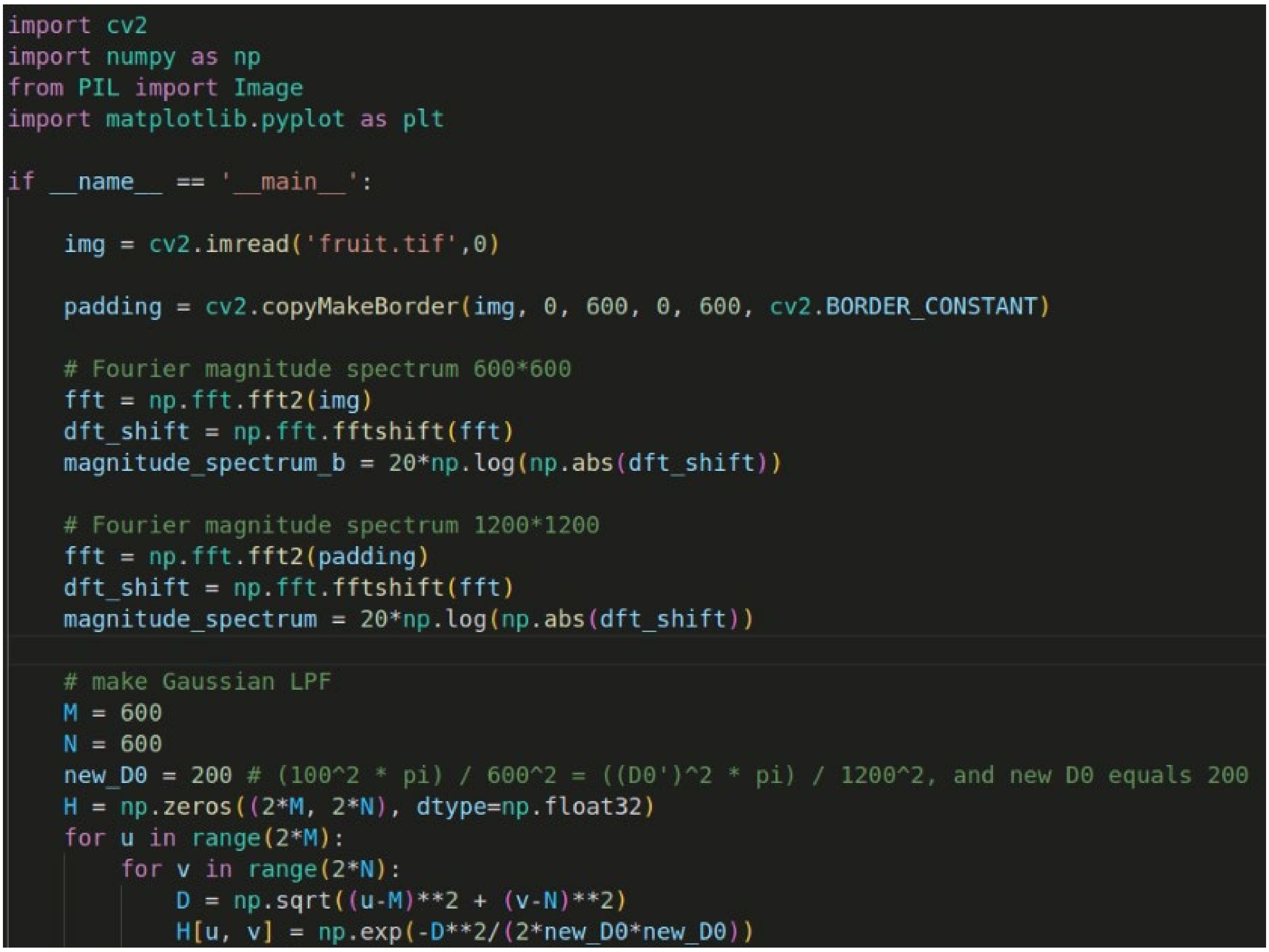
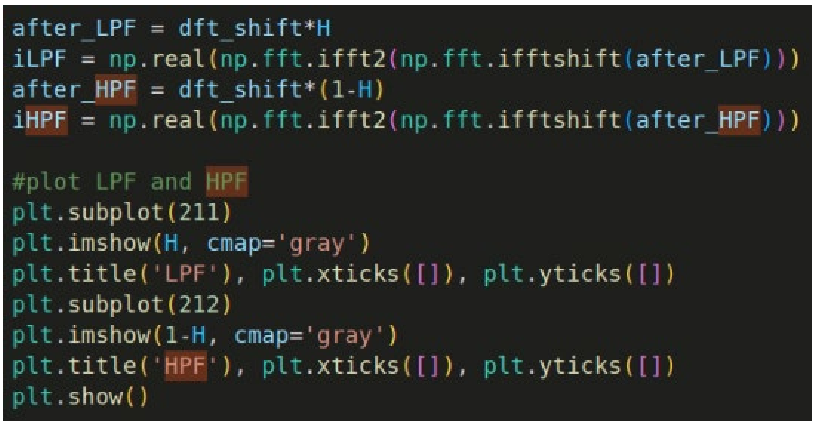
Project 2

