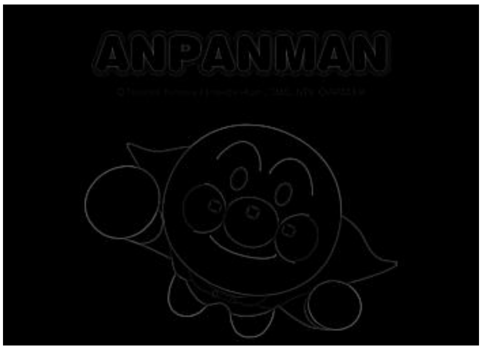
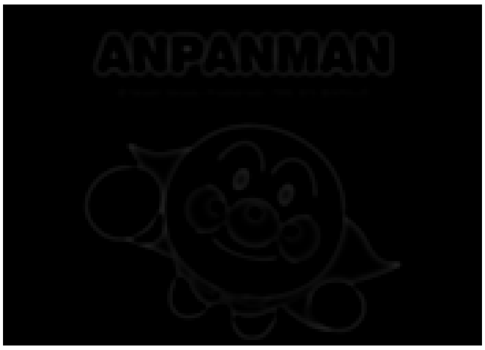
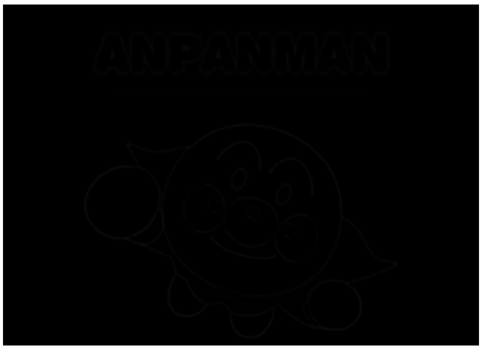
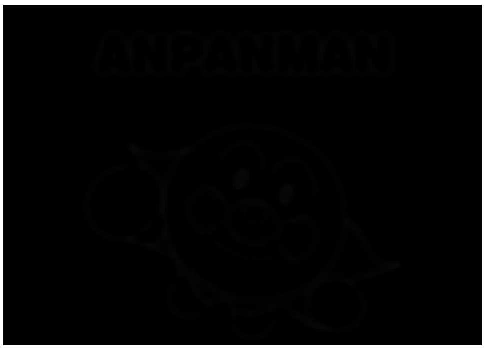
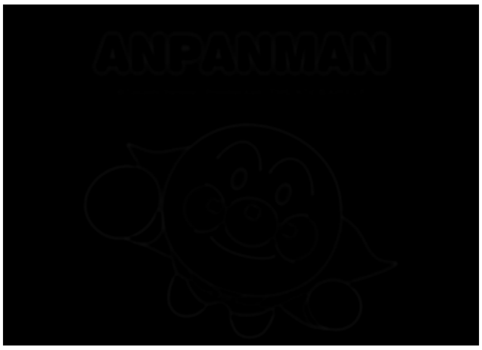
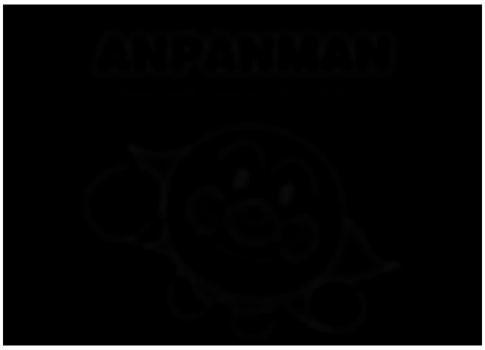
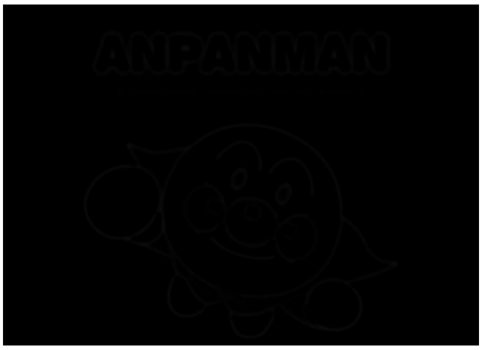
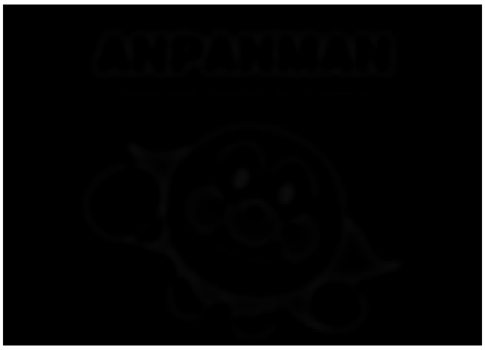
Computer Vision HW1 Report

