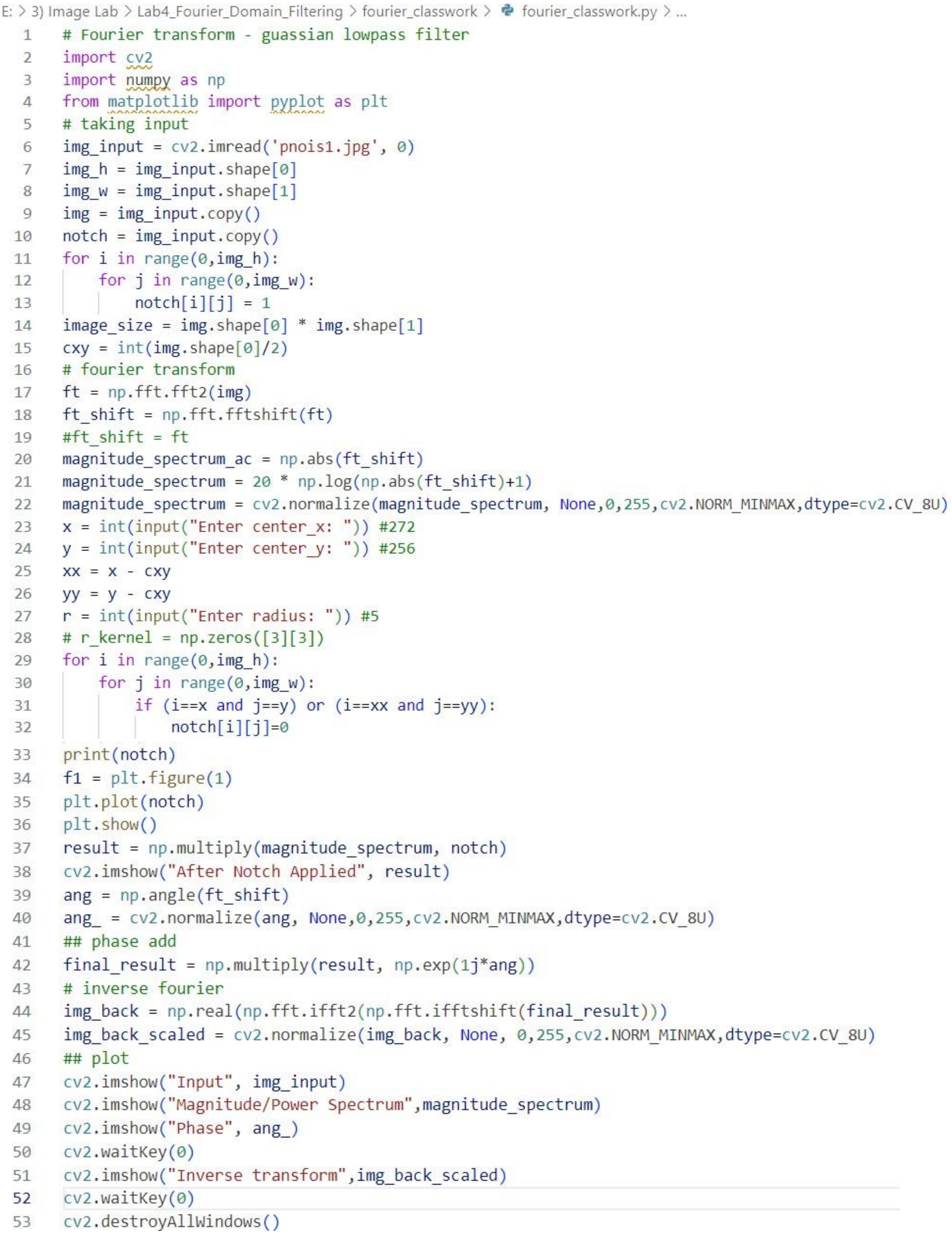
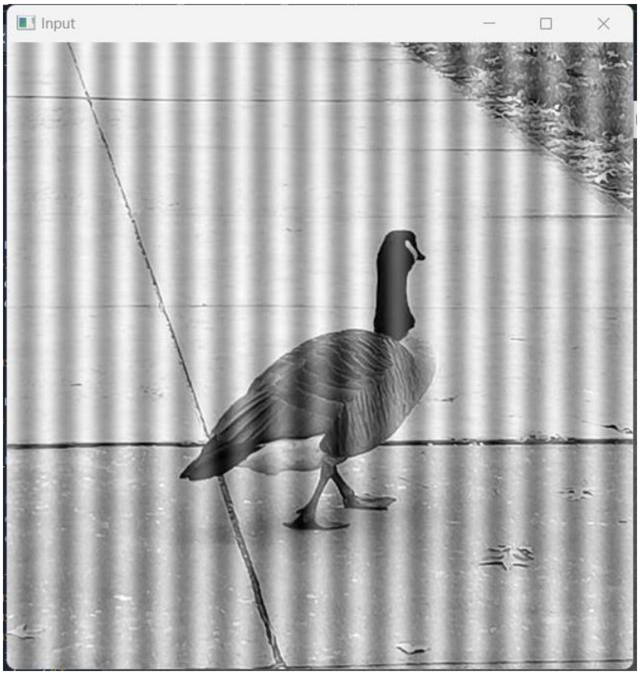
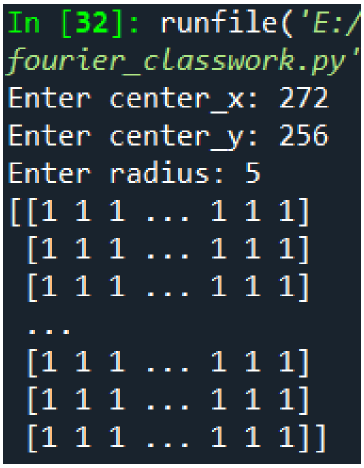