ID：311512015　　　name：謝元碩

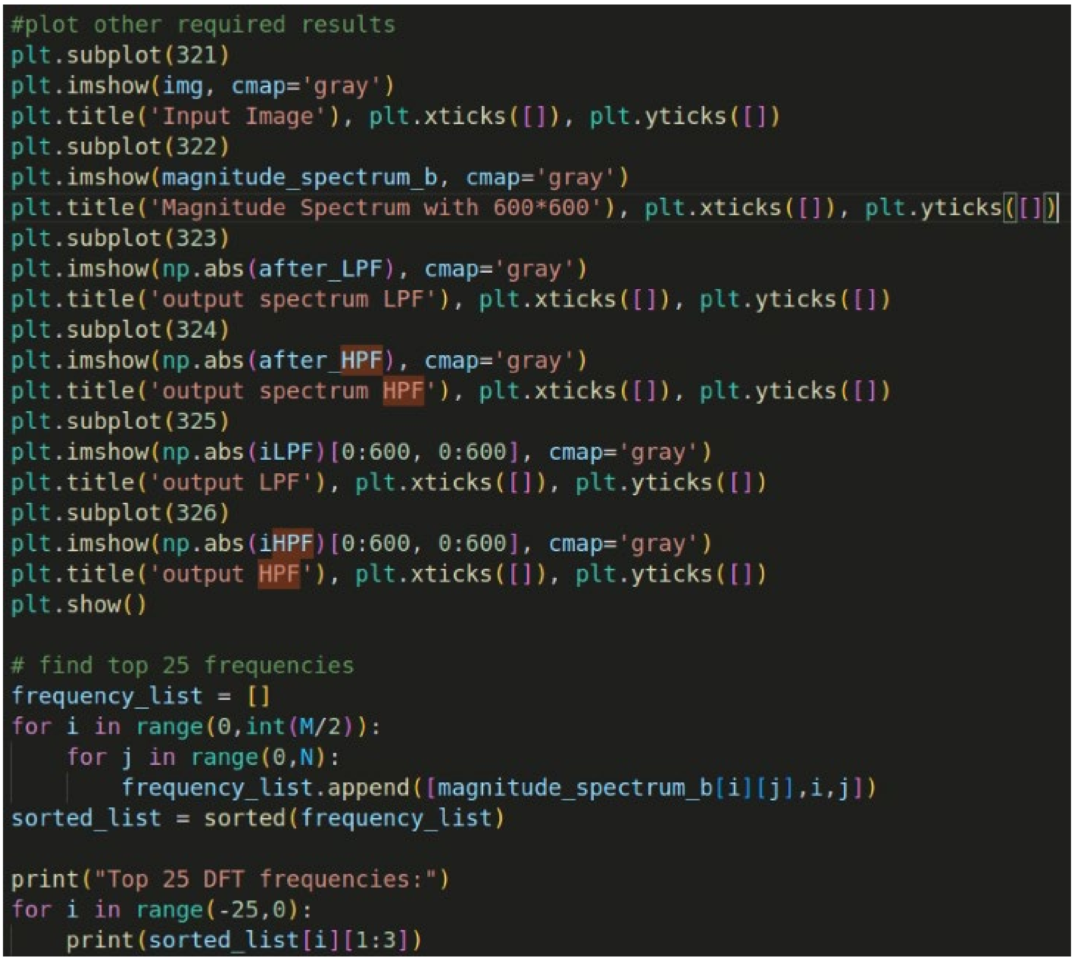
(a)code



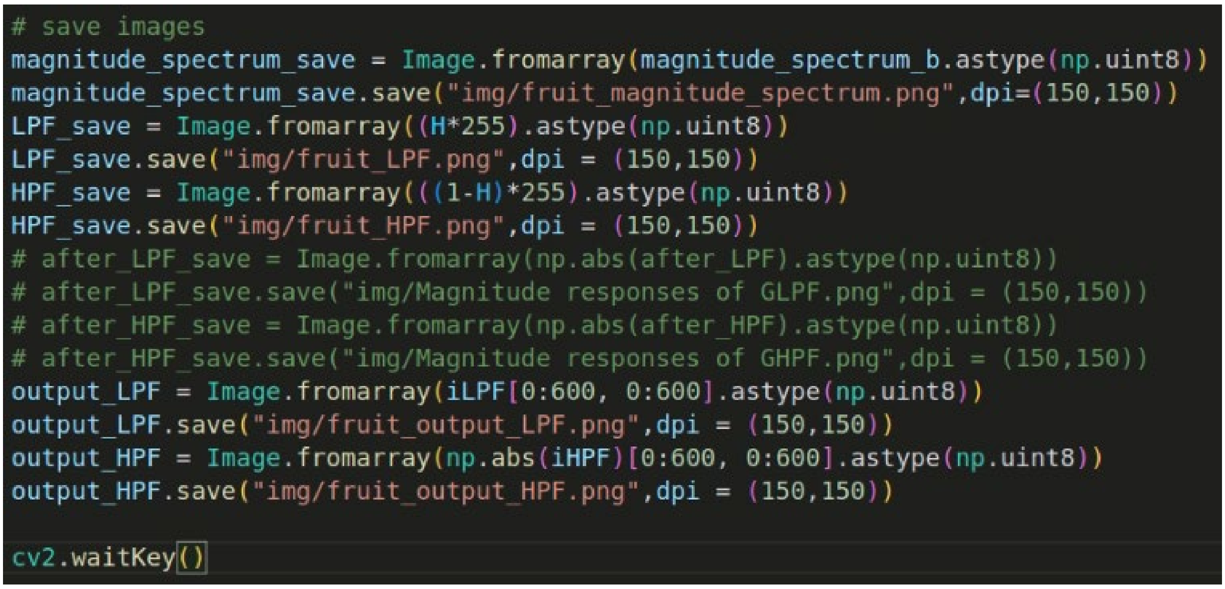
```
import cv2
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

if __name__ == '__main__':

    img = cv2.imread('fruit.tif',0)

    padding = cv2.copyMakeBorder(img, 0, 600, 0, 600, cv2.BORDER_CONSTANT)

    # Fourier magnitude spectrum 600*600
    fft = np.fft.fft2(img)
    dft_shift = np.fft.fftshift(fft)
    magnitude_spectrum_b = 20*np.log(np.abs(dft_shift))

    # Fourier magnitude spectrum 1200*1200
    fft = np.fft.fft2(padding)
    dft_shift = np.fft.fftshift(fft)
    magnitude_spectrum = 20*np.log(np.abs(dft_shift))

    # make Gaussian LPF
    M = 600
    N = 600
    new_D0 = 200 # (100^2 * pi) / 600^2 = ((D0')^2 * pi) / 1200^2, and new D0 equals 200
    H = np.zeros((2*M, 2*N), dtype=np.float32)
    for u in range(2*M):
        for v in range(2*N):
            D = np.sqrt((u-M)**2 + (v-N)**2)
            H[u, v] = np.exp(-D**2/(2*new_D0*new_D0))
```



```
after_LPF = dft_shift*H
iLPF = np.real(np.fft.ifft2(np.fft.ifftshift(after_LPF)))
after_HPF = dft_shift*(1-H)
iHPF = np.real(np.fft.ifft2(np.fft.ifftshift(after_HPF)))

#plot LPF and HPF
plt.subplot(211)
plt.imshow(H, cmap='gray')
plt.title('LPF'), plt.xticks([]), plt.yticks([])
plt.subplot(212)
plt.imshow(1-H, cmap='gray')
plt.title('HPF'), plt.xticks([]), plt.yticks([])
plt.show()
```