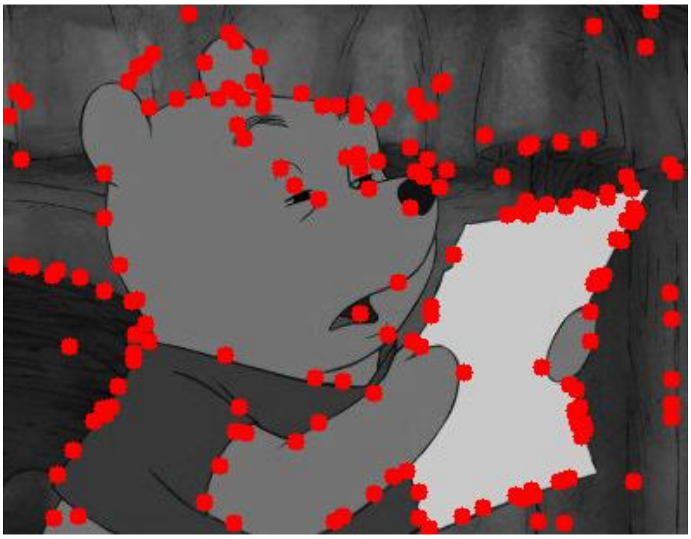
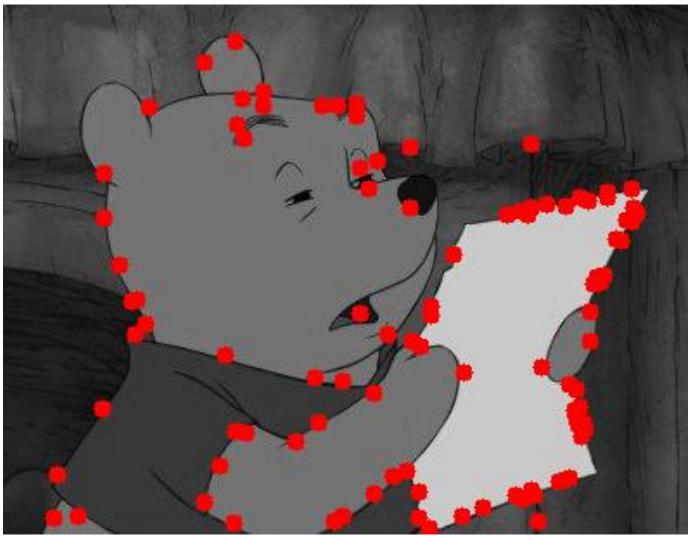
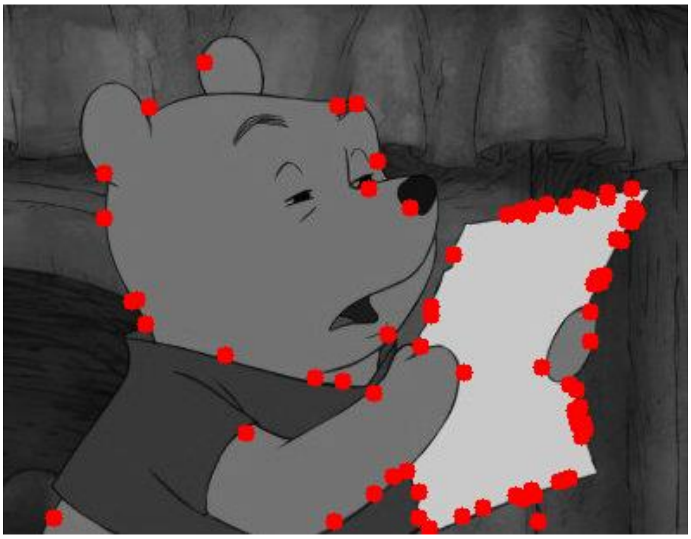
Student ID: R11521608
Name: 李俊昇

Part 1.