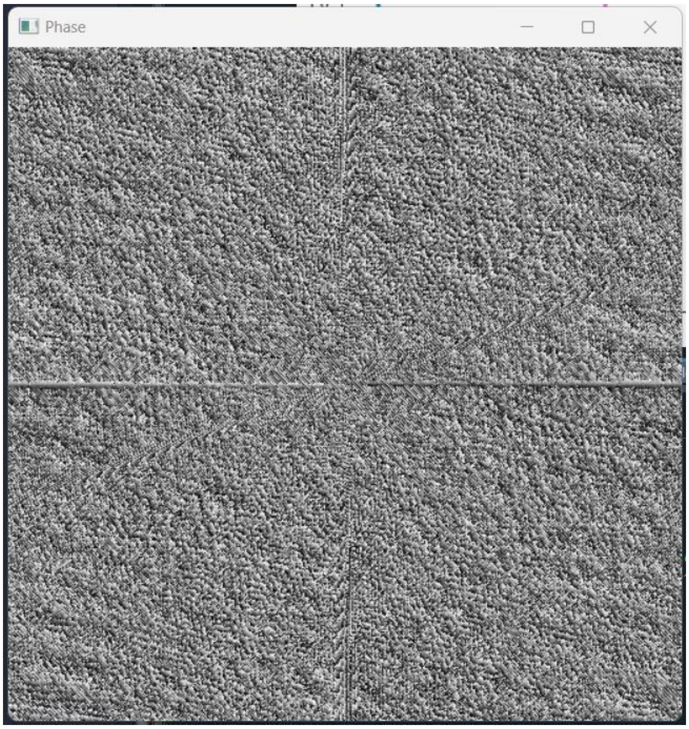
E: > 3) Image Lab > Lab4_Fourier_Domain_Filtering > fourier_classwork > fourier_classwork.py > ...