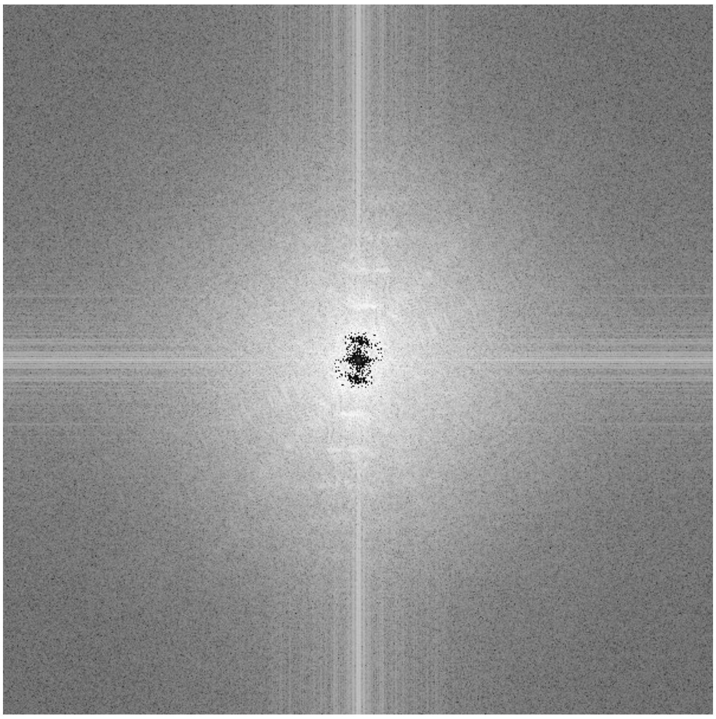
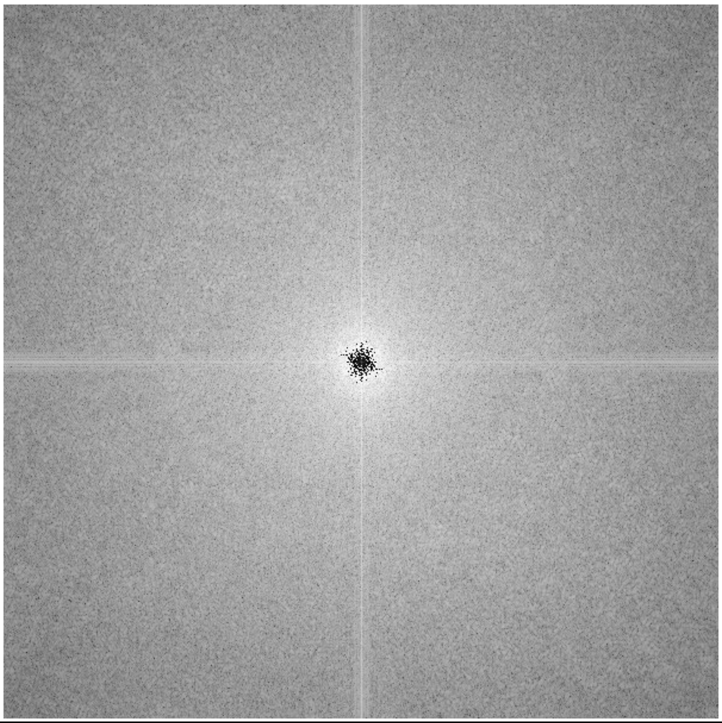


```
#plot other required results
plt.subplot(321)
plt.imshow(img, cmap='gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(322)
plt.imshow(magnitude_spectrum_b, cmap='gray')
plt.title('Magnitude Spectrum with 600*600'), plt.xticks([]), plt.yticks([])
plt.subplot(323)
plt.imshow(np.abs(after_LPF), cmap='gray')
plt.title('output spectrum LPF'), plt.xticks([]), plt.yticks([])
plt.subplot(324)
plt.imshow(np.abs(after_HPF), cmap='gray')
plt.title('output spectrum HPF'), plt.xticks([]), plt.yticks([])
plt.subplot(325)
plt.imshow(np.abs(iLPF)[0:600, 0:600], cmap='gray')
plt.title('output LPF'), plt.xticks([]), plt.yticks([])
plt.subplot(326)
plt.imshow(np.abs(iHPF)[0:600, 0:600], cmap='gray')
plt.title('output HPF'), plt.xticks([]), plt.yticks([])
plt.show()

# find top 25 frequencies
frequency_list = []
for i in range(0,int(M/2)):
    for j in range(0,N):
        frequency_list.append([magnitude_spectrum_b[i][j],i,j])
sorted_list = sorted(frequency_list)

print("Top 25 DFT frequencies:")
for i in range(-25,0):
    print(sorted_list[i][1:3])
```

