))
plt.figure(figsize=(5,5))
```

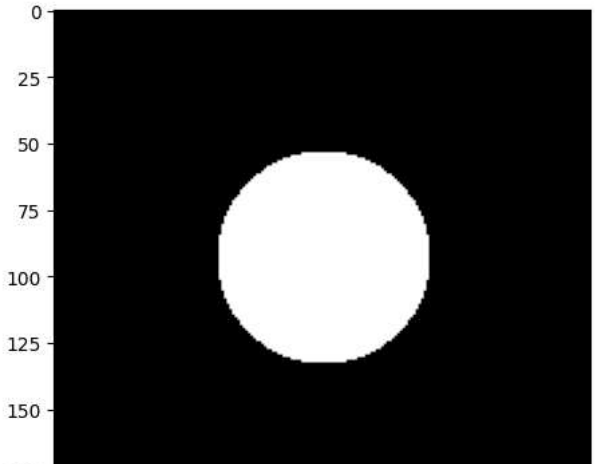
```
plt.imshow(new_img, cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a853a080>
```

2. Hight pass frequency by using Distance smooth function

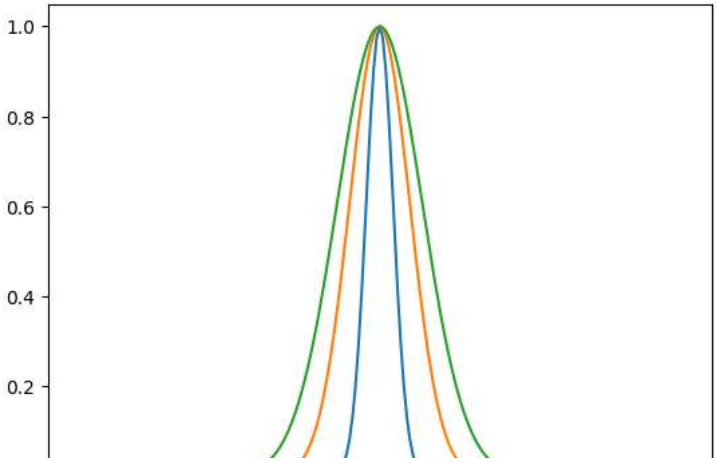
```
#(4b)
# lấy tâm bức ảnh

xc = gray.shape[1] // 2
yc = gray.shape[0] // 2


# bán kính Do
Do = 40


# Tính khoảng cách từ 1 điểm đến tâm với công thức
# sqrt((x -xc)**2 + (y - yc)**2)
def d(x, y):
  return np.sqrt((x - xc)**2 + (y - yc)**2)

for x in range(gray.shape[1]):
  for y in range(gray.shape[0]):
    if d(x, y) < Do:
      low_pass[y, x] = 255

f = f * low_pass
plt.imshow(np.log(np.abs(low_pass)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a85789d0>
```



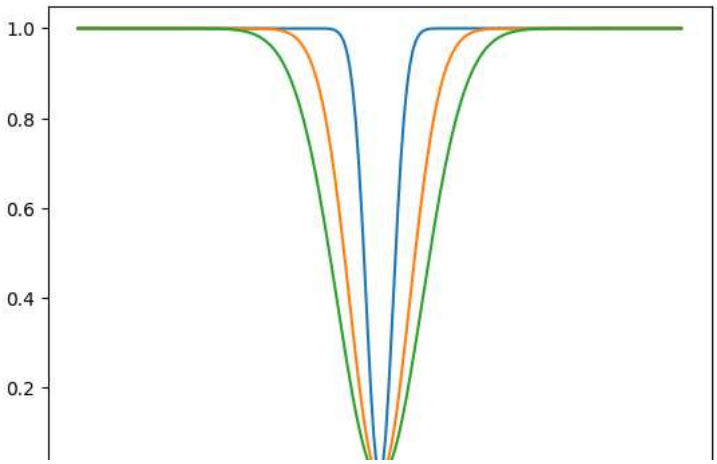
```
for sigma in [0.1, 0.5, 1.0]:
  x = np.linspace(-5, 5, 201)
  y = np.exp(-x**2 / sigma)
  plt.plot(x, y, label = " ".format(sigma))
plt.legend
```

