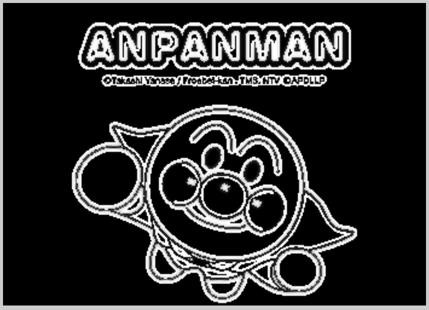
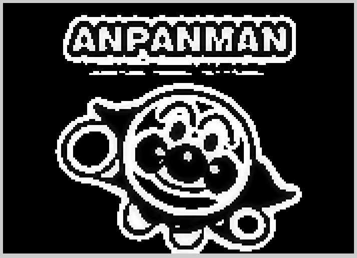
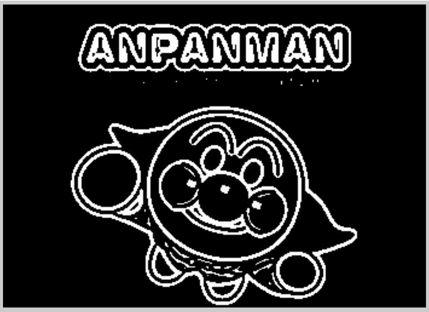
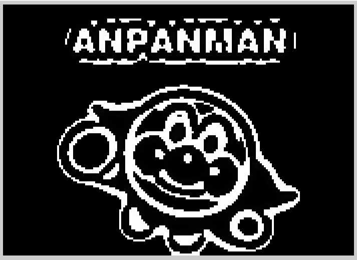
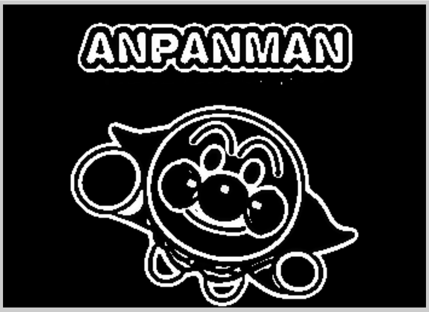
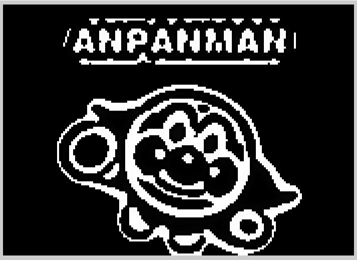
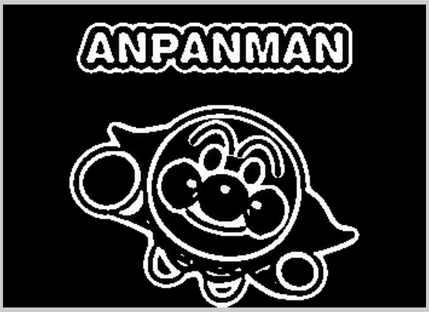
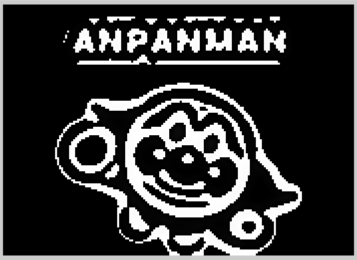
Computer Vision HW1 Report

