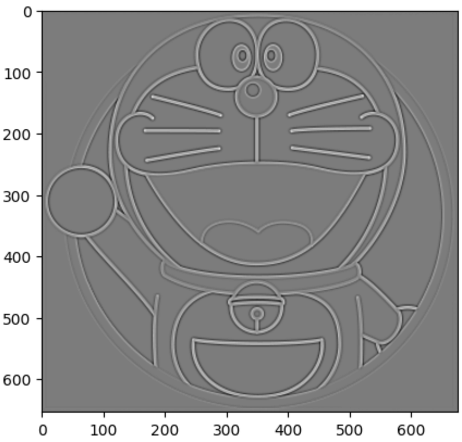
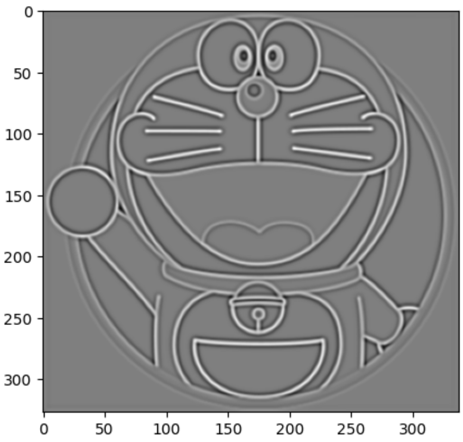
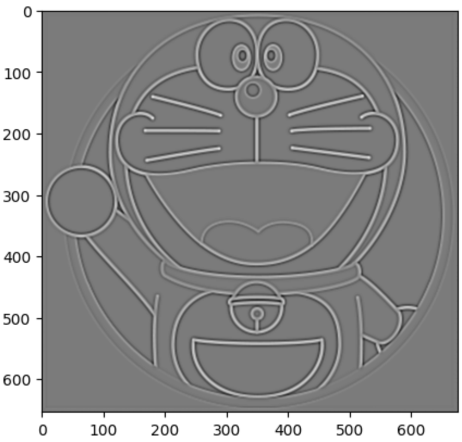
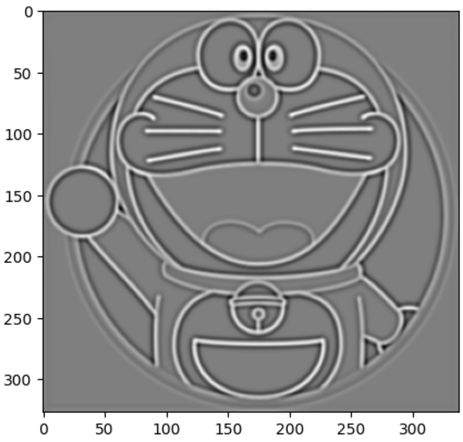
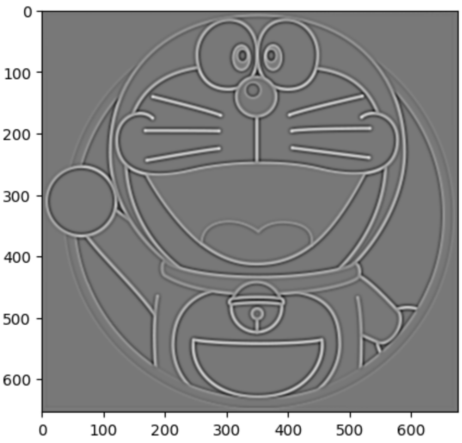
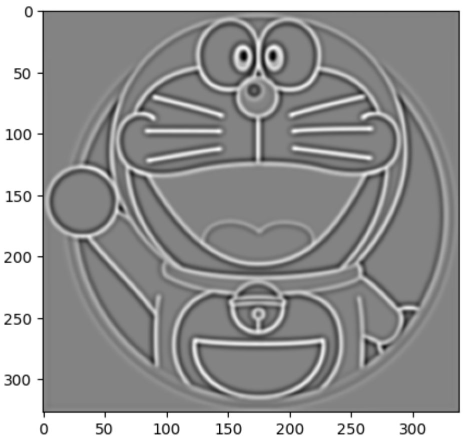
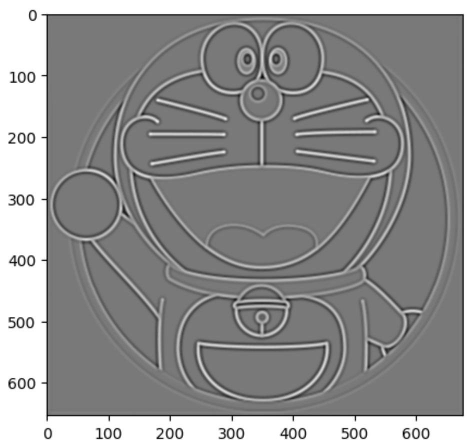
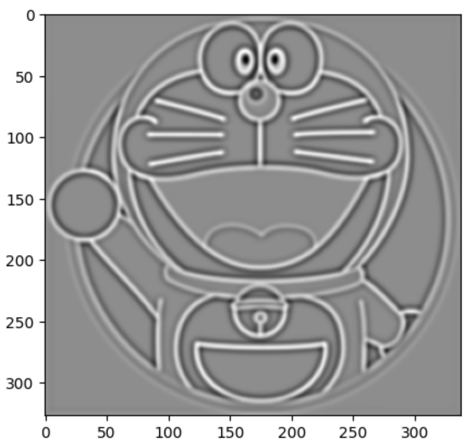
Computer Vision HW1 Report

