COMP9517 Assignment 1

Yueying Li, z5212833

Task1

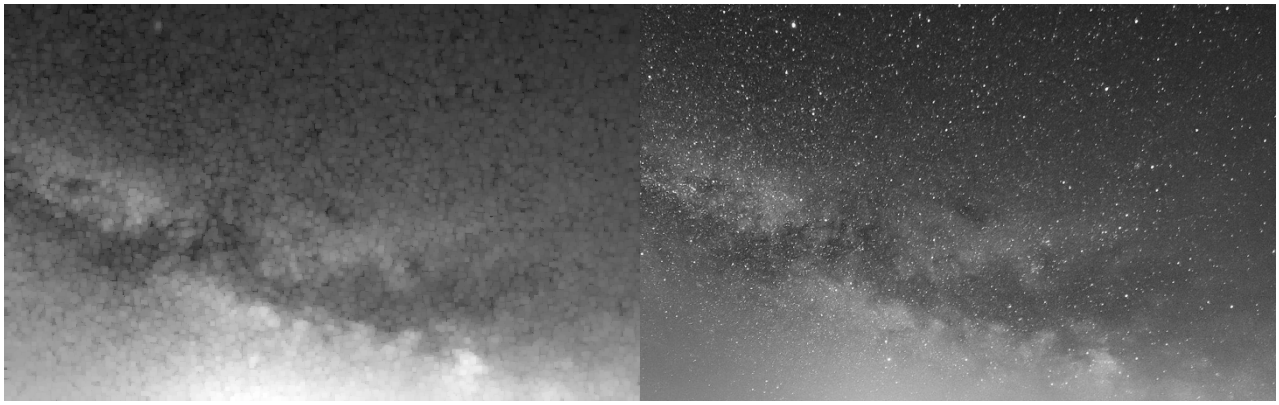
The goal of this assignment is to remove background shading for different types of images in different science projects such as astronomy and biology. Generally, the approach is to make accurate estimate of the background by perform mathematical operations then subtract the background on the given image.

For Milkyway.png, before going through the pixels of image one by one, a padding with width N//2 and value 0 is added since the background color is black. So that we can take every pixel in to account without losing edges. Next, min_filter is applied with different value of N.

To find the smallest value of N that can cause the stars disappear, I loop through N from 3 to 17 and find out that 7 is the most appropriate value of N.

This is N=3(left) and N=5(right) min_filtered result, there are still stars visible in the image. For N = 5, only one star is left.

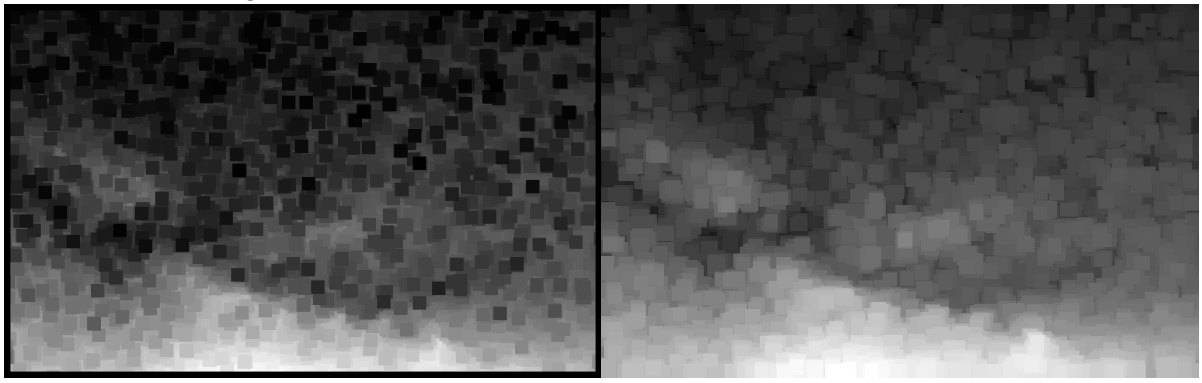
This is image A for N=7, all the stars are visually disappeared.



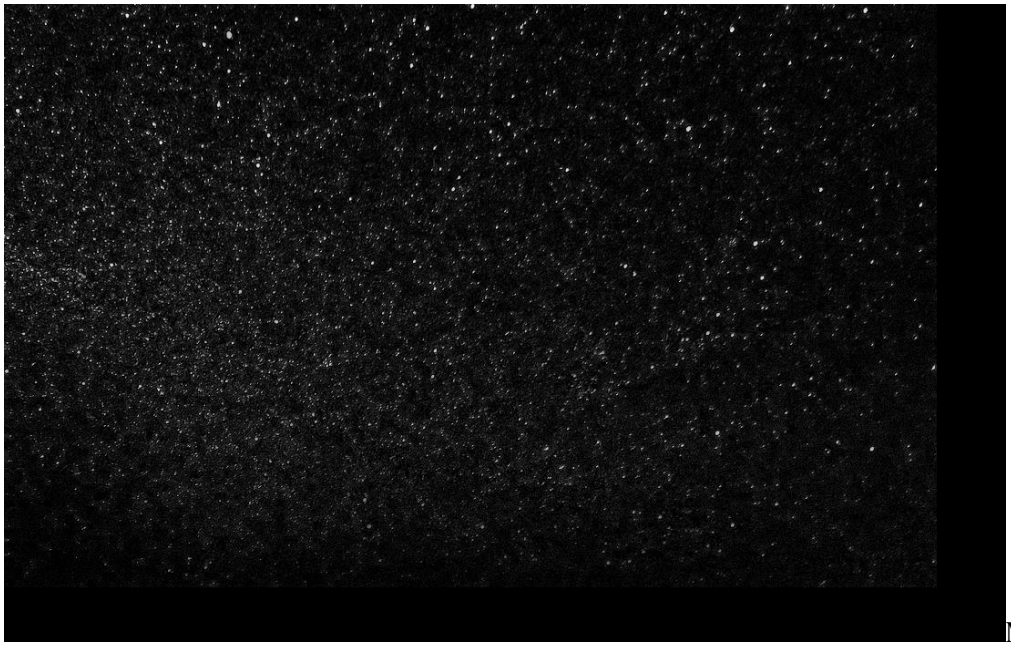
Next, a width N//2 padding is added on image A after minimum filter. Left hand side is image B (background after max_filtered with N =7. By comparing with the original image on the right-hand side, the background imageB is well estimated.