```
<function matplotlib.pyplot.legend(*args, **kwargs)>
```

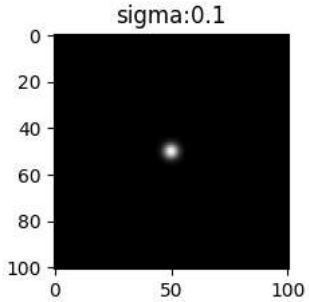


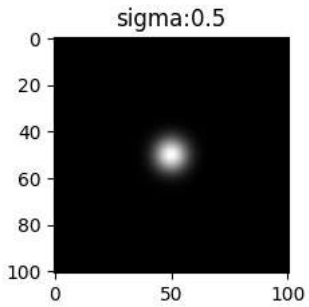

```
for sigma in [0.1, 0.5, 1.0]:
  x = np.linspace(-5, 5, 201)
  y = 1 - np.exp(-x**2 / sigma)
  plt.plot(x, y, label = " ".format(sigma))
plt.legend
```

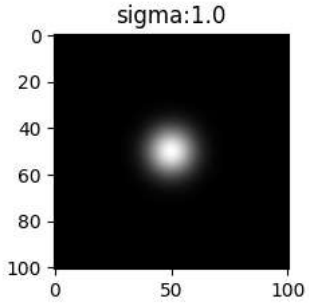
```
<function matplotlib.pyplot.legend(*args, **kwargs)>
```




```
x = np.linspace(-5, 5, 101)
y = np.linspace(-5, 5, 101)

X, Y = np.meshgrid(x, y)

for i, sigma in enumerate([0.1, 0.5, 1.0, 5.0]):
  z = np.exp(-(X**2 + Y**2) / sigma)
  plt.figure(figsize=(5,5))
  plt.subplot(2,2,i+1)
  plt.imshow(z, cmap = 'gray')
  plt.title(f"sigma:{sigma}")
```







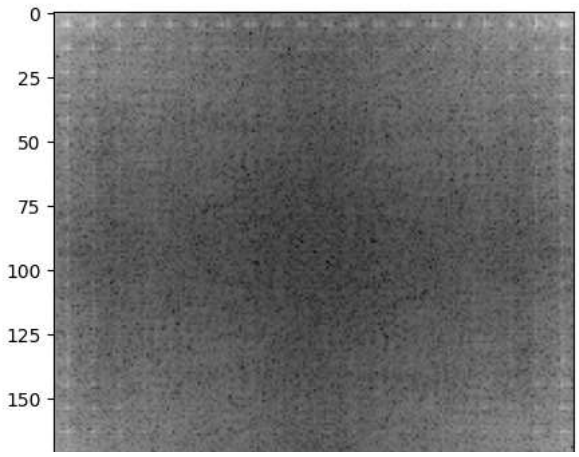
\3. Low pass frequency by using Gaussian smooth function

```
img = cv2.imread('/content/Ảnh chụp màn hình 2023-10-03 164636.png')