Student ID: B08902092
Name: 陳翰雯

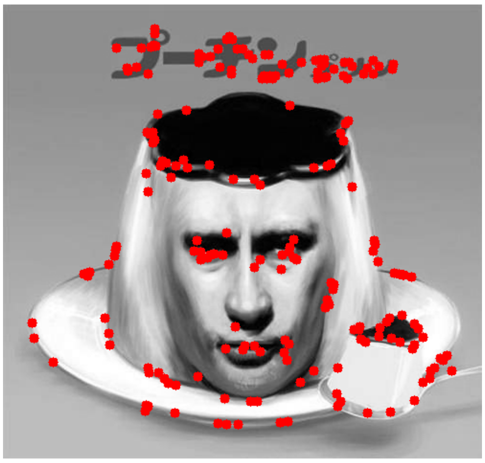
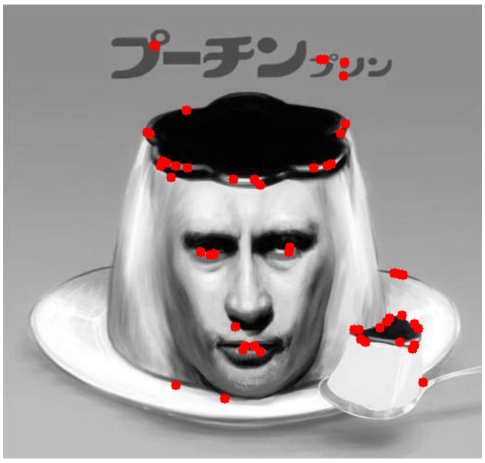
Part 1.

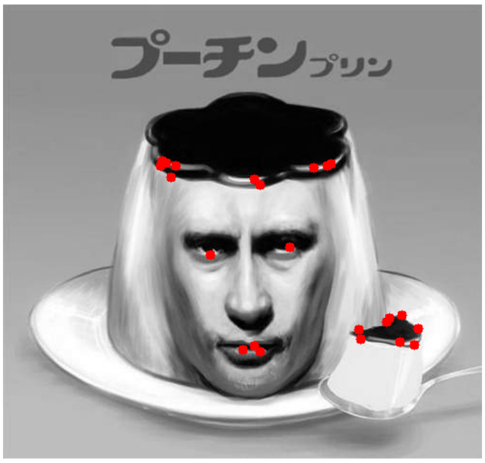
- **Visualize the DoG images for 1.png.**

	DoG Image (threshold = 5)		DoG Image (threshold = 5)
DoG1-1.png	 0 100 200 300 400 500 600 0 100 200 300 400 500 600	DoG2-1.png	 0 50 100 150 200 250 300 0 50 100 150 200 250 300
DoG1-2.png	 0 100 200 300 400 500 600 0 100 200 300 400 500 600	DoG2-2.png	 0 50 100 150 200 250 300 0 50 100 150 200 250 300
DoG1-3.png	 0 100 200 300 400 500 600 0 100 200 300 400 500 600	DoG2-3.png	 0 50 100 150 200 250 300 0 50 100 150 200 250 300

DoG1-4.png		DoG2-4.png	

- **Use three thresholds (2, 5, 7) on 2.png and describe the difference.**

Threshold	Image with detected keypoints on 2.png
2	
5	

7	

(describe the difference)

As the threshold value increases, we could see a decrease in the key points of the image. When threshold=2, we could see that there are few parts in the image where the key points cluster together. When threshold=5, there are lesser clusters the key points are more spread and is found in places that is somehow darker. When threshold=7, there are even fewer clusters and the key points are only shown in places where there is great difference in its color intensity.