- **Visualize the DoG images of 1.png.**

	DoG Image (threshold = 3)		DoG Image (threshold = 3)
DoG1-1.png		DoG2-1.png	
DoG1-2.png		DoG2-2.png	
DoG1-3.png		DoG2-3.png	
DoG1-4.png		DoG2-4.png	

- **Use three thresholds (1,2,3) on 2.png and describe the difference.**

Threshold	Image with detected keypoints on 2.png
1	
2	
3	

(describe the difference)

當 threshold 設越高時，keypoints 數量就會減少，也較為精準，比較能抓到邊緣，然而當 threshold 設越低，會發現有些變化不大的地方也被當作 keypoints，比較粗糙。

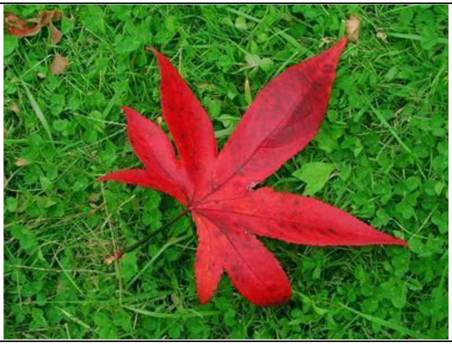
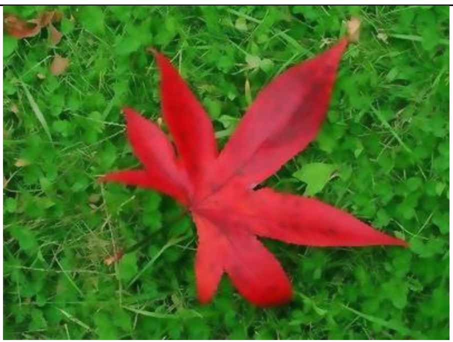
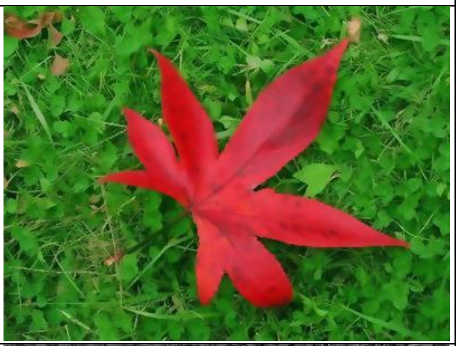
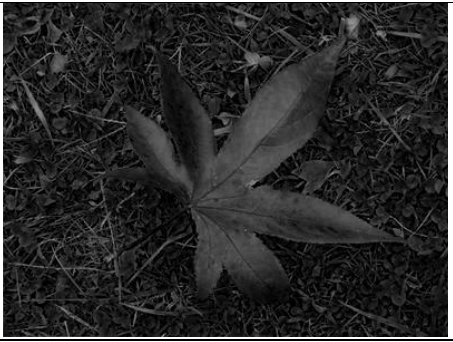
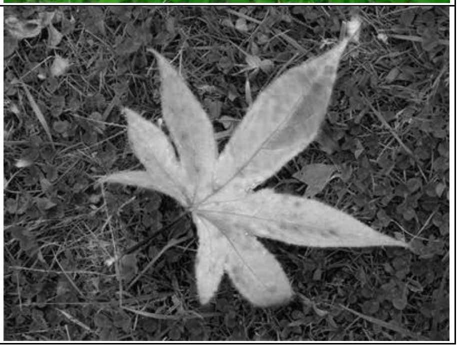
Part 2.

- **Report the cost for each filtered image.**

Gray Scale Setting	Cost (1.png)
cv2.COLOR_BGR2GRAY	1207799
R*0.0+G*0.0+B*1.0	1439568
R*0.0+G*1.0+B*0.0	1305961
R*0.1+G*0.0+B*0.9	1393620
R*0.1+G*0.4+B*0.5	1279697
R*0.8+G*0.2+B*0.0	1127913