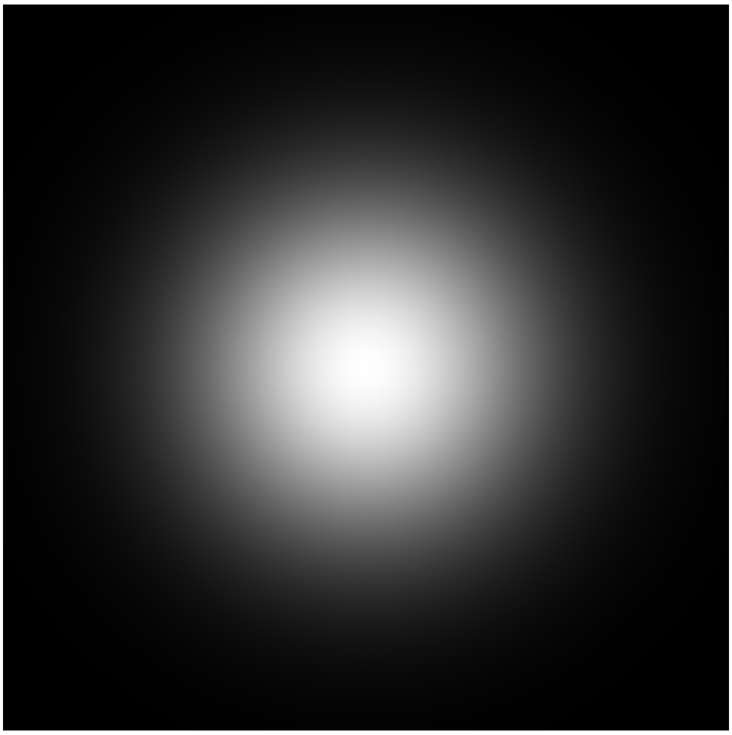
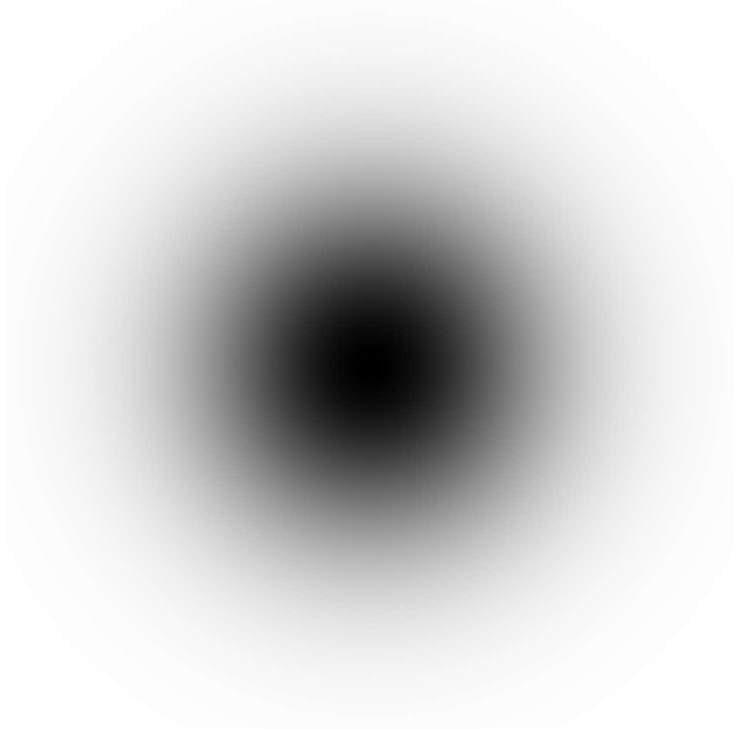


```
# save images
magnitude_spectrum_save = Image.fromarray(magnitude_spectrum_b.astype(np.uint8))
magnitude_spectrum_save.save("img/fruit_magnitude_spectrum.png",dpi=(150,150))
LPF_save = Image.fromarray((H*255).astype(np.uint8))
LPF_save.save("img/fruit_LPF.png",dpi = (150,150))
HPF_save = Image.fromarray(((1-H)*255).astype(np.uint8))
HPF_save.save("img/fruit_HPF.png",dpi = (150,150))
# after_LPF_save = Image.fromarray(np.abs(after_LPF).astype(np.uint8))
# after_LPF_save.save("img/Magnitude responses of GLPF.png",dpi = (150,150))
# after_HPF_save = Image.fromarray(np.abs(after_HPF).astype(np.uint8))
# after_HPF_save.save("img/Magnitude responses of GHPF.png",dpi = (150,150))
output_LPF = Image.fromarray(iLPF[0:600, 0:600].astype(np.uint8))
output_LPF.save("img/fruit_output_LPF.png",dpi = (150,150))
output_HPF = Image.fromarray(np.abs(iHPF)[0:600, 0:600].astype(np.uint8))
output_HPF.save("img/fruit_output_HPF.png",dpi = (150,150))

cv2.waitKey()
```