Student ID: B10705052
Name: 葉又銘

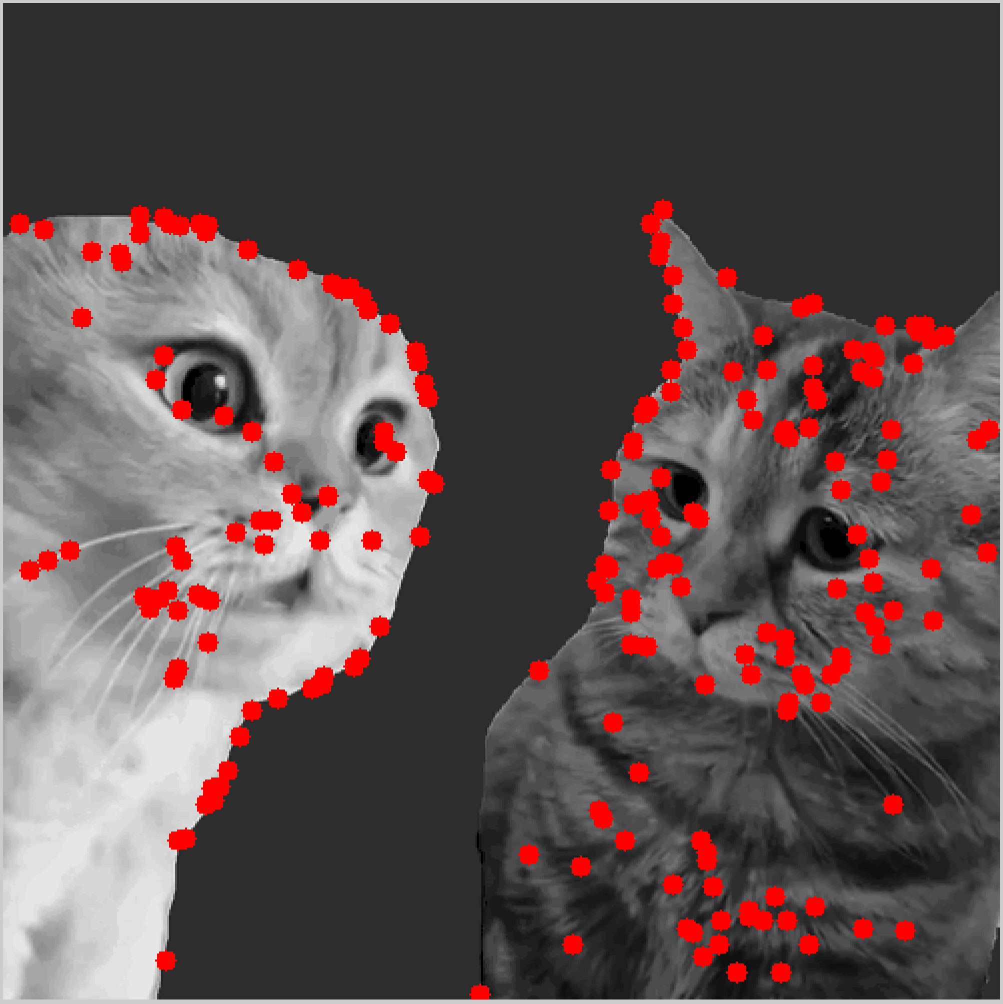
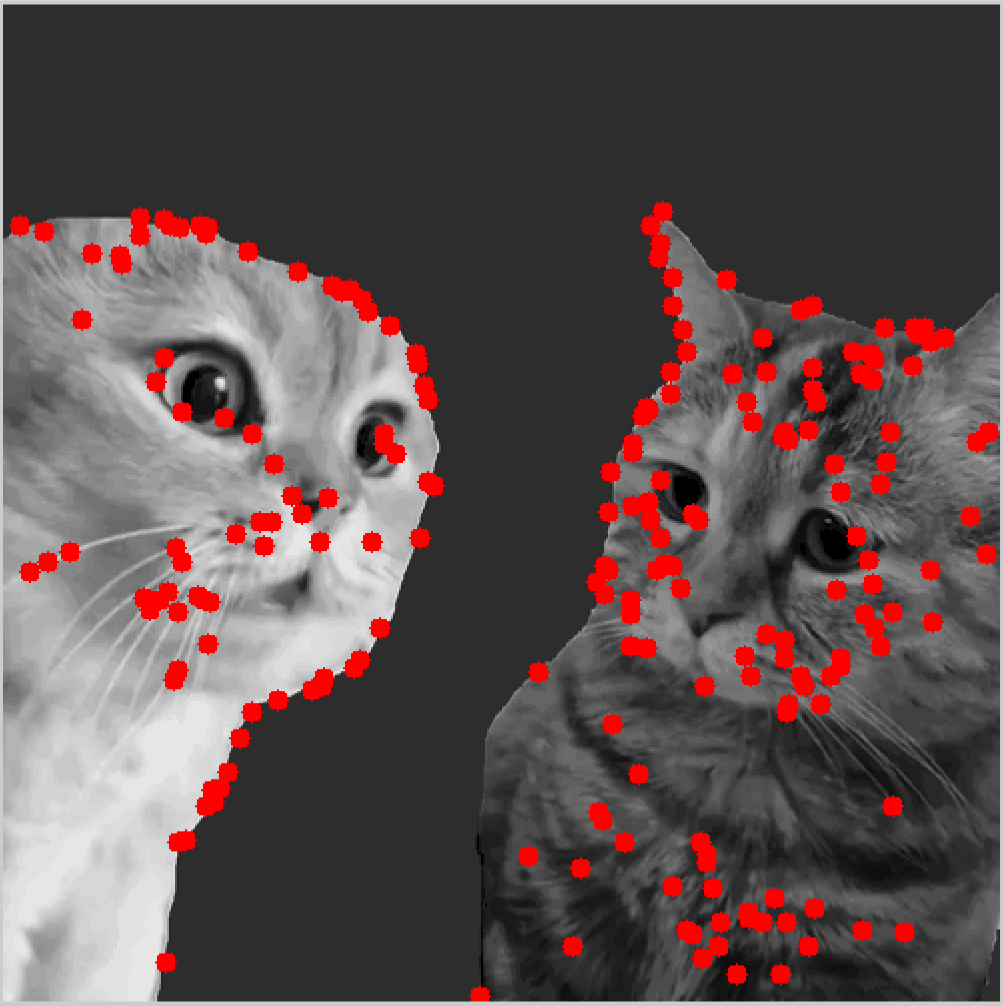
Part 1.