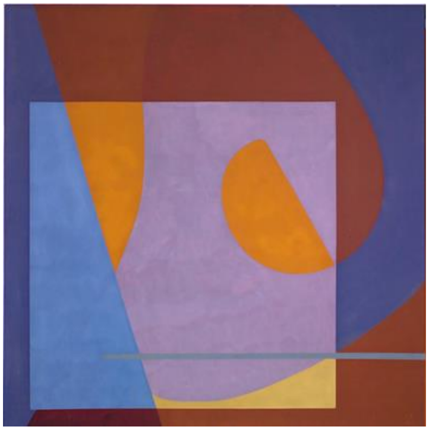
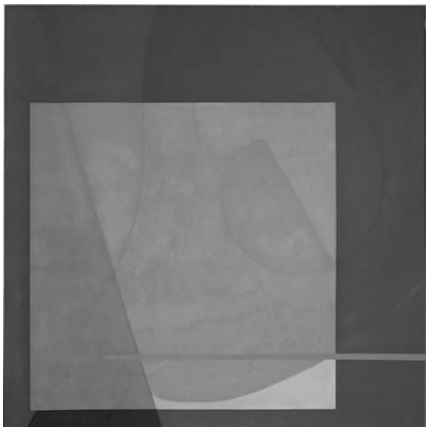
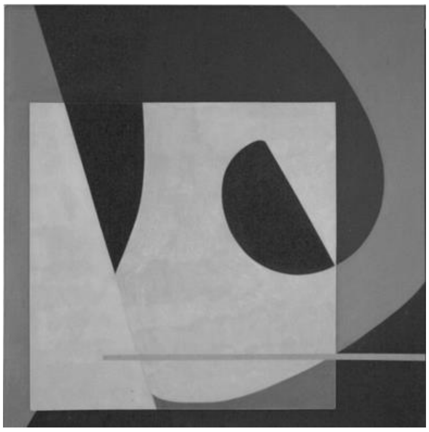
Gray Scale Setting	Cost (2.png)
cv2.COLOR_BGR2GRAY	183850
R*0.1+G*0.0+B*0.9	77882
R*0.2+G*0.0+B*0.8	86023
R*0.2+G*0.8+B*0.0	188019
R*0.4+G*0.0+B*0.6	128341
R*1.0+G*0.0+B*0.0	110862

- **Show original RGB image / two filtered RGB images and two grayscale images with highest and lowest cost.**

Original RGB image (1.png)	Filtered RGB image and Grayscale image of Highest cost	Filtered RGB image and Grayscale image of Lowest cost
		
		

(Describe the difference between those two grayscale images)

Cost 最大值整題偏暗，看不出差異，而 cost 最小值的可以發現紅色區域非常的白，灰階值較高

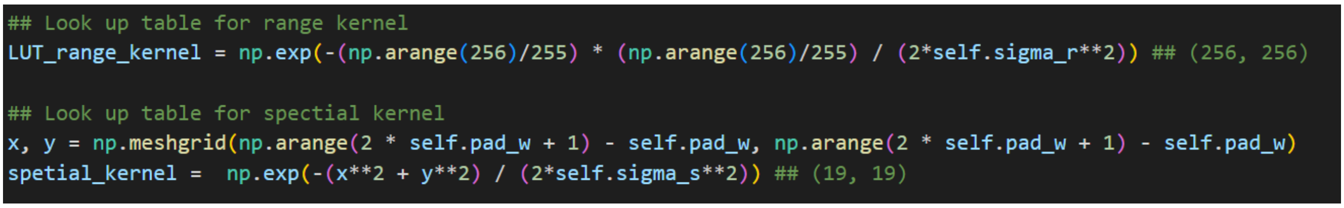
Original RGB image (2.png)	Filtered RGB image and Grayscale image of Highest cost	Filtered RGB image and Grayscale image of Lowest cost
		
		

(Describe the difference between those two grayscale images)

Cost 最大值整體色差不明顯，尤其是籃紅兩色塊，但 cost 最小值那張圖可以發現藍色區域灰階值交高，因此也比較白，而紅色區域很暗，灰階值很低。

- **Describe how to speed up the implementation of bilateral filter**

  在這部分我是利用 look up table 打成表格，這樣可以 O(1) 找值，不用再 for loop 迴圈。程式碼如下。



```
## Look up table for range kernel
LUT_range_kernel = np.exp(-(np.arange(256)/255) * (np.arange(256)/255) / (2*self.sigma_r**2)) ## (256, 256)

## Look up table for spectial kernel
x, y = np.meshgrid(np.arange(2 * self.pad_w + 1) - self.pad_w, np.arange(2 * self.pad_w + 1) - self.pad_w)
spetial_kernel =  np.exp(-(x**2 + y**2) / (2*self.sigma_s**2)) ## (19, 19)
```