作業2: Image Enhancement

班級: 碩專班

學生:5108056016張仲威

教授: 吳俊霖 教授

目錄

[一. 主要演算法 2](#_Toc59982590)

[二. 程式片斷 2](#_Toc59982591)

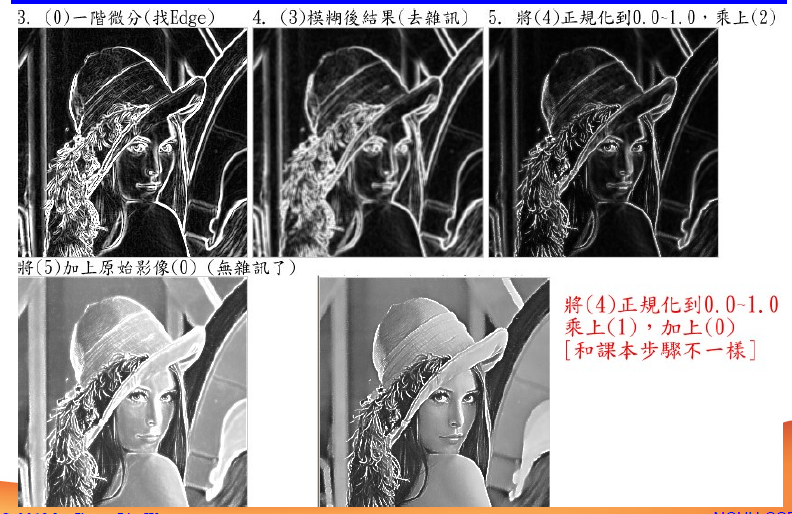
[三. 測試資料 5](#_Toc59982592)

[四. 結果呈現 5](#_Toc59982593)

[五. 討論與結論 6](#_Toc59982594)

# 主要演算法

我們自行實現的步驟如講義上



# 程式片斷

1.步驟一階微分找edge

1. # X方向一階微分
2. arr = [[ -1.,  0,   1],
3. [ -2.,  0.,  2],
4. [ -1.,  0.,  1]]
5. im3\_x = Image.new( 'RGB', size, "black")
6. pixels2 = im3\_x.load()
7. **for** i **in** range(size[0]-2):
8. **for** j **in** range(size[1]-2):
9. convolution(i+1,j+1,arr,pixels,pixels2)
11. # Ｙ方向一階微分
12. arr = [[  1.,  2,   1],
13. [  0.,  0.,  0],
14. [ -1., -2., -1]]
15. im3\_y = Image.new( 'RGB', size, "black")
16. pixels3 = im3\_y.load()
17. **for** i **in** range(size[0]-2):
18. **for** j **in** range(size[1]-2):
19. convolution(i+1,j+1,arr,pixels,pixels3)

2.步驟4的模糊後結果(去雜訊)

1. # 將一階微分結果模糊去雜訊
2. arr = [[ 1/9.,  1/9.,  1/9.],
3. [ 1/9.,  1/9.,  1/9.],
4. [ 1/9.,  1/9.,  1/9.]]
5. im4 = Image.new( 'RGB', size, "black")
6. pixels = im3.load()
7. pixels2 = im4.load()
8. **for** i **in** range(size[0]-2):
9. **for** j **in** range(size[1]-2):
10. convolution(i+1,j+1,arr,pixels,pixels2)

