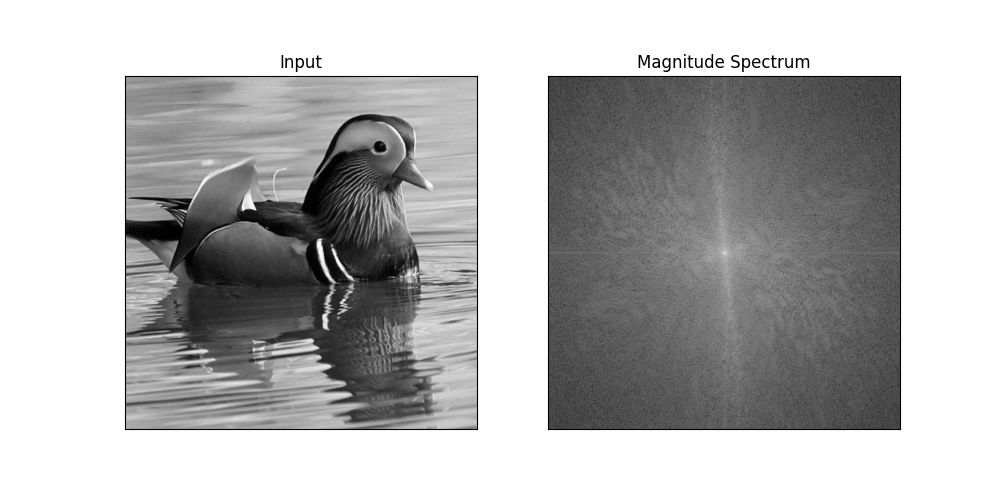
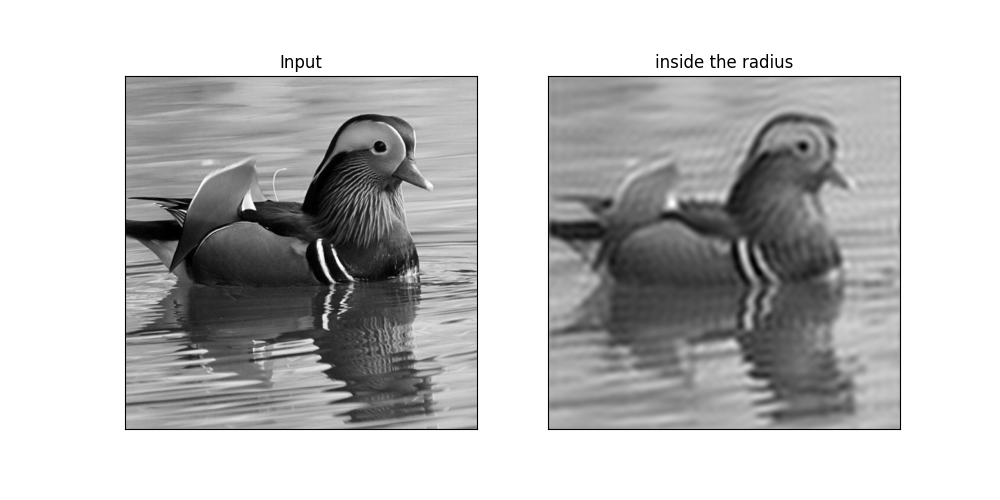
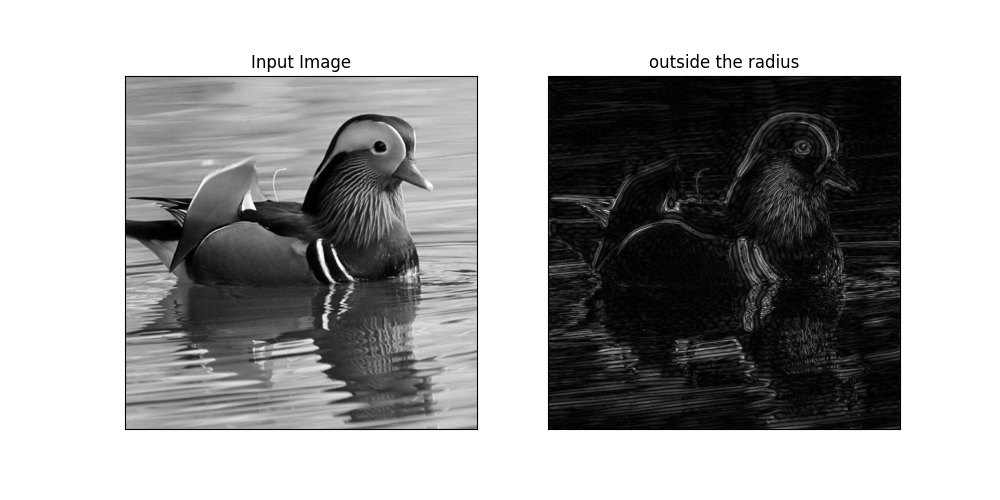
1. Plot of DFT magnitude in Log scale