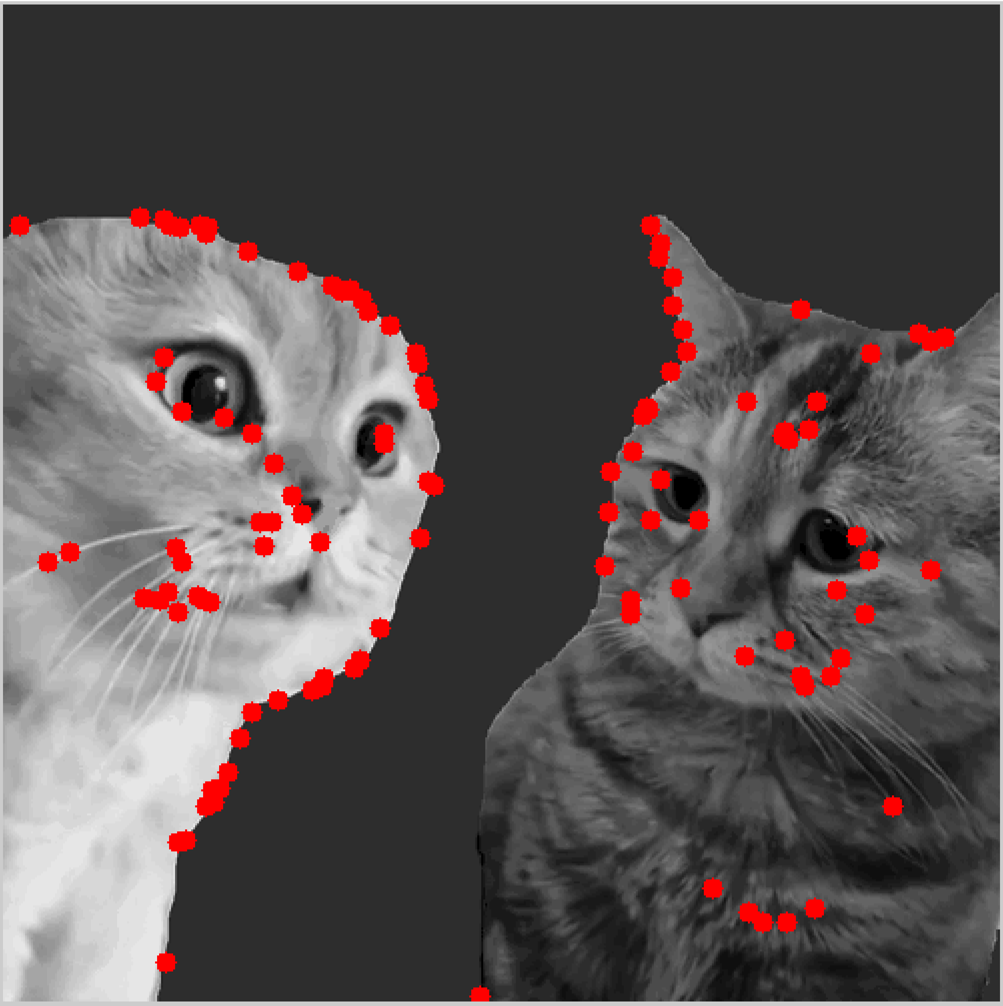
- **Visualize the DoG images of 1.png.**

	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	

- Images with keypoints detected using lower thresholds tend to have more keypoints than the higher one. This is because a lower threshold value is more inclusive, considering even the less pronounced features as significant. As a result, you'll notice more points marked on the image, including those in less distinct areas. This can be useful for detailed analysis but may also lead to noise in the form of irrelevant keypoints. On the other hand, if you want to have more precise detection, you might want to choose higher thresholds but may also lead to missing some useful keypoints.