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.figure(figsize = (5, 5))
plt.imshow(gray, cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a82b57e0>
```

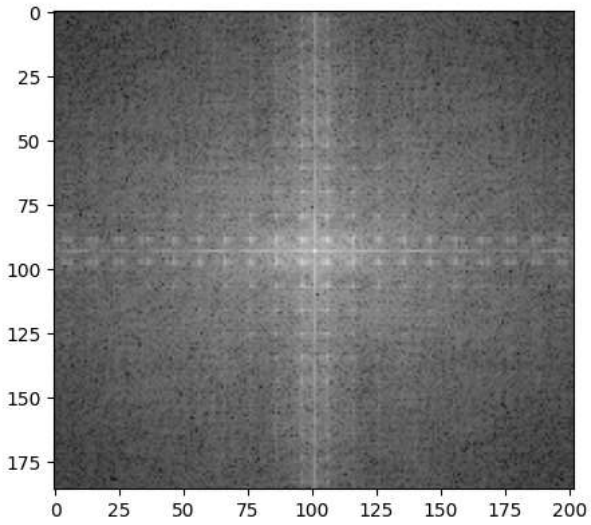
```
f = np.fft.fft2(gray)
plt.figure(figsize=(5,5))
plt.imshow(np.log(np.abs(f)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a812d330>
```



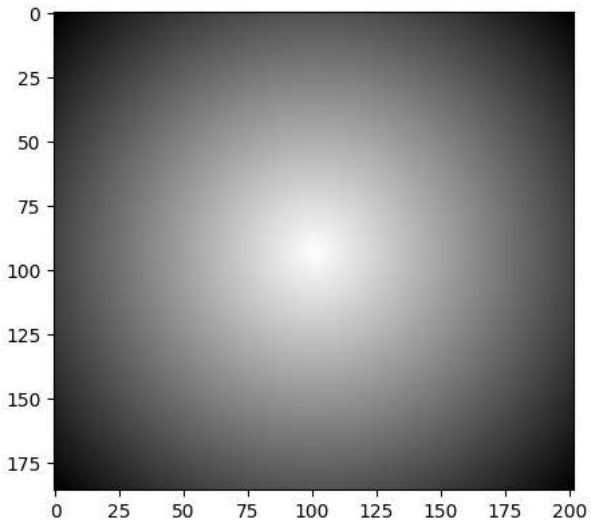

```
shifted_f = np.fft.fftshift(f)
plt.figure(figsize=(5,5))
plt.imshow(np.log(np.abs(shifted_f)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a819f430>
```




```
low_pass = np.ones(shape = gray.shape)

xc = gray.shape[1] // 2
yc = gray.shape[0] // 2

Do = 50

def d(x, y):
  return np.sqrt((x - xc)**2 + (y - yc)**2)

for x in range(gray.shape[1]):
  for y in range(gray.shape[0]):
    low_pass [y,x] = np.exp(-d(x,y))**2 / (2*sigma**2)

f = f * low_pass
plt.figure(figsize=(5,5))
plt.imshow(np.log(np.abs(low_pass)), cmap = 'gray')
```

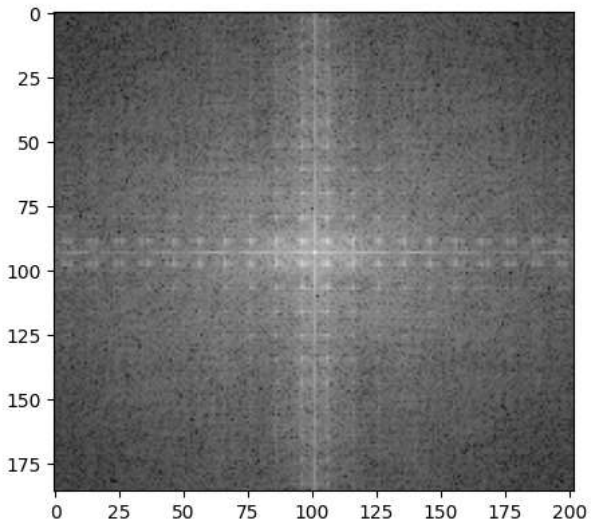
```
<matplotlib.image.AxesImage at 0x7bd4a80541f0>
```



3. Low pass frequency by using Gaussian smooth function

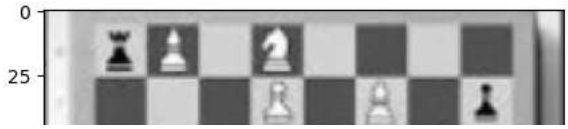
```
f = np.fft.fft2(gray)
f = np.fft.ifftshift(f)
plt.figure(figsize=(5,5))
plt.imshow(np.log(np.abs(f)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd4a80d8250>
```