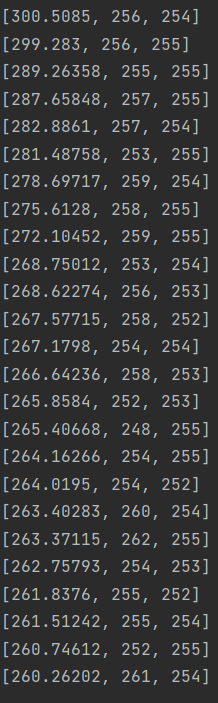
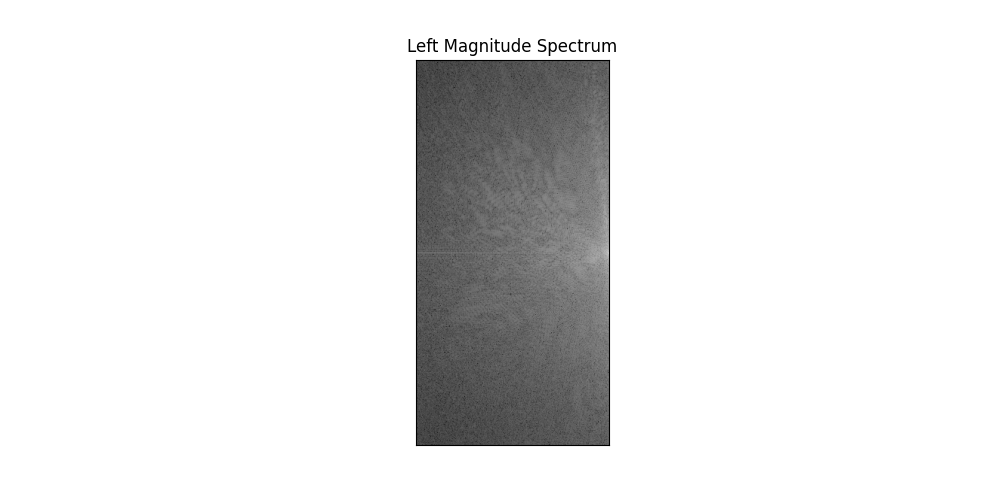
1. Image constructed by DFT coefficients **inside** the circular region with radius = 30



1. Image constructed by DFT coefficients **outside** the circular region with radius = 30



1. Table of top 25 DFT frequencies ( u v ) in the left half frequency region



CODE:

import matplotlib.pyplot as plt  
import numpy as np  
import cv2  
  
image = cv2.imread('Bird 2.tif',0) #以灰度模式讀取  
image\_float32 = np.float32(image) #這個dft要注意先將img轉化為float32的格式  
  