Part 2.

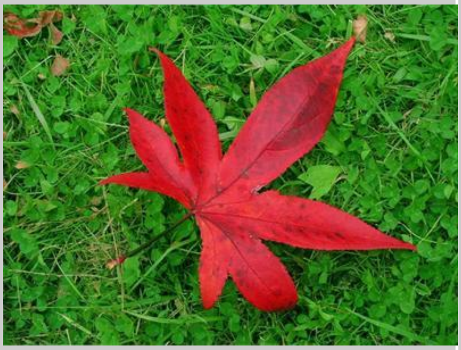
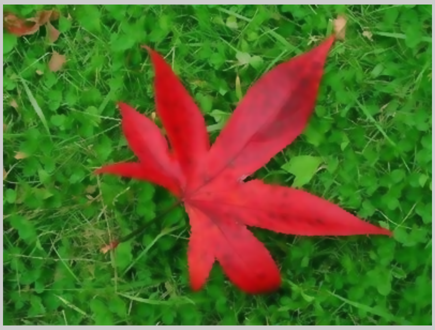
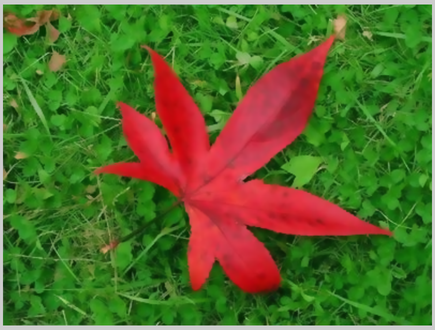
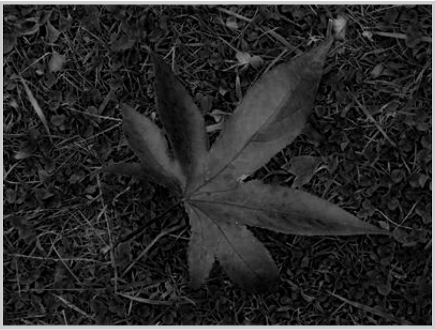
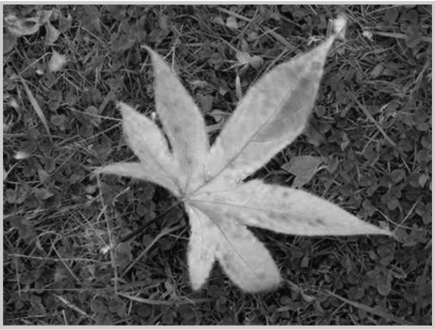
- **Report the cost for each filtered image.**

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	1390952
R*0.1+G*0.4+B*0.5	1278834
R*0.8+G*0.2+B*0.0	1127294

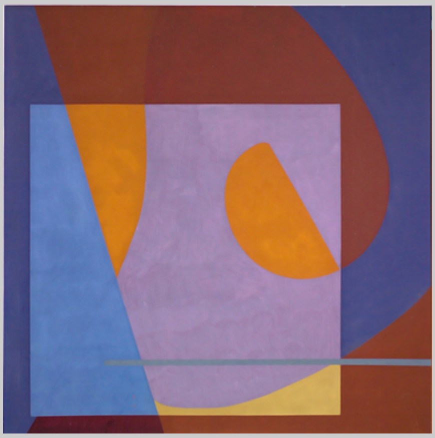
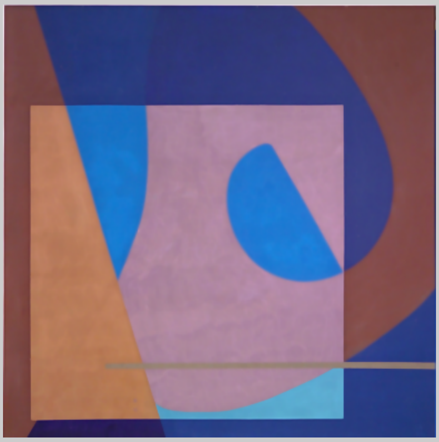
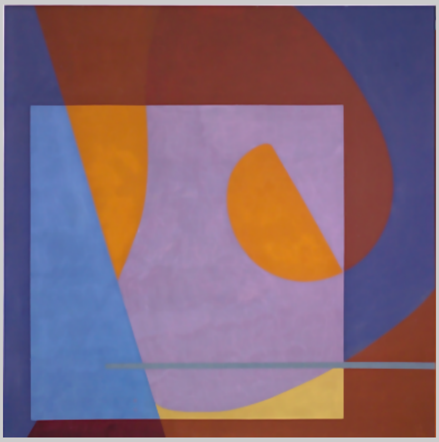
Gray Scale Setting	Cost (2.png)
cv2.COLOR_BGR2GRAY	183851
R*0.1+G*0.0+B*0.9	77797
R*0.2+G*0.0+B*0.8	85864

