影像視訊處理實驗作業結報

班　　級	電機	學　　號	M11107S08	姓　　名	王維澤
實驗題目	去雜訊干擾				

實驗內容　　*請勿抄襲，否則視為未交

一、　實驗簡介(至少 200 字)：

若圖片本身有雜訊，代表像素資料失真，此時，我們只能想辦法從圖像的其他像素中取得資訊，儘可能去模擬出接近人眼，或者人眼所無法察覺的像素值。如果是選擇鄰近 NxN 像素加總後平均（N 通常為奇數），這種方式稱為均值濾波（Mean filter），優點是實作簡單、運算速度快，既然是取平均，表示鄰近像素的差異不能太大，因而適用於去除低頻雜訊，然而，面對椒鹽雜訊時，均值濾波就沒什麼效果，而且取平均值的結果就是，圖像邊緣也會跟著模糊，N 越大就越模糊。

如果取鄰近像素值排序後取中間數，稱為中值濾波（Median filter），若鄰近像素數量選擇適當，中值濾波可以有效地處理椒鹽雜訊（或者圖片中隨機散落的雜點），也能夠保留較多邊緣的細節，不過，鄰近像素取得越多，細節保留的效果會越差。

談到均值濾波的原理，是單純取鄰近像素的平均值，高斯濾波的原理也是取鄰近像素平均值，然而，計算平均值時會考慮鄰近像素的權重，權重是透過二維的高斯函式來計算，也就是考量了一定範圍內的雜訊會具有高斯分布。而實際環境中，不少雜訊分布就具有此特性，因此高斯濾波常用來去除照片中的雜訊。

二、　實驗動機及其解決方法(至少 500 字)：

均值濾波(Average filter)

透過觀察高斯分佈的曲線可以發現一個重點，雖然高斯雜訊附加上去的值每次都是隨機產生的，但是它的期望值在 0！最簡單的方式我們可以透過做平均來將它消除掉。

中值濾波(Medium filter)

一般來說，影像中每個像數點（pixel）跟周邊的鄰點是存在一定相依關係的，即灰階值會很接近。所以這時候我們利用中值濾波器，透過將周邊的資料撈進來做排序，然後用排序後正中間的那個值做輸出，這樣就能解決問題了！

高斯濾波(Gaussian filter)

高斯濾波實質上是一種信號的濾波器，其用途是信號的平滑處理，人們知道數字圖像用於後期應用，其噪聲是最大的問題，由於誤差會累計傳遞等原因，很多圖像處理教材會在很早的時候介紹 Gauss 濾波器，用於得到信噪比 SNR 較高的圖像（反應真實信號）。與此相關的有 Gauss-Laplace 變換，其實就是為了得到較好的圖像邊緣，先對圖像做 Gauss 平滑濾波，剔除噪聲，然後求二階導矢，用二階導的過零點確定邊緣，在計算時也是頻域乘積至空域卷積。

形態學(Morphology)

Morphology 這項工具在影像處理裡面可以說是非常的強大！到目前為止介紹過的影像處理方法裡面，你幾乎都能找到其他的方法來取代去做到類似的效果。

開啟運算 Opening：

就是先做 Erosion 再做 Dilation。其作用等同是你拿 Structures Element 在 Target 的內部任意移動，Structures Element 如果到不了的地方就會被消除掉，可以將圖形凸出的銳角給鈍化。

關閉運算 Closing：

接著把 Closing 拿來跟 Opening 比你會發現，其實就是反過來做而已。變成先 Dilation 再 Erosion，等同你拿 Structures Element 在 Target 的外部移動，進不去的地方就把它填滿，可以將圖形內陷的銳角給鈍化。

三、 程式碼(須附註解說明，截圖即可)：

```
def Average_Blurring(img):
    ksize=3
    img_blur = cv.blur(img,(ksize,ksize))
    res = np.hstack([img,img_blur])
    cv_imshow(res)
Average_Blurring(img_gray)
```

```
def Medium_Blurring(img):
    ksize = 3
    img_median = cv.medianBlur(img,ksize)
    res = np.hstack([img,img_median])
    cv_imshow(res)
Medium_Blurring(img_gray)
```

```python
def Gaussian_Blurring(img):
    ksize = 3
    img_gaussianBlur = cv.GaussianBlur(img,(ksize,ksize),0)
    res = np.hstack([img,img_gaussianBlur])
    cv_imshow(res)
Gaussian_Blurring(img_gray)
```

Rectangular Kernel

```python
kernel_rectangular = cv.getStructuringElement(cv.MORPH_RECT,(5,5))
print(kernel_rectangular)
print(type(kernel_rectangular))
```

```
[[1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]]
<class 'numpy.ndarray'>
```

