Project 2

ID：311512040　　name：林胤宏

(a)code

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image,ImageOps
from openpyxl import load_workbook
```

```
def plot(pic):            #定義畫圖 function
    plt.figure(figsize=(5,5))
    plt.imshow(pic, cmap='gray')
    plt.axis('off')
    plt.show()
```

```
# open the image pic
pic = cv2.imread('./kid.tif',0)
plot(pic)
```

```
B = np.fft.fft2(pic)
Bshift = np.fft.fftshift(B)  #將低頻分量移動到頻域圖像的中心

plot(np.log1p(np.abs(B)))
plot(np.log1p(np.abs(Bshift)))
b = np.uint8(cv2.normalize(np.log1p(np.abs(Bshift)), None, alpha=0,beta=255,
norm_type=cv2.NORM_MINMAX))
b = Image.fromarray(b)
b.save('output/(b).tif',dpi=(150.0,150.0))
```

```
#將右邊及下方 padding
def padding(img,M,N):
    pad_img = np.zeros((M,N))
    for m in range(600):
        for n in range(600):
            pad_img[m,n] = pic[m,n]
    return pad_img
# Create Gaussin Filter: Low Pass Filter
M,N = 1200,1200
L = np.zeros((M,N), dtype=np.float32)
D0 = 200
for u in range(M):
    for v in range(N):
        D = np.sqrt((u-M/2)**2 + (v-N/2)**2)
        L[u,v] = np.exp(-D**2/(2*D0*D0))
print(L.shape)
```

```
plot(L)
b = np.uint8(cv2.normalize(L, None, alpha=0,beta=255, norm_type=cv2.NORM_MINMAX))
b = Image.fromarray(b)
b.save('output/(c)L.tif',dpi=(150.0,150.0))
```

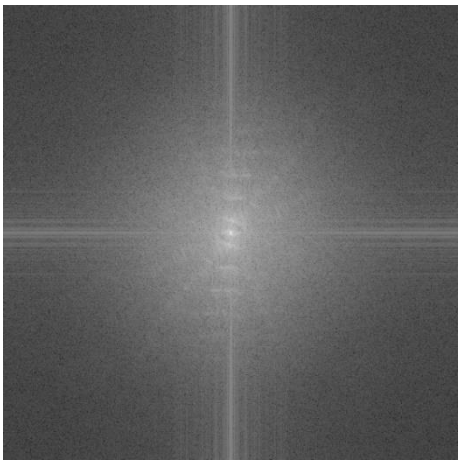
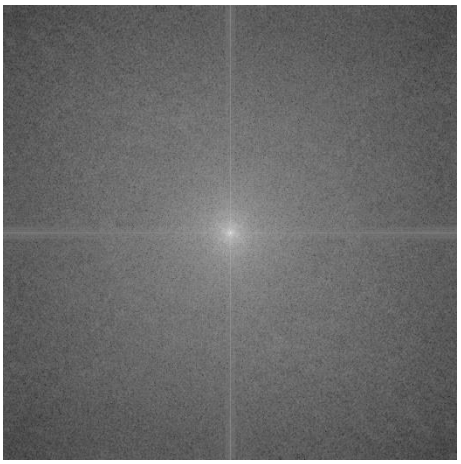
```
# Image Filters
F = np.fft.fft2(pad_img)
Fshift = np.fft.fftshift(F)
Gshift = Fshift * L
G = np.fft.ifftshift(Gshift)  # 將低頻逆轉換回圖像四角
g = np.abs(np.fft.ifft2(G))

plot(g)
plot(g[0:600,0:600])
b = np.uint8(cv2.normalize(g[0:600,0:600], None, alpha=0,beta=255,
norm_type=cv2.NORM_MINMAX))
b = Image.fromarray(b)
b.save('output/(d)GLPF.tif',dpi=(150.0,150.0))
plot(np.log1p(np.abs(Gshift)))
plot(np.log1p(np.abs(G)))
```

```
# Gaussian: High pass filter
HPF = 1 - L

plot(HPF)
b = np.uint8(cv2.normalize(HPF, None, alpha=0,beta=255, norm_type=cv2.NORM_MINMAX))
b = Image.fromarray(b)
b.save('output/(c)H.tif',dpi=(150.0,150.0))
```

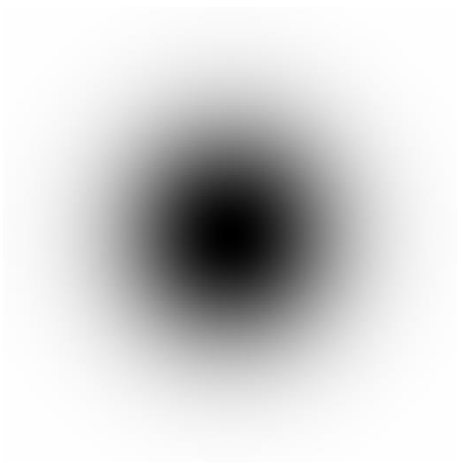
```
# Image Filters
Gshift = Fshift * HPF
G = np.fft.ifftshift(Gshift)
g = np.abs(np.fft.ifft2(G))

plot(g)
plot(g[0:600,0:600])
b = np.uint8(cv2.normalize(g[0:600,0:600], None, alpha=0,beta=255,
norm_type=cv2.NORM_MINMAX))
b = Image.fromarray(b)
b.save('output/(d)GHPF.tif',dpi=(150.0,150.0))

plot(np.log1p(np.abs(Gshift)))
plot(np.log1p(np.abs(G)))
```

```
# 找出前 25 大的值
B = np.log1p(np.abs(Bshift))
C = B[0:600,0:300]         #(0<=u<=M-1, 0<=N/2-1)
# print(C.shape)
wb = load_workbook("freq.xlsx")
print(wb.sheetnames)
sheet = wb.worksheets[0]  #抓出列數

for i in range(25):
    max_index = np.unravel_index(np.argmax(C, axis=None), C.shape)
    print(max_index,C[max_index])
    sheet.cell(row = i+2, column = 1, value = max_index[0])
    sheet.cell(row = i+2, column = 2, value = max_index[1])
    sheet.cell(row = i+2, column = 3, value = C[max_index])
    C[max_index] = 0
wb.save("freq.xlsx")
```

(b) Fourier magnitude spectra (in Log scale) of kid and fruit

kid	fruit
	

c) Magnitude responses of Gaussian LPF and HPF

(d) 4 output images

Kid LPF, HPF

Furit LPF, HPF

(e) start from left top

Kid

u	v
301	299
300	299
299	299
298	299
297	299
299	297
302	298
298	298
298	294
302	299
302	296
299	298
304	298
316	298
299	294
301	296
317	298
296	296
296	298
316	297
300	294
298	292
297	296
298	297
301	297

Fruit

u	v
300	299
301	297
300	298
296	299
303	297
300	297
299	299
295	299
302	297
297	298
301	294
298	299
300	295
302	299
304	299
303	299
296	294
299	298
303	298
299	296
296	296
306	299
297	296
299	297
302	295