
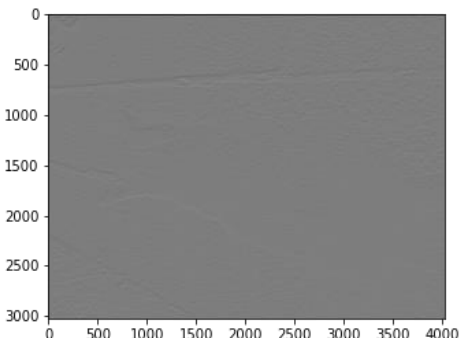

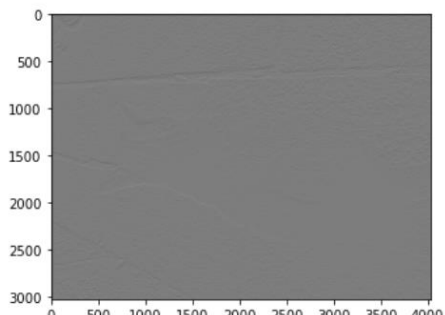



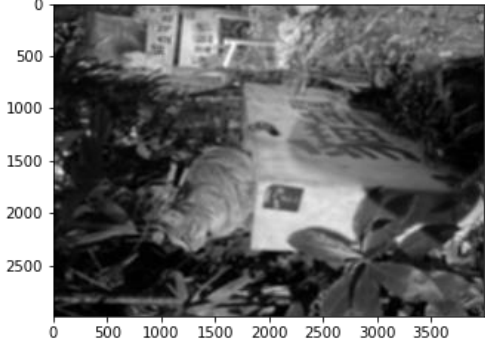
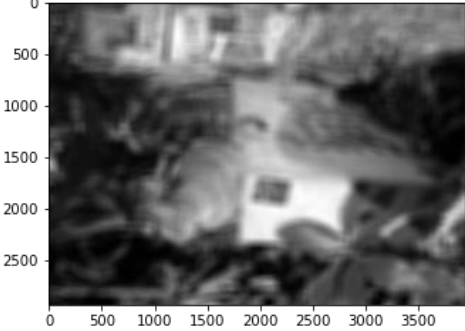
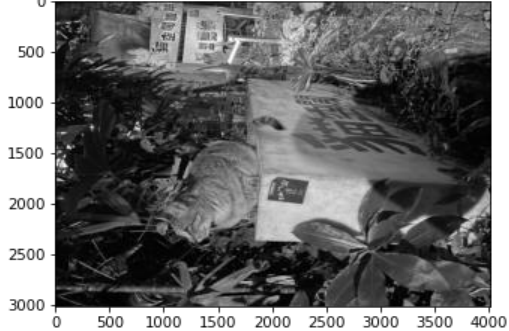

1. Sobel y 軸(邊緣偵測):對 y 軸(水平線)偵測線的特徵。在圖中可以看到水平線會被凸顯出來，然而垂直線並沒有。

表 1 Sobel y kernel

原始影像	濾波後影像
	
code	
<pre>In [2]: im = mpimg.imread("cat1.jpg") print(im.shape,type(im)) #rgb_image plt.imshow(im) im = im[:, :, 1] #turn to 2D image print(im.shape,type(im)) plt.savefig('sobel_original.png')  (3024, 4032, 3) &lt;class 'numpy.ndarray'&gt; (3024, 4032) &lt;class 'numpy.ndarray'&gt;</pre> 	
<pre>In [3]: import numpy as np from scipy.signal import convolve2d as conv2 #sobel y-axis filter kernel = np.array([[ -1, -2, -1], [ 0, 0, 0], [ 1, 2, 1]]) kernel output = conv2(im, np.flip(kernel), mode = 'valid') plt.imshow(output, cmap = 'gray') plt.savefig('sobel.png')</pre> 	

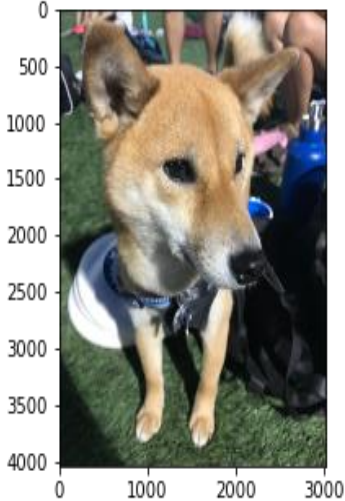
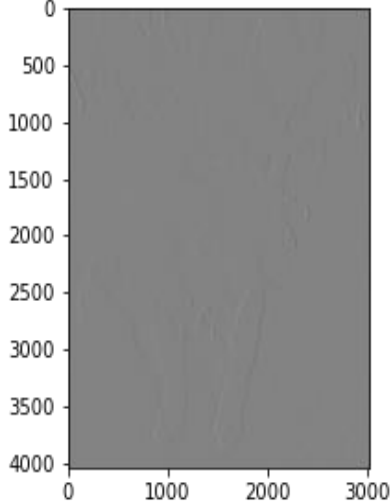
2. Blur(模糊化):將 kernel 設成不同大小的矩陣，矩陣中的數字設定成周圍點的平均值，可以看到影像被模糊化，當矩陣越大影像也會愈模糊。該特性的 kernel 可用來消除雜訊。

表 2 Blur kernel

原始影像	濾波後影像(40 × 40)
	
濾波後影像(100 × 100)	濾波後影像(10 × 10)
	
code	
<pre>In [13]: #blur filter kernel = np.ones((10,10), dtype = 'float') * (1.0/(10*10)) kernel output = conv2(im, np.flip(kernel), mode = 'valid') plt.imshow(output,cmap = 'gray') plt.savefig('blur_.png')</pre> 	

3. Sobel x 軸(邊緣偵測):對 x 軸(垂直線)偵測線的特徵。在表 3 可以看到垂直線會被凸顯出來與表 1 的特性相反，

表 3 Sobel x kernel

原始影像	濾波後影像
	

code

```
In [28]: #sobel x filter
kernel = np.array([[ -1,0,1],[ -2,0,2],[ -1,0,1]])
kernel
output = conv2(im, np.flip(kernel), mode = 'valid')
plt.imshow(output, cmap = 'gray')
plt.savefig('sobelx.png')
```

