



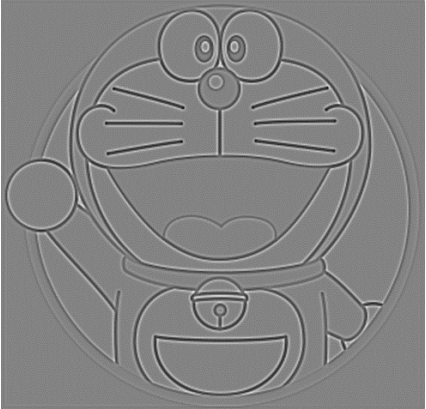



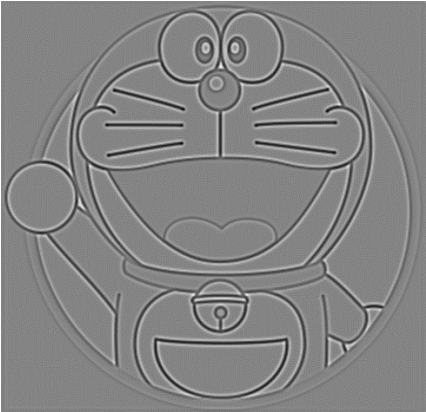

Computer Vision HW1 Report

Student ID: R10522606

Name: 曾柏翔

Part 1.

- Visualize the detected corner for 1.png.

| | DoG Image (threshold = 5) | | DoG Image (threshold = 5) |
|------------|-------------------------------------------------------------------------------------|------------|---------------------------------------------------------------------------------------|
| DoG1-1.png |  | DoG1-1.png |  |
| DoG1-2.png |  | DoG1-2.png |  |
| DoG1-3.png |  | DoG1-3.png |  |
| DoG1-4.png |  | DoG1-4.png |  |

- Use three thresholds (5, 10, 15) on 2.png and describe the difference.

| Threshold | Image with detected keypoints on 2.png | |
|-----------|--------------------------------------------------------------------------------------|--|
| 2 |  | |
| 5 |  | |
| 7 |  | |

(describe the difference)

可以從上圖看到，keypoints 皆出現在顏色落差的邊緣處，並且當 thresholds 越來越大時，極值點數目則越來越少。正如同前面所說的 keypoints 出現在落差處，而點數目下降正是因為落差不夠明顯，因此被閾值過濾掉了。換句話說，thresholds 提高時，只會減少原有的 keypoints，並不會產生新的點。






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 |


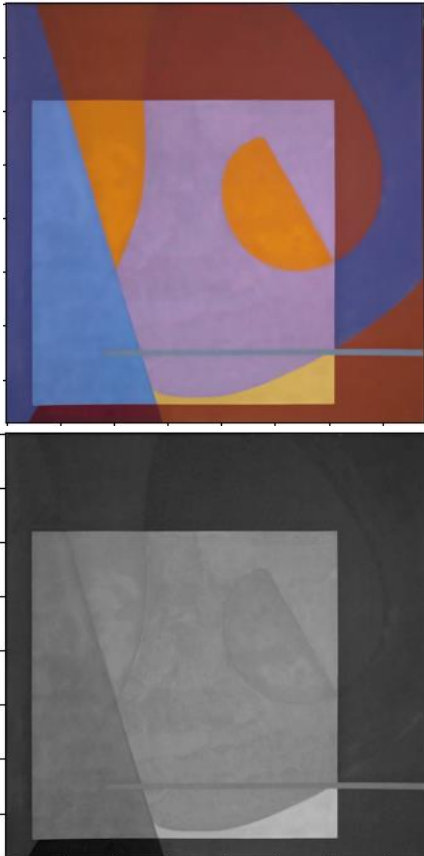
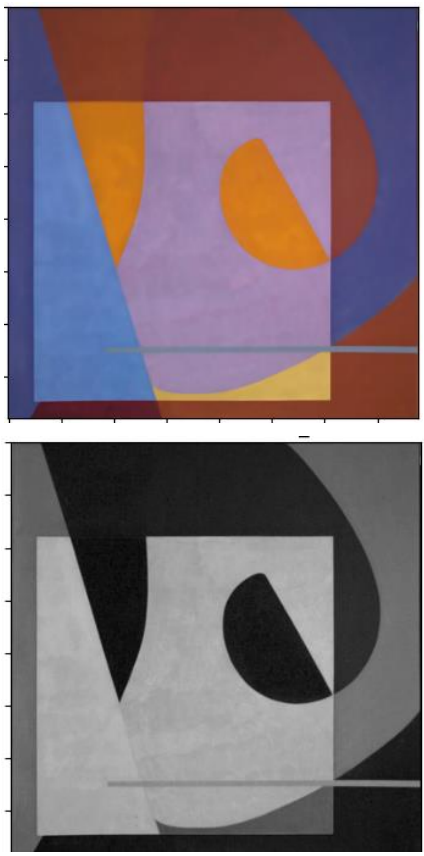
| Gray Scale Setting | Cost (2.png) |
|---------------------|--------------|
| cv2.COLOR_BGR2GRAY | 183850 |
| $R*0.1+G*0.0+B*0.9$ | 77882 |
| $R*0.2+G*0.0+B*0.8$ | 86023 |
| $R*0.2+G*0.8+B*0.0$ | 188019 |
| $R*0.4+G*0.0+B*0.6$ | 128341 |
| $R*1.0+G*0.0+B*0.0$ | 110862 |

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

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

(Describe the difference between those two grayscale images)

可以先明顯看到兩張 grayscale images 圖整體的深淺很不一樣，但這並無法直接斷言哪一張的結果更好。接著注意到的是內部的一些細節，整張照片的重點應該是楓葉，因此我們應該把焦點放在楓葉的明顯(對比)度，可以看到右邊的楓葉相對與草地是更顯眼的。相反地，左邊楓葉與草地都屬於較深的灰階。這也符合計算出來的 cost，左邊為 Highest cost、右邊為 Lowest cost。

| 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)

正如前一部分所提到，無法直接用深淺程度來斷言哪一張照片更好。這部分就與前一部分剛好相反，右邊的圖比左邊的更深，但在整體的輪廓線條上，有更強烈的對比，因此得到了較低的 cost，也更能呈現原始圖片的顏色差異。

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

在運算 spatial kernel 和 range kernel 時，由於牽扯到 exponential 的計算，因此會比較費時。所以在運算時，以 look up table 的方式來避免在 for 迴圈中，做過多的反覆運算。並再加上平行計算的方式，來減少 for 迴圈的使用，讓矩陣運算時更省時間。