dft = cv2.dft(image\_float32, flags = cv2.DFT\_COMPLEX\_OUTPUT)  
dft\_shift = np.fft.fftshift(dft) #將低頻部分移動到影象中心  
fig, (ax1, ax2) = plt.subplots(figsize=(10, 5), nrows=1, ncols=2)  
ax1.imshow(image, cmap = 'gray')  
ax1.set\_title('Input')  
ax1.set\_xticks([])  
ax1.set\_yticks([])  
  
ax2.imshow(20\*np.log(cv2.magnitude(dft\_shift[:,:,0],dft\_shift[:,:,1])),cmap='gray')#強度光譜顯示  
ax2.set\_title('Magnitude Spectrum')  
ax2.set\_xticks([])  
ax2.set\_yticks([])  
  
plt.show()  
  
#inside the circular radius30  
  
rows, cols = image.shape  
crow, ccol = rows//2 , cols//2 # 長寬/2剛好在中間點  
# 做一個遮罩式除了中間-30~30是有數值以外其他全部歸零  
mask = np.zeros((rows, cols, 2), np.uint8)  
mask[crow-30:crow+31, ccol-30:ccol+31] = 1  
# 加上遮罩並反轉  
fshift = dft\_shift\*mask  
f\_ishift = np.fft.ifftshift(fshift)  
img\_back = cv2.idft(f\_ishift)  
  
fig, (ax1, ax2) = plt.subplots(figsize=(10, 5), nrows=1, ncols=2)  
ax1.imshow(image, cmap = 'gray')  
ax1.set\_title('Input')  
ax1.set\_xticks([])  
ax1.set\_yticks([])  
  
ax2.imshow(cv2.magnitude(img\_back[:,:,0],img\_back[:,:,1]),cmap='gray')  
ax2.set\_title('inside the radius')  
ax2.set\_xticks([])  
ax2.set\_yticks([])  
  
plt.show()  
  
#High pass filter(outside)  
  
rows, cols = image.shape  
crow, ccol = rows//2 , cols//2#中間點  
# create a mask first, center square is 0, remaining all ones  
mask = np.ones((rows, cols, 2), np.uint8)#全部保留  
mask[crow-30:crow+31, ccol-30:ccol+31] = 0#遮罩在-30~30是0  
# 加上遮罩並反轉  
fshift = dft\_shift\*mask  
f\_ishift = np.fft.ifftshift(fshift)  
img\_back = cv2.idft(f\_ishift)  
  
fig, (ax1, ax2) = plt.subplots(figsize=(10, 5), nrows=1, ncols=2)  
ax1.imshow(image, cmap = 'gray')  
ax1.set\_title('Input Image')  
ax1.set\_xticks([])  
ax1.set\_yticks([])  
  
ax2.imshow(cv2.magnitude(img\_back[:,:,0],img\_back[:,:,1]),cmap='gray')  
ax2.set\_title('outside the radius')  
ax2.set\_xticks([])  
ax2.set\_yticks([])  
  
plt.show()  
  
#顯示左半邊top25 DFT frequencies  
  
magnitude\_spectrum = 20\*np.log(cv2.magnitude(dft\_shift[:,:,0],dft\_shift[:,:,1])) # compute magnitude spectrum  
fig, (ax2) = plt.subplots(figsize=(10, 5))  
left=[]  
#print(size)  
# 裁切區域的 x 與 y 座標（左上角）  
x = 0  
y = 0  
# 裁切區域的長度與寬度  
w = 256  
h = 512  
# 裁切圖片  
cut = magnitude\_spectrum[y:y+h+1, x:x+w+1]  
  
#取圖片最大值  
for i in range(256):  
 for j in range(512):  
 d=[magnitude\_spectrum[j][i],j,i]  
 left.append(d)  
left.sort(reverse= True)#反向排序  
for i in range(25):#取TOP25  
 print(left[i])  
  
ax2.imshow(cut, cmap = 'gray')  
ax2.set\_title('Left Magnitude Spectrum')  
ax2.set\_xticks([])  
ax2.set\_yticks([])  
  
plt.show()