```
new_img = np.abs(np.fft.ifft2(f))
plt.figure(figsize=(5,5))
plt.imshow(new_img, cmap = 'gray')
```

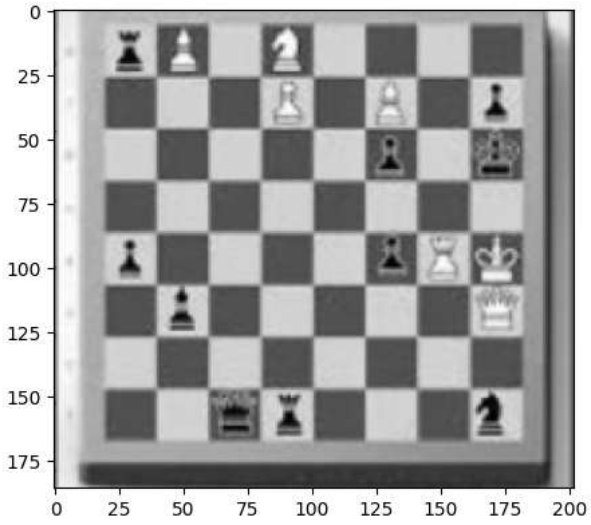
```
<matplotlib.image.AxesImage at 0x7bd49cc05990>
```



4. Hight pass frequency by using Gaussian smooth function

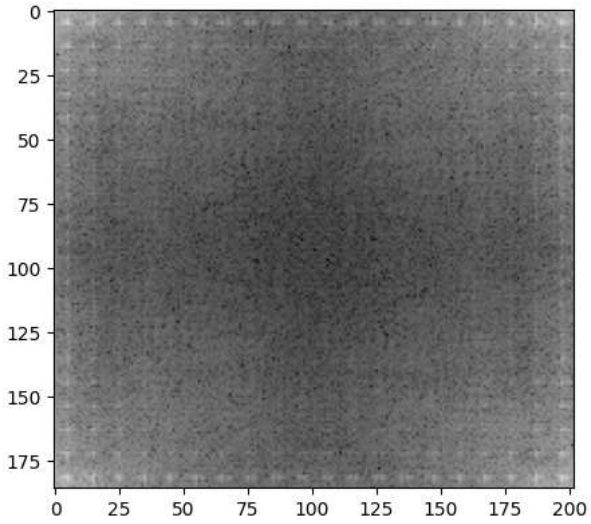

```
img = cv2.imread('/content/Ảnh chụp màn hình 2023-10-03 164636.png')

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.figure(figsize = (5, 5))
plt.imshow(gray, cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd49cc82ad0>
```



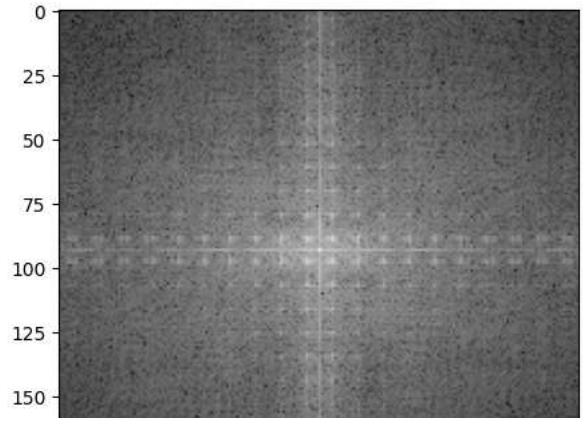