The effect of taking larger values is that the removed area for each star become larger and the background we got is not the actual background we want to subtract.

For example, this is image A for N =23, and image B on the right. It can be seen that not only stars but also background is disappeared in A and almost all pixels are with same valued in B, we could not see a clear background in this case. So that we could not use larger values. The shading patterns come back when N becomes larger.



Task B



Milkyway Output

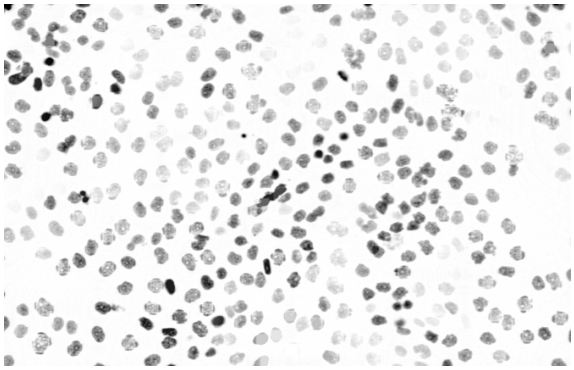
Task3

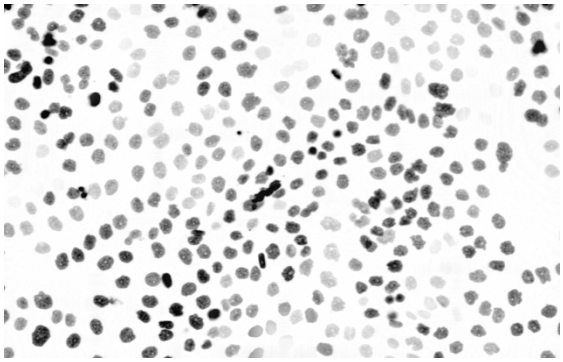
For other images with bright background such as Cells.png, the shading patterns that needed to be removed is dark, thus the algorithm is extended for these types of images with arbitrary sizes. Since the cells is dark in the image, min_filter is not able to remove the cells since it would change every pixel to the minimum value within the neighborhood which is black thus the whole background would be changed.

Max_filter should be used in the first step to remove the dark cells into maximum values in the neighbors which most likely to be bright. Then a min_filtered correction is applied to get a more precise background shadings of cells. Moreover, cells are dark with low pixel values, if we only apply simple subtraction in the end, some pixels will result in negative since the background pixel value is larger than the cells. Thus, it requires add 255.

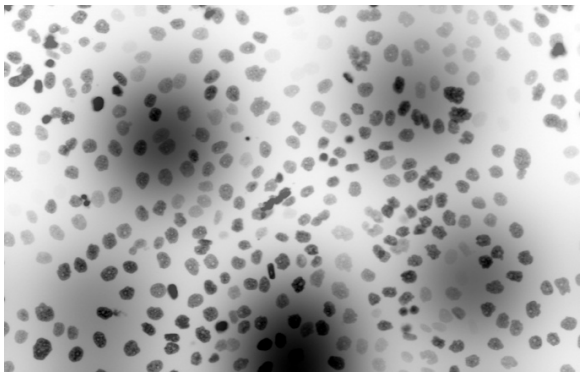
In order to find the good value of N, I loop through N from 3, the output image is almost all white until N = 17.

Below is the output for N = 19, there are still some cells that is white and not clear enough to be seen. So, 19 is not the best N to choose.



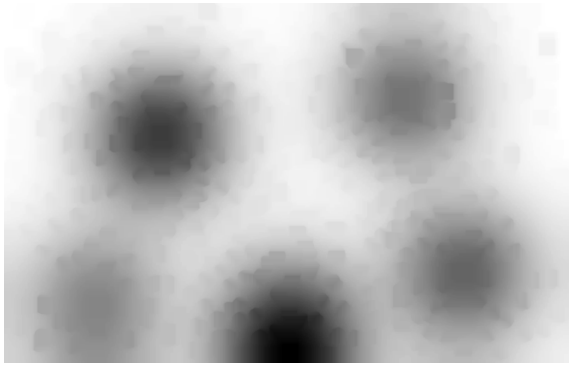


N = 29 output



Cells.png

After checking the output from 3 to 31, I would choose N = 29 for Cells.png, because all cells in the image could be clearly identified and the color of them are the closest to input image by comparing with lower N values. Besides, later output from larger N values do not have further improvement comparing with this one.(Other outputs are shown in ipynb)



N = 29 Image B