Project2

studentID : 311512040　　name : 林胤宏

(a) Source Codes

```
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)))
```

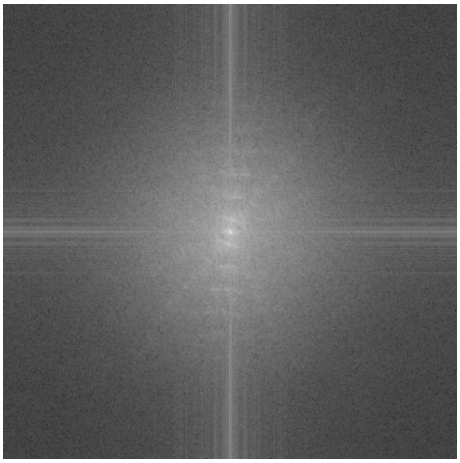
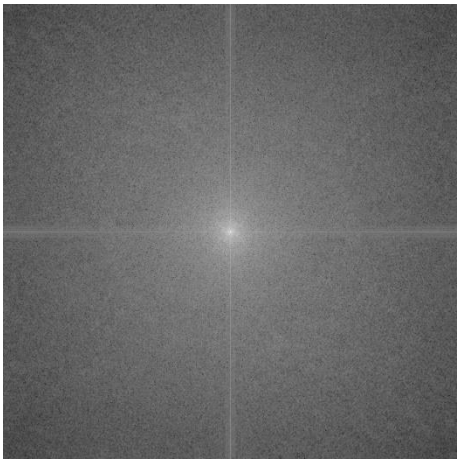
```
# Gaussian: High pass filter
HPF = 1 - L

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))
```

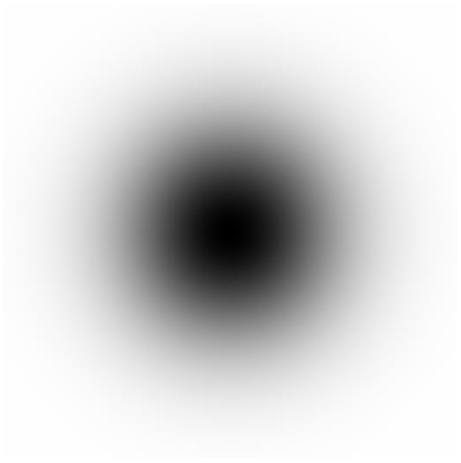
```
# Image Filters
Gshift = Fshift * HPF
G = np.fft.ifftshift(Gshift)
g = np.abs(np.fft.ifft2(G))

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)))
```

```
v = np.array((25,2))
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

max_freq = []
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")
```

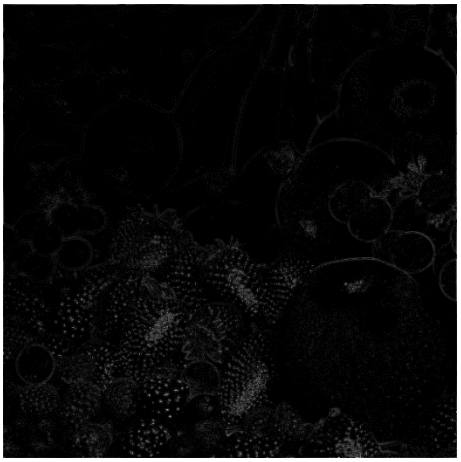
(b) Fourier magnitude spectra

kid	fruit
	

(C)Magnitude responses of Gaussian LPF and HPF

LPF	HPF
	

(d) 4 output images

Kid GLPF  \	Kid GHPF 
Fruit GLPF 	Fruit GHPF 

(e)Tables of top 25 DFT frequencies (u,v) of b

kid

u	v	frequency
301	299	15.39528
300	299	15.27479
299	299	15.13128
298	299	14.78255
297	299	14.76084
299	297	14.71626
302	298	14.67134
298	298	14.265
298	294	14.24574
302	299	14.19937
302	296	14.17446
299	298	14.14705
304	298	14.11478
316	298	14.10889
299	294	14.05498
301	296	14.03818
317	298	14.02321
296	296	13.98458
296	298	13.97809
316	297	13.95948
300	294	13.93995
298	292	13.89982
297	296	13.88896
298	297	13.88214
301	297	13.87699

fruit

u	v	frequency
300	299	15.35358
301	297	14.75635
300	298	14.72176
296	299	14.48491
303	297	14.46972
300	297	14.44892
299	299	14.44428
295	299	14.43025
302	297	14.40497
297	298	14.34304
301	294	14.32425
298	299	14.30921
300	295	14.28946
302	299	14.25971
304	299	14.24107
303	299	14.18918
296	294	14.13862
299	298	14.08459
303	298	14.08444
299	296	13.96281
296	296	13.9347
306	299	13.90292
297	296	13.87225
299	297	13.82528
302	295	13.79231