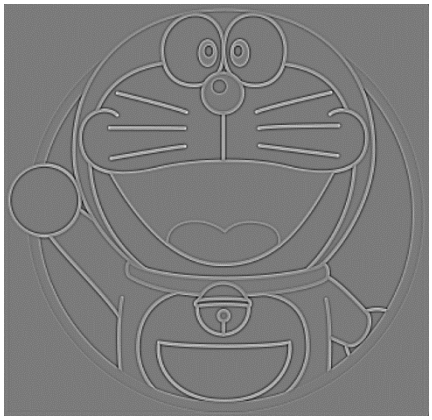
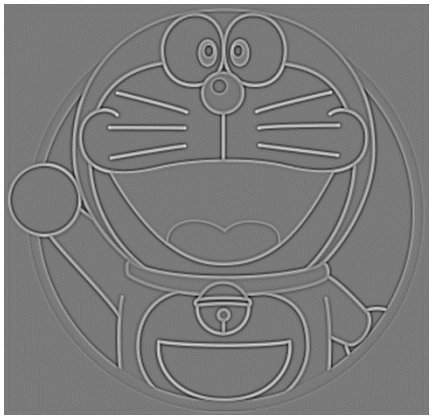
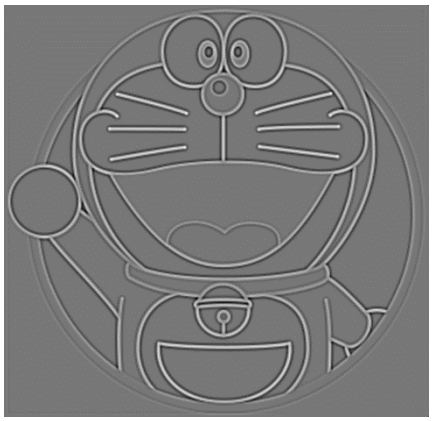
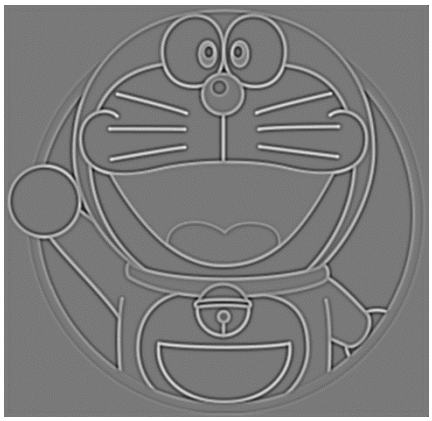
Computer Vision HW1 Report

Student ID: R10521802
Name: 李睿莆

Part 1.

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

	DoG Image (threshold = 5)		DoG Image (threshold = 5)
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 (2, 5, 7) on 2.png and describe the difference.**

Threshold	Image with detected keypoints on 2.png
2	
5	

7	

(describe the difference)

The number of detected keypoints depends on the magnitude of threshold. The higher the threshold, the higher value the local extremum in the DoG needs to reach so that it can be detected as a keypoint. Hence, we can see the number of keypoints decreases when we raise the threshold.

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

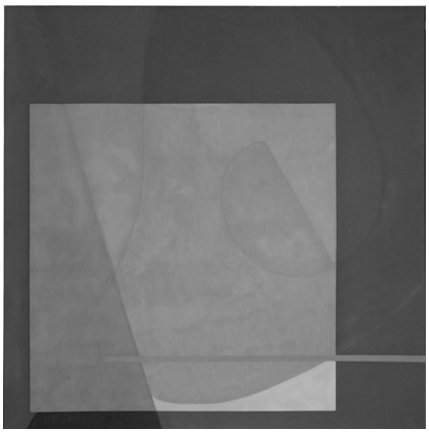
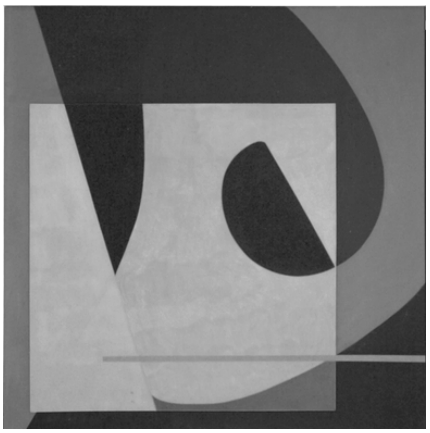
- **Show original RGB image / two filtered RGB images and two grayscale images with highest and lowest cost.**

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 lowest cost grayscale image performs better than the highest cost image. As we can see the texture and the luminance of the maple is clear enough to tell. Hence, (R*0.8+G*0.2+B*0.0) **seems to be the appropriate combination to transform to grayscale image.**

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 lowest cost grayscale image performs better than the highest cost image. As we can see the edge and the contrast in the image is clear enough to tell. Hence, (R*0.1+G*0.0+B*0.9) **seems to be the appropriate combination to transform to grayscale image.**

- **Describe how to speed up the implementation of bilateral filter.**
- Compute the spatial kernel once, then traverse pixels of original image inside padded image from left to

right and from top to bottom.

- Compute and update range kernel whenever we iterate to the next center pixel. Then multiply spatial kernel and range kernel to get bilateral filter, so we get the filtered pixel.