Computer Vision HW1 Report

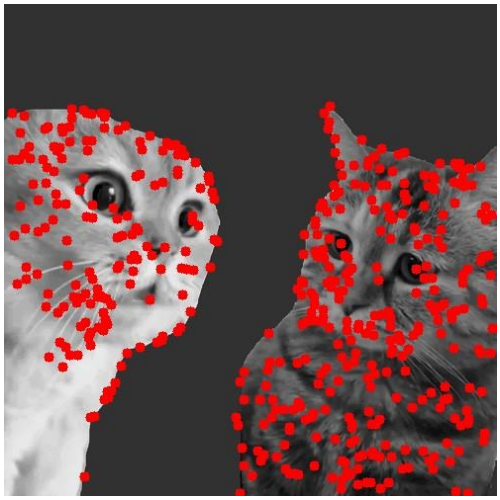
Student ID: R11521201
Name: 廖沁柔

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)

If a lower threshold value is set (threshold=1), it means that pixel points with even minor intensity changes might be identified as keypoints. This can lead to a large number of keypoints being detected, including many noise points or insignificant feature points. While this increases the quantity of detected features, it might decrease the accuracy and efficiency of subsequent processing steps (like feature matching and object recognition) due to the inclusion of many irrelevant or low-quality feature points.

In the contrast, if a higher threshold value is set (threshold=3), only those pixel points with significant intensity changes will be recognized as key points. This reduces the number of detected key points but increases the quality of the key points since they represent more prominent features in the image. A higher threshold helps exclude noise and insignificant features but might also miss some smaller or more subtle genuine features.

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

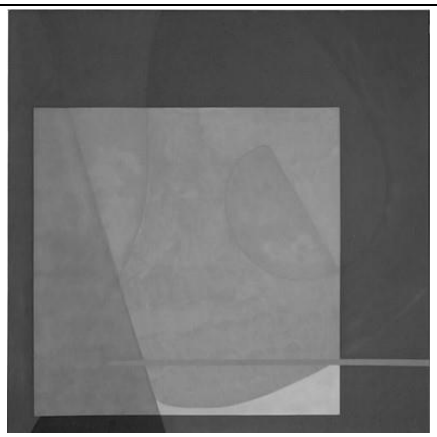
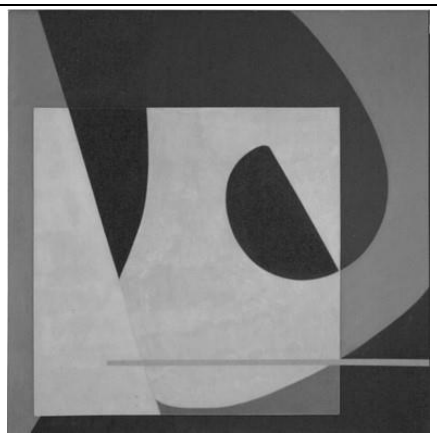
- **Show original RGB image / two filtered RGB images and two grayscale images with highest and lowest cost.**

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)

The grayscale image of highest cost has grayscale values that are more homogeneous and a lower contrast. This results in less distinction between the red leaf and the green grass, with overall details and textures being more blurred. This low-contrast treatment may cause the features within the image to be less prominent and make it more challenging to discern details.

The grayscale image of lowest cost exhibits higher contrast. The shape and edges of the leaf are clearer, and the separation from the background is more pronounced. The higher contrast enhances the visual depth of the image, allowing for clear identification of textures in both the leaf and the grass, providing richer visual information.

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 grayscale image of lowest cost has higher contrast. This means the boundaries of objects within the image are more pronounced, and the distinction between objects and the background is more apparent. For example, the lines and internal textures of various geometric shapes are clear. Such an image generally provides better visual effectiveness, enabling viewers to identify and understand the contents of the image more easily.

The image in the middle, the grayscale image of highest cost, has relatively lower contrast. The separation between the object boundaries and the background is not as distinct as in the lowest cost image, making the overall image appear smoother but also blurrier. The contours of the geometric shapes are less defined, which might make visual recognition more challenging.

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

Vectorization:

Replace the nested for-loops with vectorized operations that leverage NumPy's efficient array processing.

Look-Up Tables (LUTs):

Use LUTs for the spatial and range kernels.

LUTs are being used efficiently and are not being recalculated unnecessarily.