COMP9517: Computer Vision
2020 Term 2

Assignment 1

ZID: z5225807

Task 1:

[task1 and task2 are included in one python file – assignment1.py]

Background estimation Open the given image Particles.png has a background shading pattern that needs to be removed. The first step to get rid of the shading is to make an accurate estimate of the background of the image.

The first half of the Algorithm has following steps:

1. create a new image, image A, with the same size (number of pixel rows and columns) as the input image(I).
2. The main logic, where algorithm goes through each pixel of I one by one, and for each pixel in position (x, y) it finds the maximum value in in an N x N neighbourhood around that pixel
3. write that maximum value in the corresponding pixel location (x, y) in A. The resulting image A is called a max-filtered image of input image I.

to find the neighbourhood pixels (including the pixel at (x, y)), I have implemented sliding window with window size 13*13 (N*N). I decided the value of N by trial and error method. Till N=12 I could detect darker pixels in image A. but with N=13 all the dark spots vanished. So that's Max_filtering.

Reference : blogs

Second Half:

1. create another image, image B, of the same size as I and A.
2. Now the algorithm goes through each pixel of A one by one, and for each pixel in position (x, y) algorithm finds the minimum value in an N x N neighbourhood around that pixel. N value for both max-filtered and min filtered are same.
3. write that minimum value in position (x, y) in B. The resulting image B is called a min-filtered image of intermediate image A.

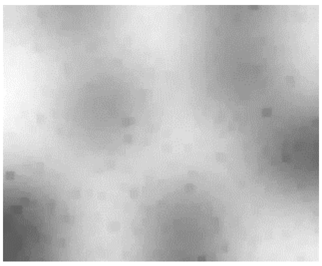

fig 1: Image B

Task 2:

In this task as the algorithm can estimate the background B of an image I, removing the shading artifacts from I can be done simply by subtracting B pixel by pixel from I which results in the output image O (fig 2.2)

The final output image is then normalised using cv2.normalise() to get better contrast and clarity of the image (fig 2.1)

Reference : StackOverflow

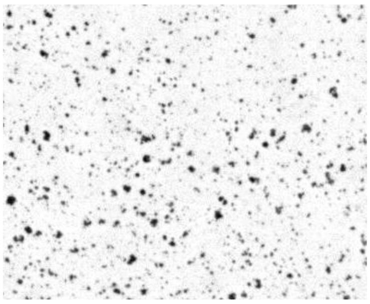

Fig 2.1: Normalised output

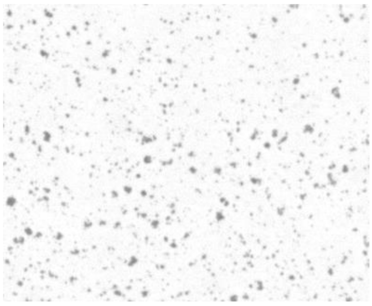

Fig 2.2: Final output without normalization

Task 3:

[Code is included as a separate python file- task3.py]

With another free parameter, named M, algorithm lets to choose between min-filtering and max-filtering.

- If the user sets M = 0, the algorithm should perform max-filtering (image I to A), then min-filtering (image A to B), then subtraction (O = I – B).
- If the user sets M = 1, the algorithm should perform first min-filtering, then max-filtering, then subtraction.

Particle.png should perform max-filtering first and then min-filtering because the background is brighter than the foreground pixels. After removing the darkest areas and just focusing on the background we then be able to get foreground pixels by just subtracting background pixel from input image. The resulting output will have higher pixel values only which seems like the image is having higher contrast.

Cells.png should perform min-filtering and then max-filtering because the background is darker than the foreground. So, we remove higher pixel values to focus on foreground.

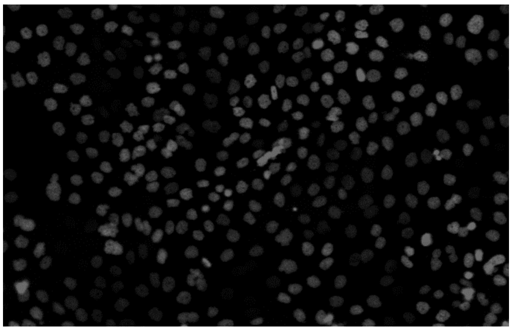

fig 3.1: final output of task 3

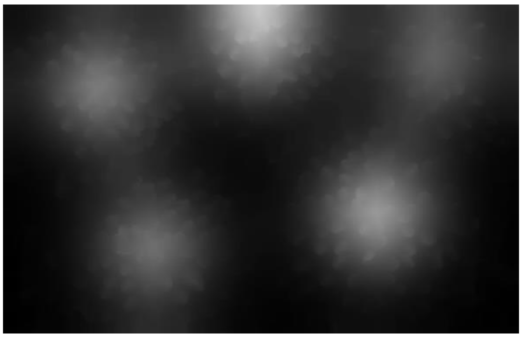

fig 3.2: image B