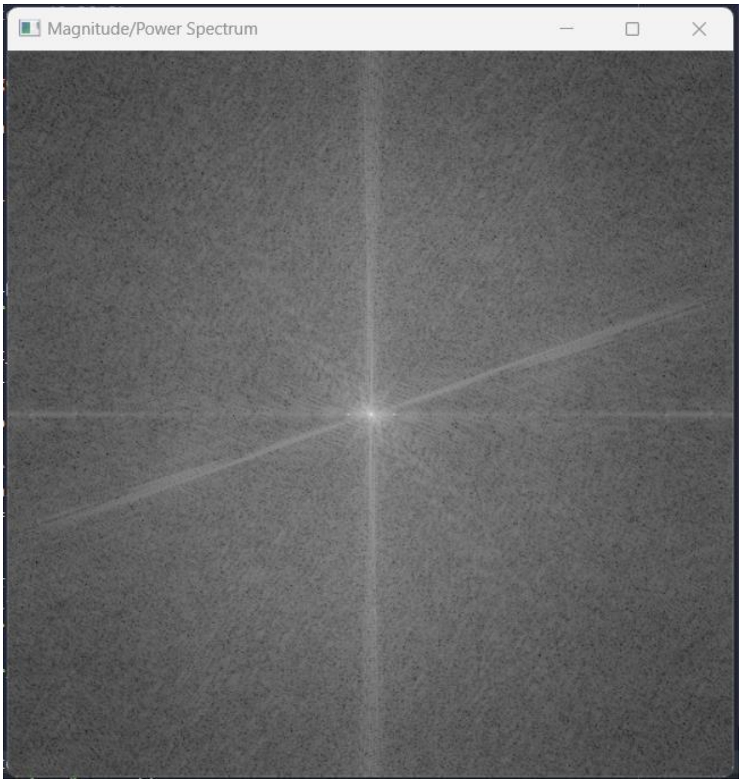
```python
# Fourier transform - guassian lowpass filter
import cv2
import numpy as np
from matplotlib import pyplot as plt
# taking input
img_input = cv2.imread('pnois1.jpg', 0)
img_h = img_input.shape[0]
img_w = img_input.shape[1]
img = img_input.copy()
notch = img_input.copy()
for i in range(0,img_h):
    for j in range(0,img_w):
        notch[i][j] = 1
image_size = img.shape[0] * img.shape[1]
cxy = int(img.shape[0]/2)
# fourier transform
ft = np.fft.fft2(img)
ft_shift = np.fft.fftshift(ft)
#ft_shift = ft
magnitude_spectrum_ac = np.abs(ft_shift)
magnitude_spectrum = 20 * np.log(np.abs(ft_shift)+1)
magnitude_spectrum = cv2.normalize(magnitude_spectrum, None,0,255,cv2.NORM_MINMAX,dtype=cv2.CV_8U)
x = int(input("Enter center_x: ")) #272
y = int(input("Enter center_y: ")) #256
xx = x - cxy
yy = y - cxy
r = int(input("Enter radius: ")) #5
# r_kernel = np.zeros([3][3])
for i in range(0,img_h):
    for j in range(0,img_w):
        if (i==x and j==y) or (i==xx and j==yy):
            notch[i][j]=0
print(notch)
f1 = plt.figure(1)
plt.plot(notch)
plt.show()
result = np.multiply(magnitude_spectrum, notch)
cv2.imshow("After Notch Applied", result)
ang = np.angle(ft_shift)
ang_ = cv2.normalize(ang, None,0,255,cv2.NORM_MINMAX,dtype=cv2.CV_8U)
## phase add
final_result = np.multiply(result, np.exp(1j*ang))
# inverse fourier
img_back = np.real(np.fft.ifft2(np.fft.ifftshift(final_result)))
img_back_scaled = cv2.normalize(img_back, None, 0,255,cv2.NORM_MINMAX,dtype=cv2.CV_8U)
## plot
cv2.imshow("Input", img_input)
cv2.imshow("Magnitude/Power Spectrum",magnitude_spectrum)
cv2.imshow("Phase", ang_)
cv2.waitKey(0)
cv2.imshow("Inverse transform",img_back_scaled)
cv2.waitKey(0)
cv2.destroyAllWindows()
```





```
In [32]: runfile('E:/
fourier_classwork.py'
Enter center_x: 272
Enter center_y: 256
Enter radius: 5
[[1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 ...
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]
 [1 1 1 ... 1 1 1]]
```




Magnitude/Power Spectrum