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

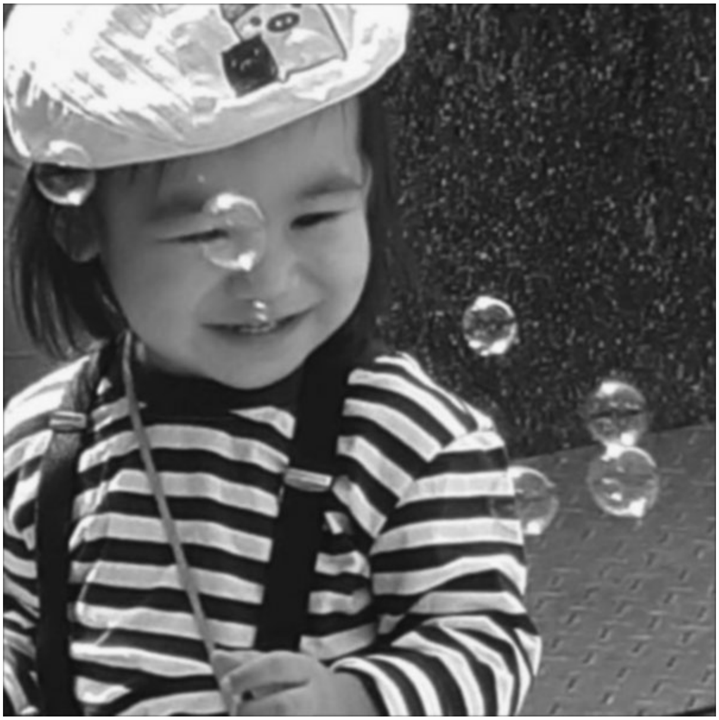
kid	fruit
	

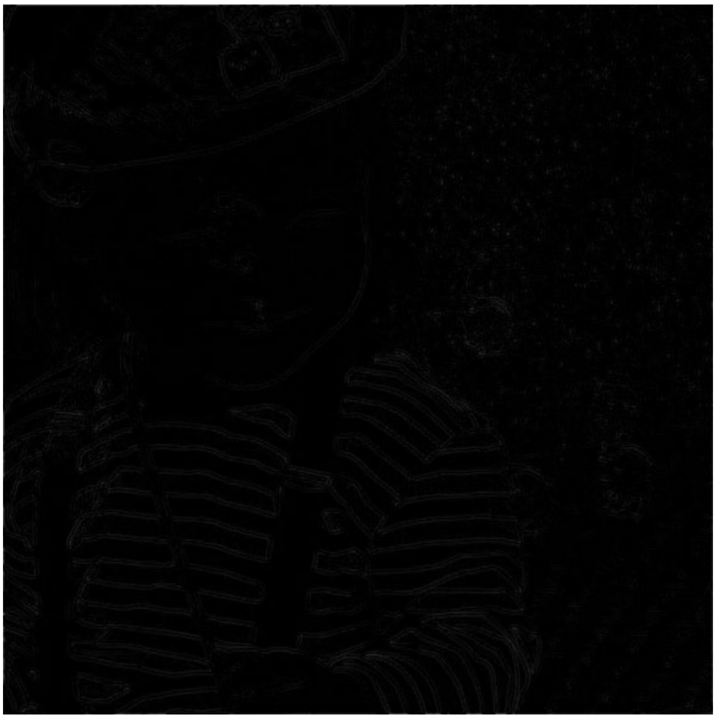
c) Magnitude responses of Gaussian LPF and HPF

LPF	HPF
	

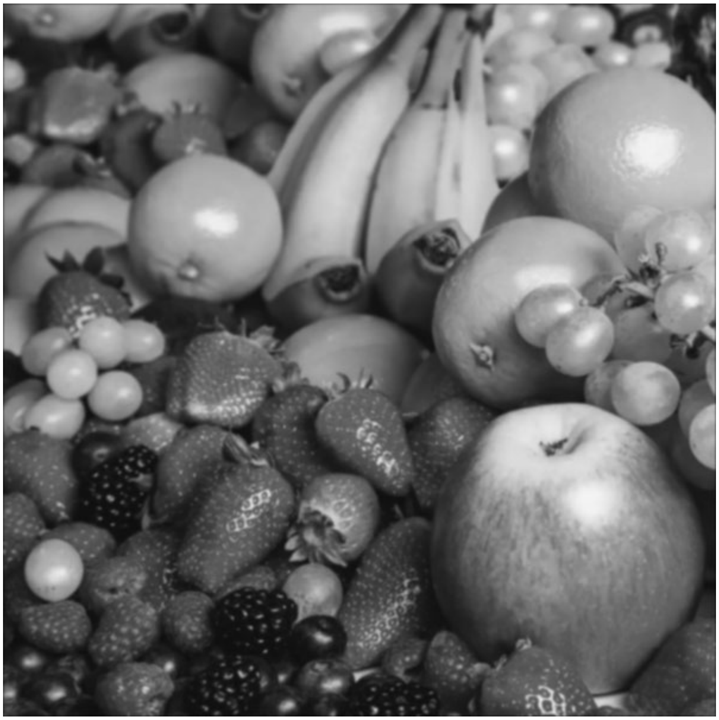
(d) 4 output images

Kid LPF, HPF





Fruit LPF, HPF



(e) start from left top

※由上至下，頻率由低到高

Kid	Fruit
[297, 296]	[296, 298]
[298, 292]	[298, 305]
[283, 300]	[299, 297]
[284, 303]	[297, 296]
[296, 298]	[294, 301]
[296, 296]	[296, 296]
[283, 302]	[299, 296]
[299, 304]	[297, 302]
[299, 294]	[299, 298]
[297, 300]	[296, 294]
[284, 302]	[297, 301]
[296, 302]	[296, 301]
[299, 298]	[298, 301]
[298, 304]	[298, 299]
[298, 301]	[296, 300]
[298, 294]	[299, 306]
[298, 298]	[297, 298]
[298, 302]	[298, 303]
[299, 297]	[295, 299]
[297, 299]	[299, 299]
[298, 300]	[297, 303]
[298, 299]	[296, 299]
[299, 299]	[299, 303]
[299, 300]	[298, 300]
[299, 301]	[299, 300]