**COMP9517: Computer Vision**

**2021 Term 3**

**Assignment's Report**

**zID: z5298319**

**NAME: TONG ZHE**

In the biological experiment picture, the main task of identifying image information is to emphasize the foreground and background in the picture. In other words, it means to enhance the contrast.

**Task 1**

Task 1 aims to estimate the background. In that case, the task of the code is to make sure the value of pixel of foreground tend to 0(the color tends to black), and the value of pixel of background tend to 255(the color tends to white). Get image A by max_filter from the input image I, and get the image B by min_filter from A.

Origin image I as Figure 0 shows.

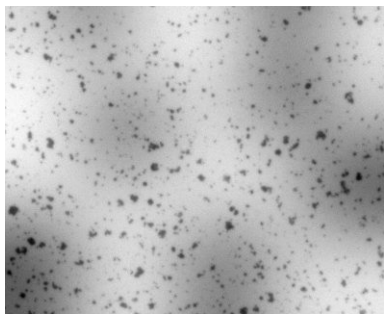

Figure 0
Image I

For the image A, the details always small than I. And with the change of the value of N, it shows different results. As Figure 1 shows, when N = 5, A get clear and smaller foreground(details). And when N = 9, the details almost disappeared as Figure 2 shows.

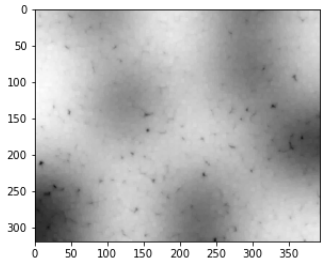


Figure 1
Image A, N=5

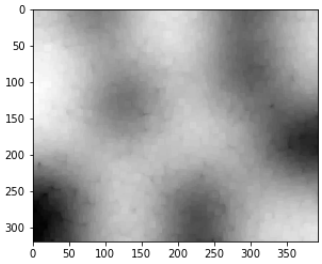


Figure 2
Image A, N=9

For the image B, comparing with A, it gets larger details. And with the change of the value of N, it also shows different results. As Figure 3 shows, when N = 5, the details in B are larger than Figure 1(almost return to origin size). And as Figure 4 shows, when N = 9, it shows less changes.

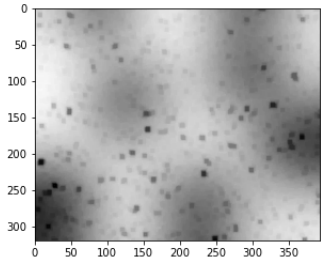

Figure 3
Image B, N=5

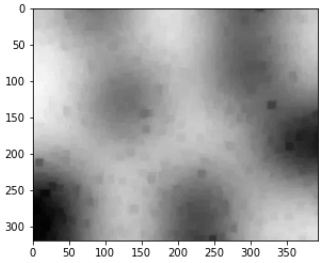

Figure 4
Image B, N=9

**Task 2**

For the task 2, it needs to use I subtract B to get the output image O. As the pic Figure 5 shows, when the N = 5, which is the smallest value, the functions can totally output the background of the picture. And as the Figure 6 shows, when the N = 9, which is the best suitable value, the functions can output the background and clear foreground. Obviously, in that case, when the value of N reduces, the background become clearer. And if the value of N increase, the picture can keep clearer foreground(details) than lower N.

In that case, it likely to appear a problem that if the subtracted values are not in the range of 0 to 255. It has two methods to deal with. One is use cv2.subtract() function, which default that the form of value is uint8, which can prevent the data roll back. Another is to use an ‘if’ clause to judgment the situation.

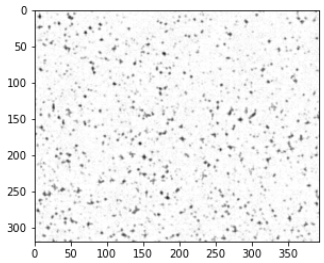

Figure 5
Image O, N=5

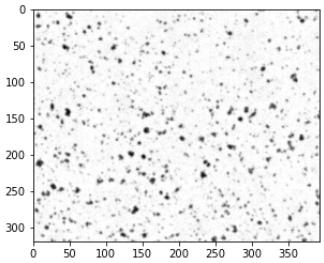

Figure 6
Image O, N=9

**Task 3**

In task 1 and 2, it solves the problems that Background Estimation and Background Subtraction. But it only suitable for image which is bright background and dark foreground. Because max_filter can bright the background and minimize the size of dark ‘detail’ and min_filter is opposite. So that in task 3, for the image with dark background and bright foreground, the input image I should get the image A by min_filter first, then using A get image B by max_filter. In the end, using I subtract B to get the output image O. And in coding, an use M==0 or M==1 to recognize which kind of picture will be chosen.

With tests, N = 33 is suitable for the ‘Cell.png’. Figure 7 is the origin picture I, which is a dark image with few bright parts.

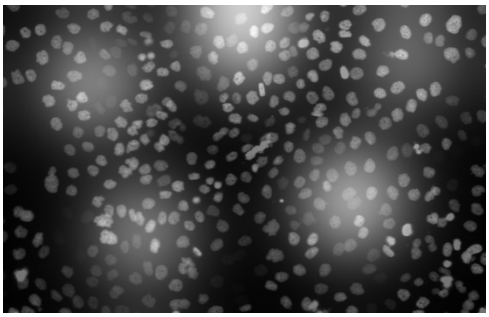

Figure 7
Image I

For the image A, as Figure 8 shows, min_filter get the bright parts in image I. And for the image B, as Figure 9 shows, it bright the particles.

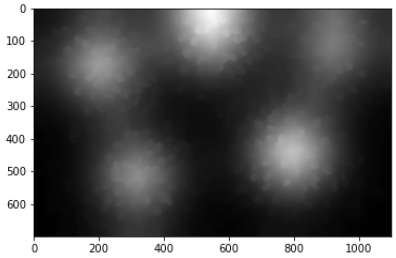


Figure 8
Image A

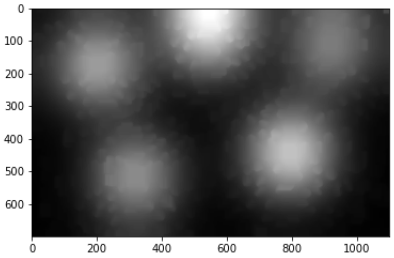


Figure 9
Image B

In the end, with the subtract, a picture without ‘bright shading’ can be gotten. As Figure 10 shows.

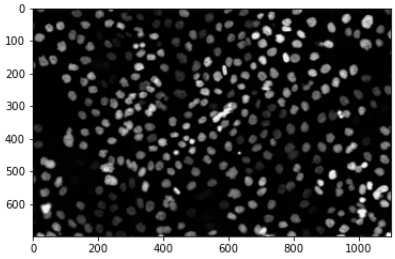


Figure 10
Image O