

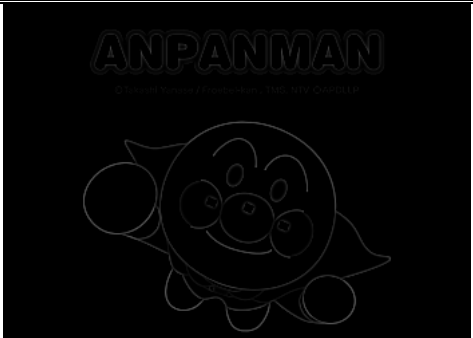







# Computer Vision HW1 Report

Student ID: R12945040

Name: 郭思言

## Part 1.

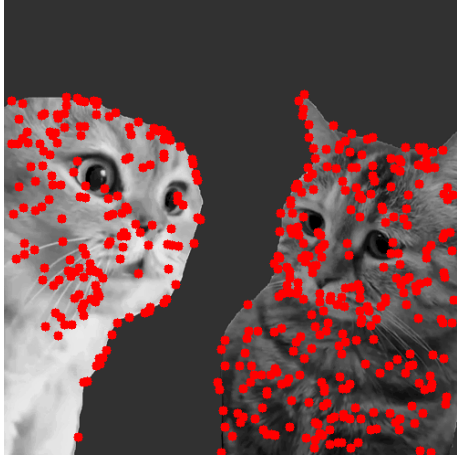
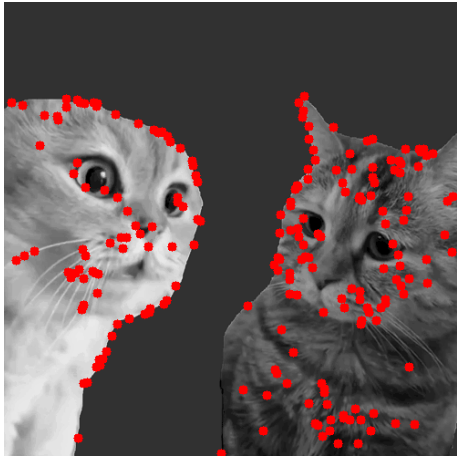

- Visualize the DoG images of 1.png.

	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	

### Notes:

- When using the Difference of Gaussian (DoG) window to find features, the range covered in one sweep (in the original image) will vary at different levels.

- 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)

#### Difference:

- With a lower threshold value (e.g., 1), more feature points can be detected in the image, but the quality of these features may be lower (some feature points within the cats' bodies are included).
- Using a higher threshold value (e.g., 3) results in fewer observed feature points, but the remaining features generally have higher quality compared to the first image (feature points on the edges of the cats are mostly retained).
- The feature points are overlapped: the feature points detected from threshold 3 are also detected by threshold 2 and 1.

## Part 2.





- Report the cost for each filtered image.

(For this part, the cost is calculated as the L1 norm between the bilateral filter output image and the joint bilateral filter output 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




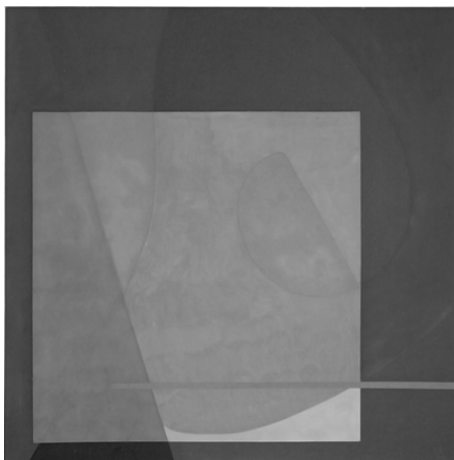
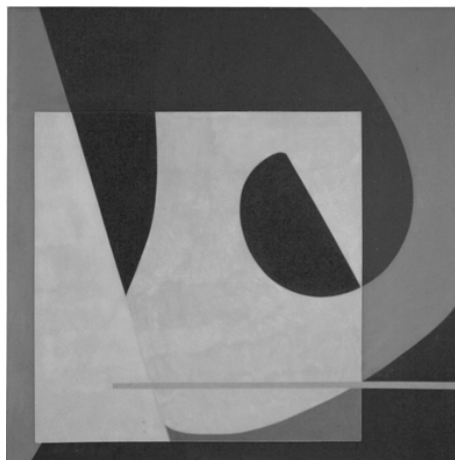
Gray Scale Setting	Cost (2.png)
cv2.COLOR_BGR2GRAY	183850
$R*0.1+G*0.0+B*0.9$	77883
$R*0.2+G*0.0+B*0.8$	86023
$R*0.2+G*0.8+B*0.0$	188019
$R*0.4+G*0.0+B*0.6$	128341
$R*1.0+G*0.0+B*0.0$	110862

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

Original RGB image (1.png)	Filtered <u>RGB image</u> and <u>Grayscale image</u> of Highest cost	Filtered <u>RGB image</u> and <u>Grayscale image</u> of Lowest cost
		
		

(Describe the difference between those two grayscale images)

- The grayscale image with a higher cost appears relatively darker than the grayscale image with a lower cost. The lower cost image is brighter and looks overall clearer, preserving more details.
- In the main part of the leaves, the leaf edges are brighter, making it easy to distinguish between the leaves and the background.

Original RGB image (2.png)	Filtered <u>RGB image</u> and <u>Grayscale image</u> of Highest cost	Filtered <u>RGB image</u> and <u>Grayscale image</u> of Lowest cost
		
		

(Describe the difference between those two grayscale images)

- The grayscale image with a higher cost loses many of the geometric shapes from the original image. We can only roughly see a dark background with a lighter rectangular area in the middle and a few lines in between.
- The grayscale image with a lower cost shows various geometric shapes from the original image very clearly, including the darkest diagonal half-circle in the middle, the darker irregular shape above it, and many curved and diagonal lines.

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

1. Kernel Precomputation: The spatial and range kernels can be precomputed outside the main loop. This reduces redundant computations since the same kernels are reused for each pixel in the image.
2. Rolling Operation: Using `np.roll` allows for efficient shifting of the padded image to compute the range kernel weights. This avoids the need for explicit nested loops to iterate over the neighboring pixels.
3. Weighted Sum Calculation: The weighted sum of neighboring pixels is computed using array operations, which are more efficient than explicit loops. This is done for each channel separately, which allows for parallel computation.