```
f = np.fft.fft2(gray)
plt.figure(figsize=(5,5))
plt.imshow(np.log(np.abs(f)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd49cb1c400>
```



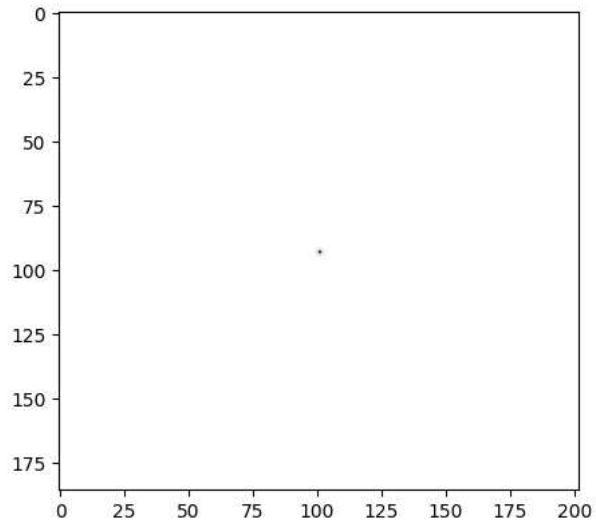

```
shifted_f = np.fft.fftshift(f)
plt.figure(figsize=(5,5))
plt.imshow(np.log(np.abs(shifted_f)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd49cb9d060>
```



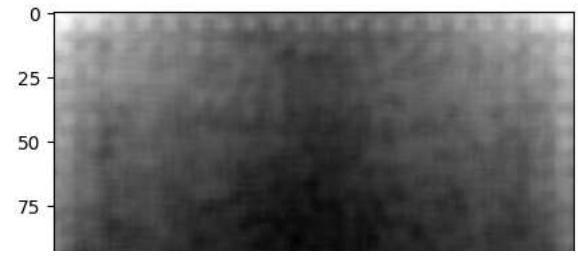

```
low_pass = np.ones(shape = gray.shape)

xc = gray.shape[1] // 2
yc = gray.shape[0] // 2

Do = 40

def d(x, y):
  return np.sqrt((x - xc)**2 + (y - yc)**2)

for x in range(gray.shape[1]):
  for y in range(gray.shape[0]):
    low_pass [y,x] = 1 - np.exp(-d(x,y))**2 / (2*sigma**2)

f = f * low_pass
plt.figure(figsize=(5,5))
plt.imshow(np.log(np.abs(low_pass)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd49c872740>
```



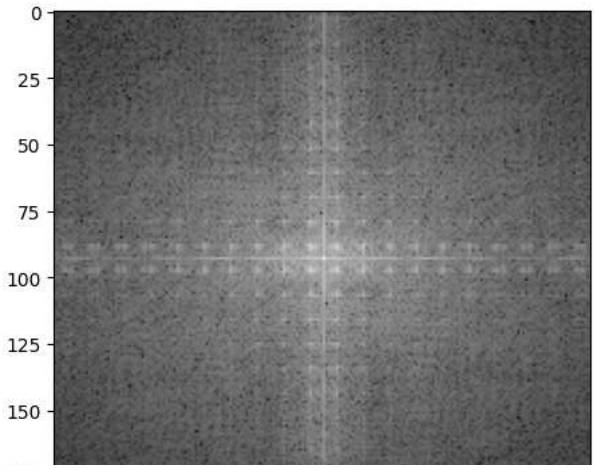

```
img_gau = cv2.GaussianBlur(np.log(np.abs(f)), (5, 5), 10)

plt.figure(figsize=(5,5))
plt.imshow(img_gau, cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd49caa7610>
```




```
g_4 = np.fft.ifftshift(f)
plt.imshow(np.log(np.abs(g_4)), cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7bd49c79b340>
```




```
new_img_4 = np.abs(np.fft.ifft2(g_4))
plt.figure(figsize = (6,6))
plt.imshow(new_img_4, cmap= 'gray')
plt.figure(figsize = (6,6))
plt.imshow(gray, cmap = 'gray')
```

<matplotlib.image.AxesImage at 0x7bd49c665e10>