

Image Processing Task

Abstract:	1
Requirements:.....	1
Document Description:.....	1
Block Diagram:	2
Completed Tasks:	2
Constraints:	2
Filters Description:	2
Average Filter:	2
Max Filter:	2
References:	3

Abstract:

A software is required that takes the average of the loaded images and then apply a max filter on the averaged images. Finally store the coordinates into the text file. So, our step is to load all image files within a folder into memory and processes one after the other. Convert the attached loaded files to gray scale. Apply average algorithm according to the kernel size after that takes the maximal value and finally get the positions of the lines in the image. The processing deliver all image pixel coordinates where the pixel brightness of the image after it has been averaged over +/-5 pixels horizontally is maximal within a neighborhoods of horizontally +/- 10 pixels.

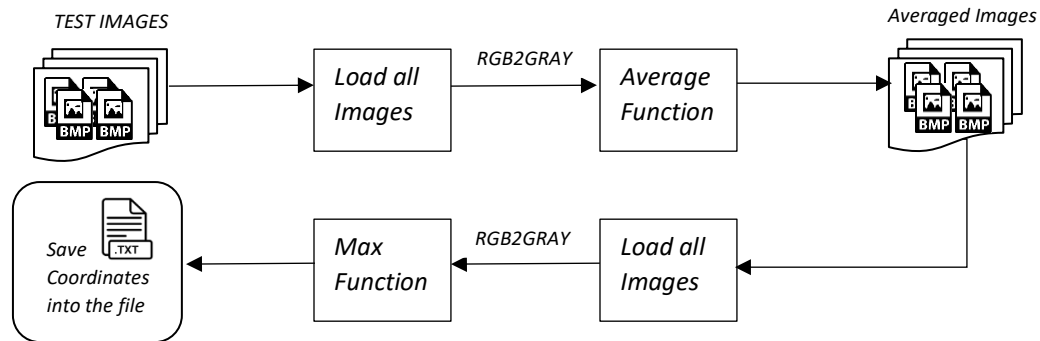
Requirements:

- Load all Images to memory in Matrix form.
- Convert the image format. (RGB2GRAY).
- Apply average filter according to the kernel size and filtration technique.
- Find Maximal value according to the kernel size and only when pixel value is greater than 15 from the neighborhoods according to the distance chosen.
- Save the coordinates in the text file.
- Check whether the correct maxima have been found.
- All the matrix processing is done on GPU using CUDA Kernel by applying vector techniques.

Document Description:

- How the problem was solved.
- Which resources were used?
- How the algorithm works.
- Any problems encountered and how they were solved.

Block Diagram:



Completed Tasks:

- Load all Images to memory in Matrix form.
- Convert the image format. (RGB2GRAY).
- Apply average filter according to the kernel size 3 and filtration technique 1D box filter.
- Find Maximal value only when pixel value is greater than 15 from the neighborhoods according to the distance chosen.
- Save the coordinates in the text file.

Constraints:

- CUDA compatible GPU is required.

Filters Description:

According to my research and previous understanding. To create any filter we have to define a kernel that contains the required filtration technique (Convolution filter, Average filter, blur, etc...). These filters have dimensions according to the requirement. However, here the requirement is 1D box filter that means we apply a filter on one dimension that is horizontally.

Average Filter:

Average filter takes average of the box pixels and put the value in the center pixel.

For example, we have pixel values P1-P10 and we want to apply an average filter using 1D box filter with kernel size 3.

To calculate the value of $P2 = (P1+P2+P3)/3$, for $P3 = (P2+P3+P4)/3$ and so on for whole array.

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
----	----	----	----	----	----	----	----	----	-----

Function Name: `void average_fn(vector< Mat >& in_img);`

Max Filter:

Max filter finds the max number from the neighborhoods of the box pixels and then compare the results with the center pixel and if the center pixel is greater than neighborhoods so record the value into the file.

For example, we have pixel values P1-P21 and we want to apply a max filter with kernel size $-/+10$.

To calculate the value of $PC = (\max(P1-P10) \leq PC \leq \max(P12-P21))$ and so on for whole array.

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	PC	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21
----	----	----	----	----	----	----	----	----	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Function Name: `void process_images(vector< Mat >& img_in);`

References:

1. <http://tech-algorithm.com/articles/boxfiltering/#:~:text=Box%20filtering%20is%20basically%20an,to%20produce%20a%20third%20one.>
2. <https://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm>
3. https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_peak_local_max.html
4. <https://www.youtube.com/watch?v=ZoaEDbivmOE>
5. https://www.youtube.com/watch?v=C_zFhWdM4ic