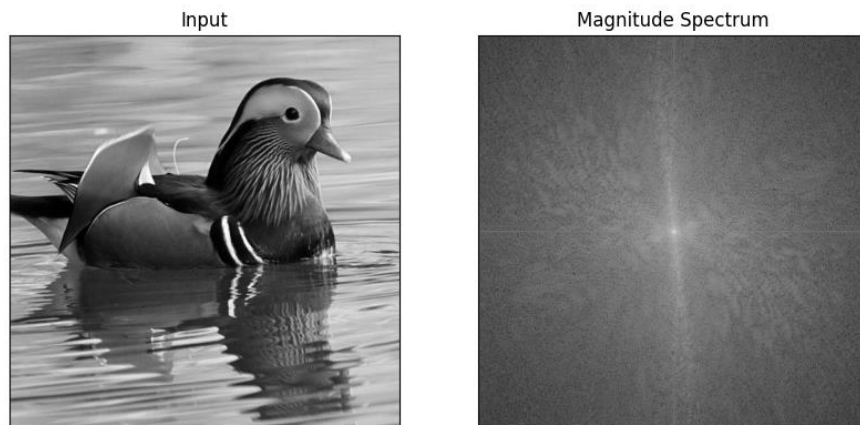
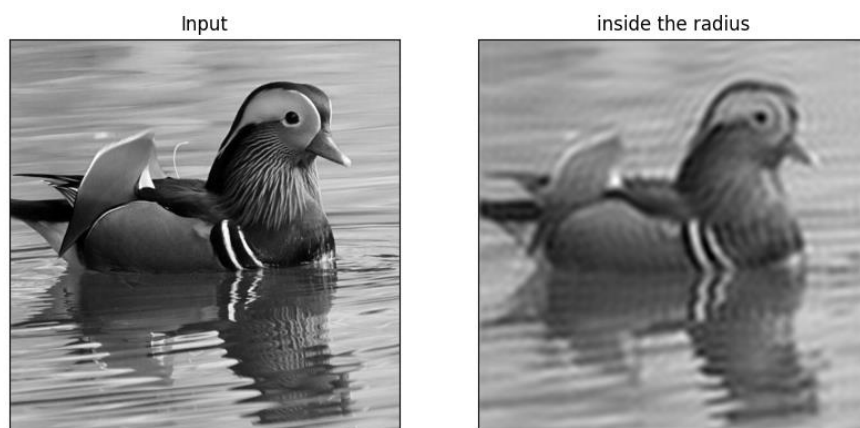


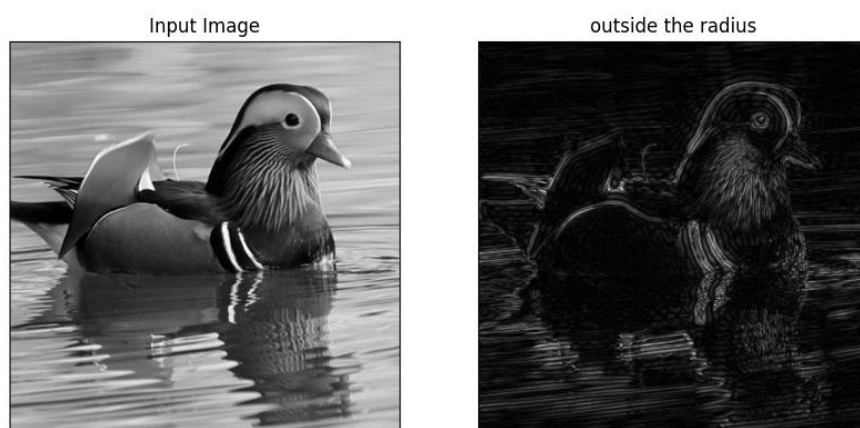
1. Plot of DFT magnitude in Log scale



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

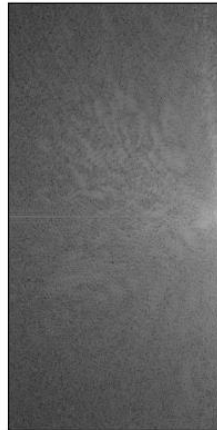


3. Image constructed by DFT coefficients **outside** the circular region with radius = 30



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

Left Magnitude Spectrum



```
[300.5085, 256, 254]
[299.283, 256, 255]
[289.26358, 255, 255]
[287.65848, 257, 255]
[282.8861, 257, 254]
[281.48758, 253, 255]
[278.69717, 259, 254]
[275.6128, 258, 255]
[272.10452, 259, 255]
[268.75012, 253, 254]
[268.62274, 256, 253]
[267.57715, 258, 252]
[267.1798, 254, 254]
[266.64236, 258, 253]
[265.8584, 252, 253]
[265.40668, 248, 255]
[264.16266, 254, 255]
[264.0195, 254, 252]
[263.40283, 260, 254]
[263.37115, 262, 255]
[262.75793, 254, 253]
[261.8376, 255, 252]
[261.51242, 255, 254]
[260.74612, 252, 255]
[260.26202, 261, 254]
```

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像中心
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()

```