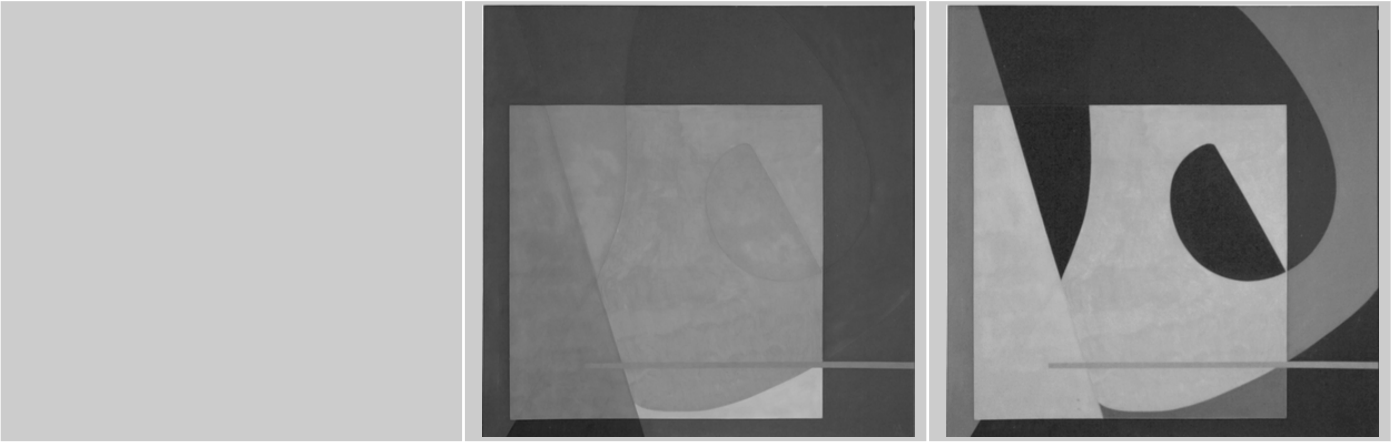
R*0.2+G*0.8+B*0.0	188065
R*0.4+G*0.0+B*0.6	128150
R*1.0+G*0.0+B*0.0	110862

- **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
		
		

The grayscale image of the lowest cost is preferred because it better preserves the perceptual similarity to the original color image. The grayscale image of the lowest cost preserves more details in the background (the grass) behind the leaf, which can provide more guidance to the filter.

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



The grayscale image of the lowest cost is preferred because it better preserves the brightness and darkness of the original color image. This can be observed simply by comparing the brighter area (the square) in the original color image with the grayscale images. The brighter area corresponds to the grayscale image of the lowest cost.

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

1. Vectorization with NumPy: By utilizing NumPy's ability to perform operations on entire arrays at once rather than element-by-element, the need for explicit for-loops is eliminated. This significantly reduces the computational overhead, leading to faster execution times.
2. Precomputation of Spatial Kernel: The spatial component of the bilateral filter, which does not change during the filtering process, is computed ahead of time. This step avoids redundant calculations for each pixel, further improving efficiency.
3. Selective Iteration Over Pixels: Rather than processing every pixel in the image, this optimized approach focuses on iterating through each pixel in a specific window. This method modifies the computational complexity from $O(c_1\ n^2\ \cdot\ c_2 m^2)$ to $O(c_3\ n^2\ \cdot\ c_4 m^2)$, where n is the total number of image pixels, and m is the number of pixels in the window. Here, c_1 and c_3 represent the computational efforts before and after optimization for handling the entire image, respectively, while c_2 and c_4 indicate the efforts for processing each window, with c_3 being reduced thanks to NumPy's built-in functions for vectorized operations, thereby optimizing speed.
4. Efficient Ordering of Operations: Adjusting the sequence of operations, such as performing division by the total weight at once, can avoid unnecessary repetition of operations.