Part 2.

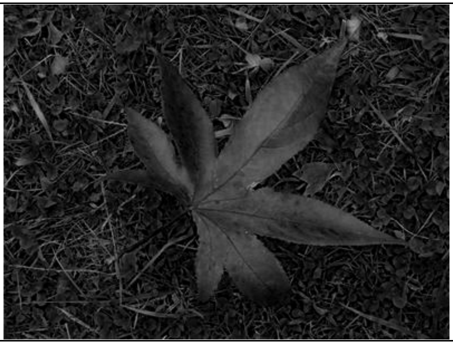
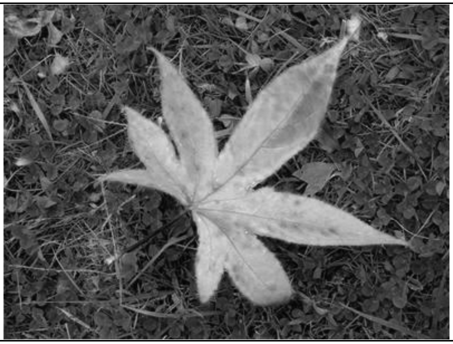
- **Report the cost for each filtered image.**

Gray Scale Setting	Cost (1.png)
cv2.COLOR_BGR2GRAY	120779
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	1394216
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	183851
R*0.1+G*0.0+B*0.9	77901
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

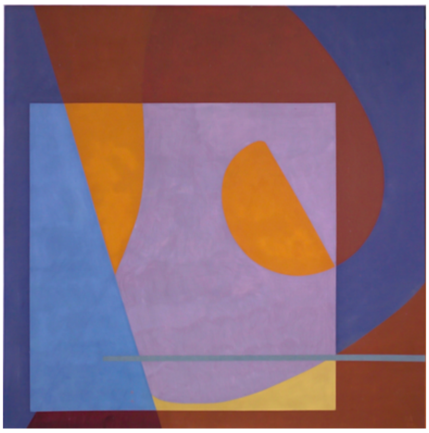
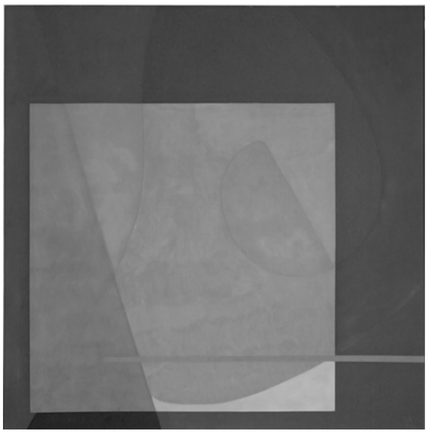
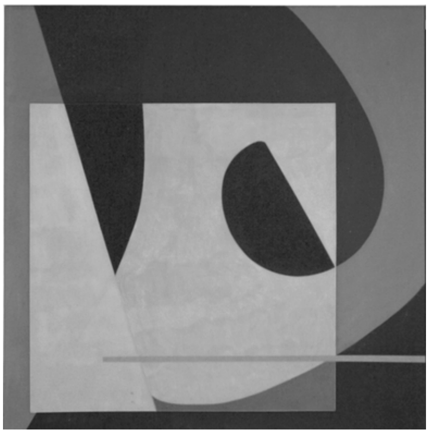
- **Show original RGB image / two filtered RGB images and two grayscale images with highest and lowest cost.**

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)

In general, the highest cost grayscale image is overall darker than the lowest cost grayscale image. The main difference, however, is the color of the leaves. In the case of highest cost grayscale image, the leaf has darker color, moreover, we could also see that the color of the leaf and the background are somehow similar (both have dark colors). On the other hand, in lowest cost, we could see great contrast in the color of the leaf and its background (the leaf has light color, and the background has dark color).

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)

The main difference I could see is the overall color in general. In the highest cost grayscale images, we could see that there is no big difference in color intensity between one part and the others. However, in lowest cost grayscale images, we could see great contrast in the intensity of colors. In 2.png image, in highest cost grayscale image details are not clearly seen, what we mainly see is ‘a small box inside a bigger box’. But, in lowest cost grayscale images, we could see the semicircle and many more details due to its high contrast in color intensity.

Describe how to speed up the implementation of bilateral filter.

1. The use of LUT for range kernel: It prevents the need to count it in a for loop. Whereas accessing a table or array takes only O(1) time complexity.
2. The use of numpy arrays: The numpy library provides lots of functions that allow you to manipulate the whole array rather than looping through each value inside the array.