Elliptical Kernel

```python
kernel_ellipical = cv.getStructuringElement(cv.MORPH_ELLIPSE,(5,5))
print(kernel_ellipical)
print(type(kernel_ellipical))
```

```
[[0 0 1 0 0]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [0 0 1 0 0]]
<class 'numpy.ndarray'>
```

Rectangle Kernel

```python
kernel = np.ones((5,5),np.uint8)
print(kernel)
print(type(kernel))
```

```
[[1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]]
<class 'numpy.ndarray'>
```

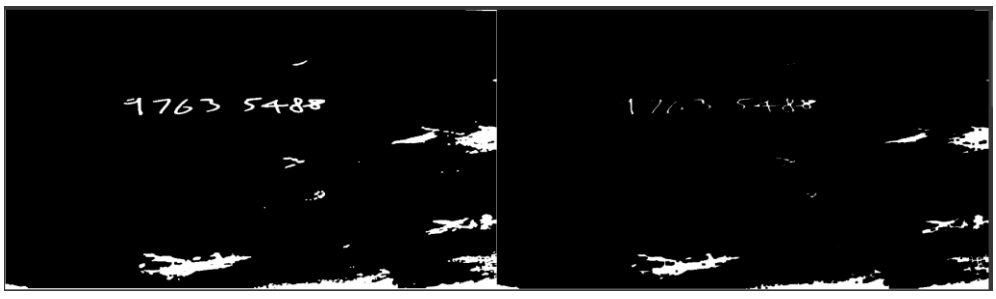
四、 實驗結果(貼圖與簡述說明)：

Averaging Blurring (均值濾波/平滑)

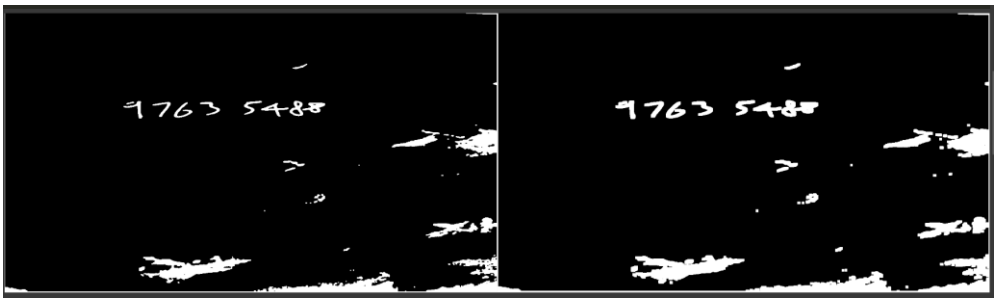
Median Blurring (中值平滑/濾波)

Gaussian Blurring (高斯平滑/濾波)

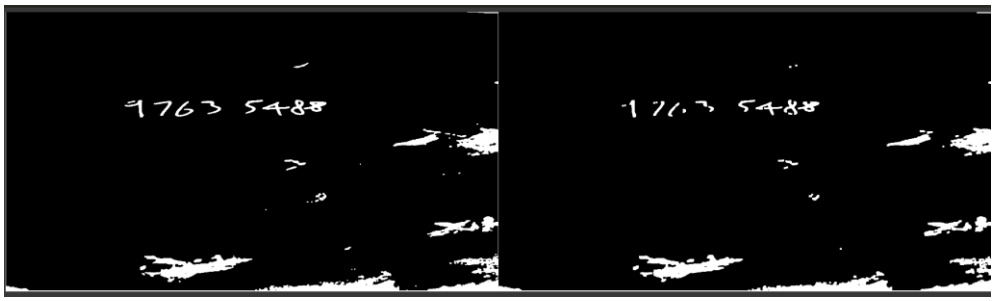
Morphological Transformations (形態學變化)



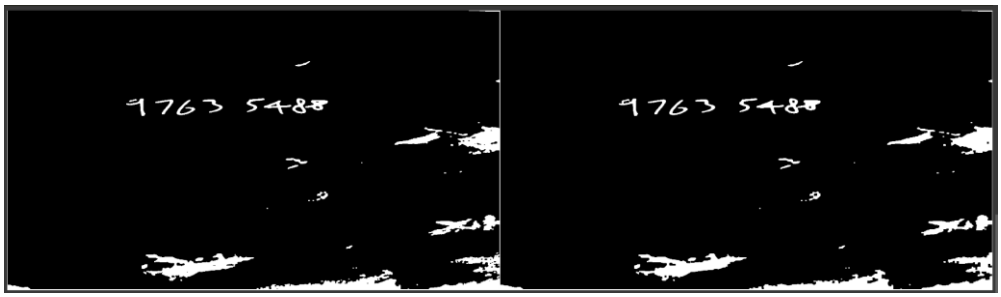
Erosion(侵蝕)



Dilation (膨脹)



Opening (斷開/開運算)



Closing (閉合/閉運算)

(可自行增頁)

*內容字級為 12 字級、中文為標楷體、英文為 Times New Roman