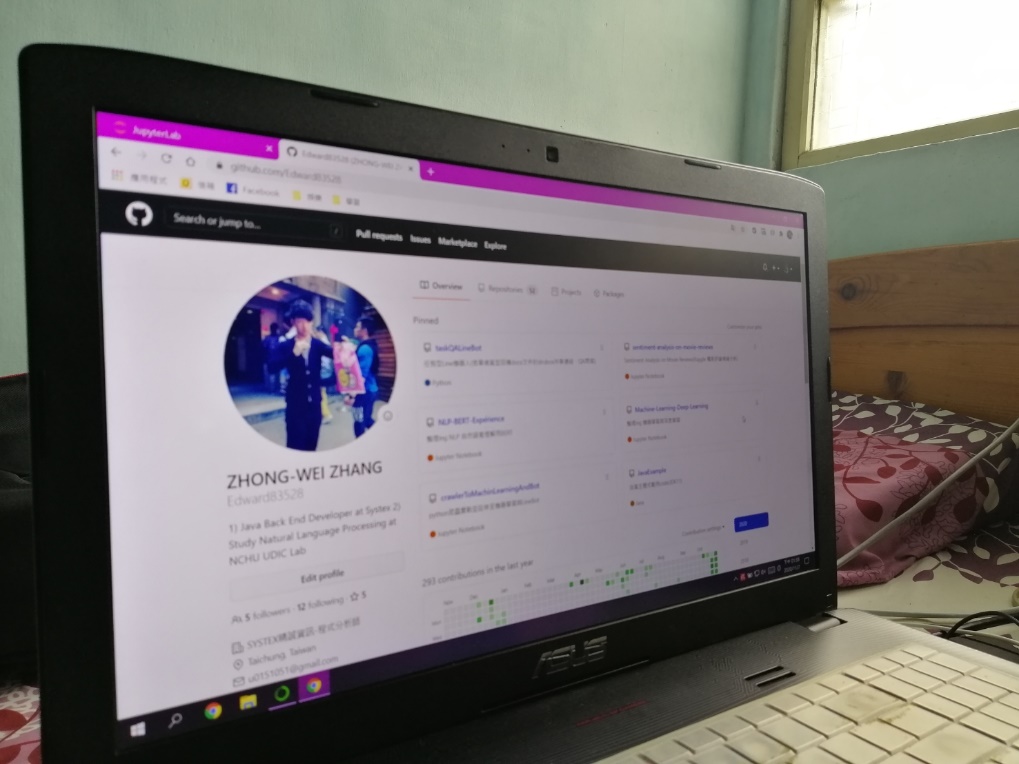
3.步驟五的正規化

1. # ＸＹ方像取絕對值相加
2. **for** x **in** range(size[0]-2):
3. **for** y **in** range(size[1]-2):
4. pixels2[x,y] = (abs(pixels2[x,y][0]),abs(pixels2[x,y][1]),abs(pixels2[x,y][2]))
5. pixels3[x,y] = (abs(pixels3[x,y][0]),abs(pixels3[x,y][1]),abs(pixels3[x,y][2]))
6. im3 = image\_add(pixels2,pixels3)
7. show("3 (0)一階微分",im3)

4.步驟6的加上原始影像

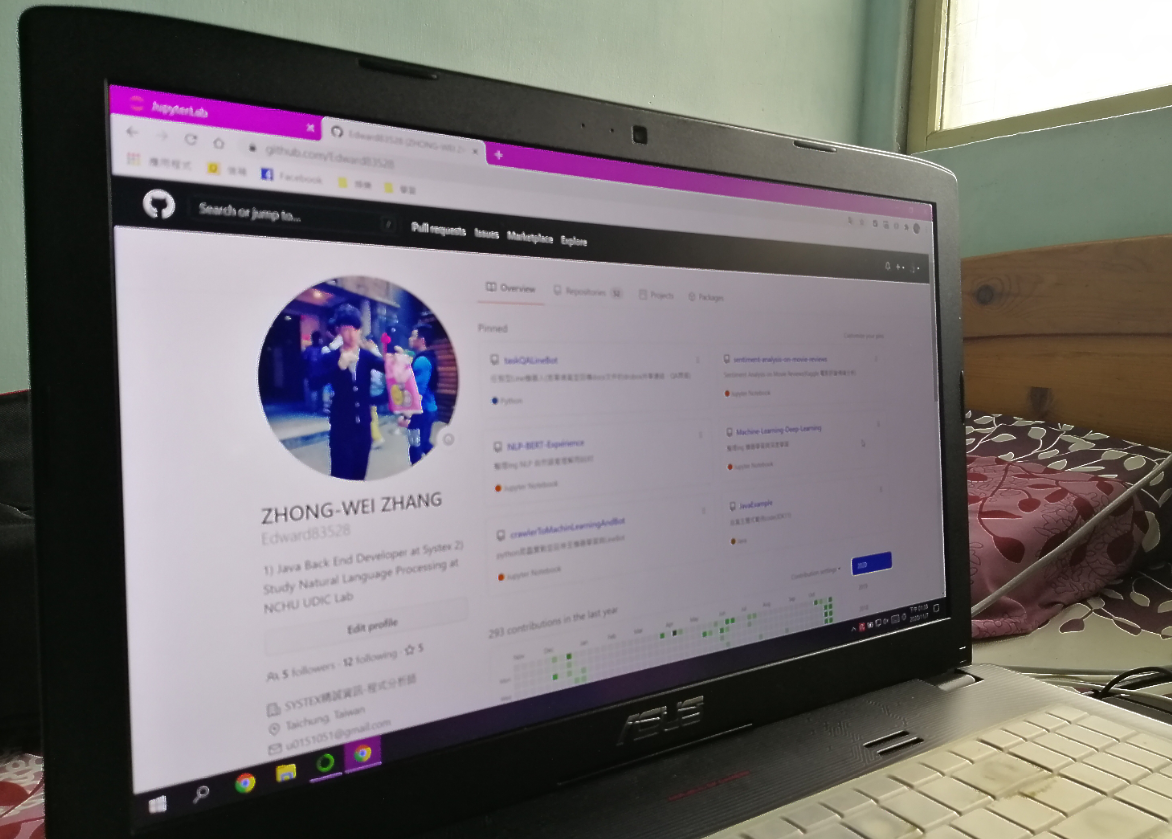
1. im6 = image\_add(im.load(),im5.load())
2. show("6 (5)加上原始影像（0）無雜訊",im6)

# 測試資料



# 結果呈現

結果的圖比原有的圖更清晰



# 討論與結論

如果以原有的平均的方法去雜訊,會連雜訊也放大,透過1階微分的方式,效果更好