

## 作業 4

1091402 顏心妤

### 題目敘述：

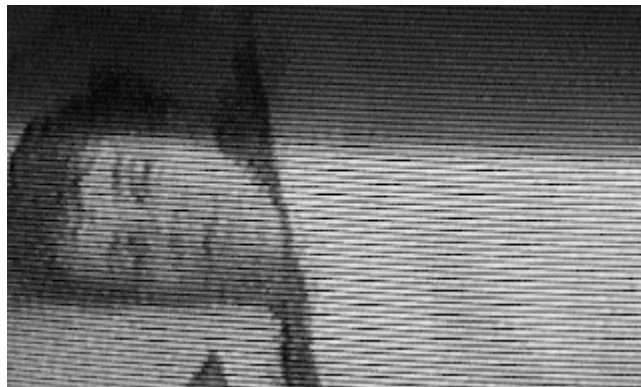
#### 影像還原練習

附件中的 image4 似乎受到某種頻域雜訊干擾，撰寫一個程式嘗試復原此圖像 (將圖中雜訊去除)。

### 開發環境：

Windows10、Spyder (Python 3.8)、OpenCV 4.7.0

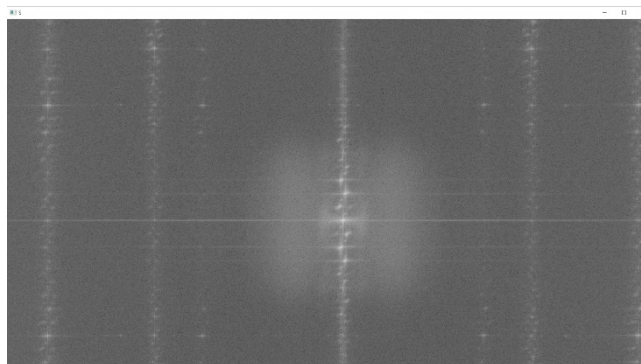
### 說明：



(原圖)

前半部分(至 27 行)為作業 3 的內容

延用作業 3 得到的頻譜圖，可以看到六個明顯的白點，即雜訊(峰值較高)



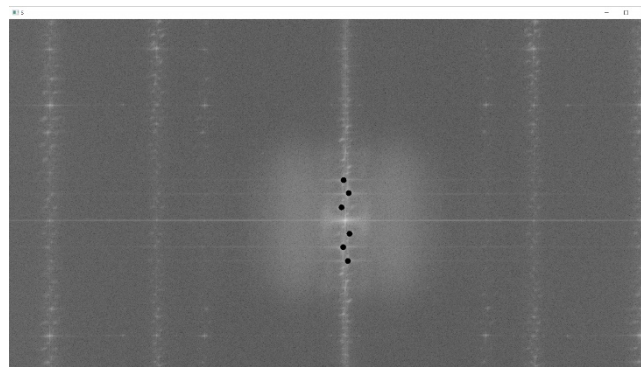
(頻譜圖)

利用 notch filter 去除雜訊

用滑鼠點選雜訊部分，並用黑點遮蓋，結束後按下 **esc** 離開繪圖。

```
#draw circle
def show_xy(event,x,y,flags,param):
    if event==1: #mouse click
        cv2.circle(S, (x,y), 7, (0,0,0), -1) #img,coordinate,radius,color,Solid
        cv2.imshow('S',S)
cv2.setMouseCallback('S', show_xy)

while(1):
    key=cv2.waitKey(0)
    if key==27:
        break
cv2.destroyAllWindows()
```



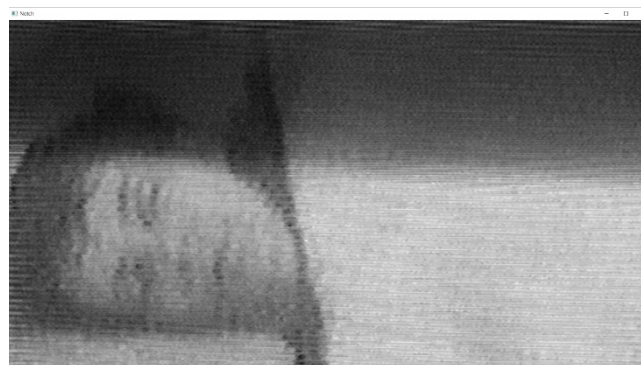
將修改後的頻譜圖乘上原本傅立葉轉換後的值

然後進行 Inverse Fourier Transform，得到過濾後圖像

經過 `magnitude()` 計算幅值、`normalize()` 標準化，讓圖像以灰階 256 色圖像方式呈現，並轉成 `uint8` 型態(避免負數)

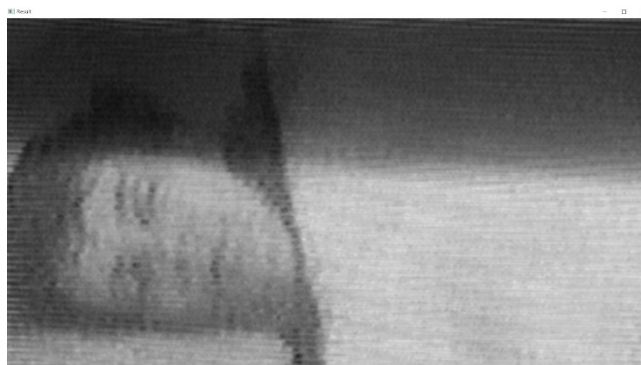
呈現出圖像。

```
S = np.expand_dims(S, axis=2) #讓S跟fshift形狀一樣(原本一個二維一個三維)
fshift=DFT*S
idft=cv2.idft(fshift, flags=cv2.DFT_COMPLEX_OUTPUT)
planes2 = cv2.split(idft)
idft = cv2.magnitude(planes2[0], planes2[1])
idft = cv2.normalize(idft, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8)
```



(經 notch filter 後的圖)

最後再用高斯模糊處理其他雜訊



(高斯模糊後的圖)