FACULTY OF COMPUTING

SEMESTER 1/2223

SECV3213- FUNDAMENTAL OF IMAGE PROCESSING

SECTION 1

Assignment 3: **Image Thresholding**

Lecturer: Dr Md Sah Hj Salam

Group 3

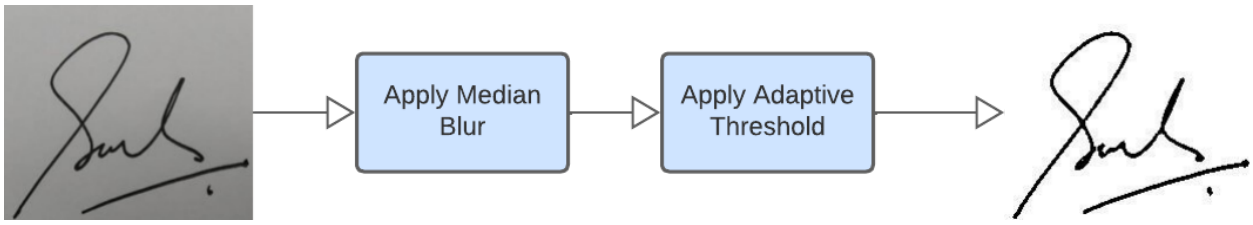
Name	**Matric No**
Fatin Aimi Ayuni Binti Affindy	A20EC0190
Bilkis Musa	A20EC0233

TABLE OF CONTENT

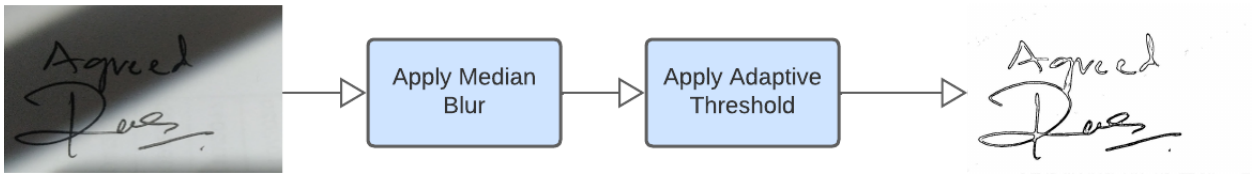
Introduction

Image thresholding is a method of manipulating an image to make it easier to be analyzed by the computer. It converts the image which has many gray levels to another image that has fewer gray levels. This method is usually used to differentiate between the parts of interest and not interest. This can be used to detect the details of the image because this method can focus on the region that the user is interested in. There are different ways of implementing threshold such as using basic thresholding, Otsu threshold and adaptive threshold. To do the segmentation, different combinations of functions are used to produce a good output. In this assignment, we will be dealing with black and white images with and without shadows issues and colored images.

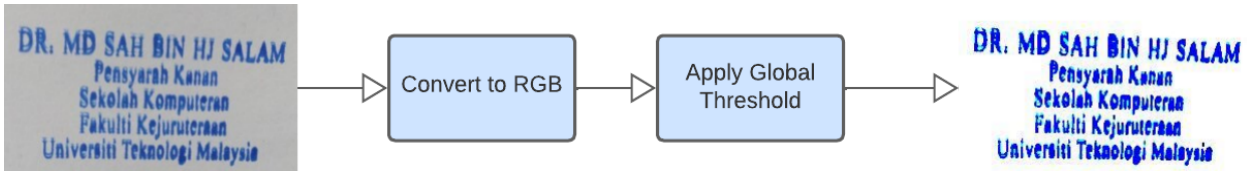
Process flow/Pipeline



Black and white image without shadow



Black and white image with shadow



Colored image

Proposed Solution

To deal with black and white images, we apply median blur and adaptive thresholding. The median blur is used to reduce the noise in the image so it can be segmented better in the adaptive thresholding functions. We use adaptive thresholding for black and white images to deal with the shadow issue in the sample images given. As for colored images, we convert the image read to RGB scale to preserve the color of the image and to make sure the output of the thresholding processes still contain the color from the input image. Then, we apply basic thresholding for the image.

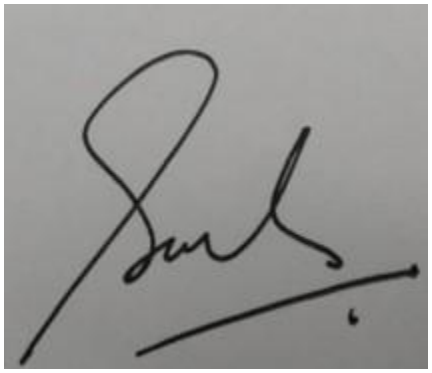
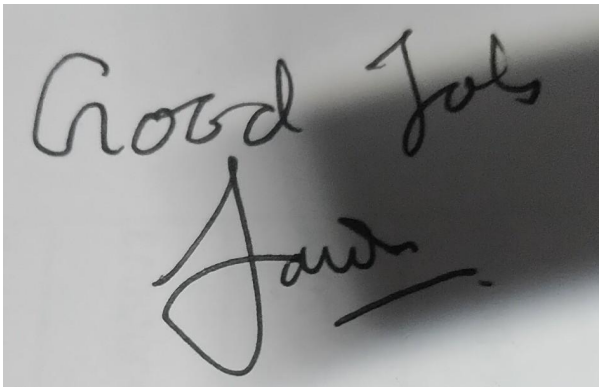
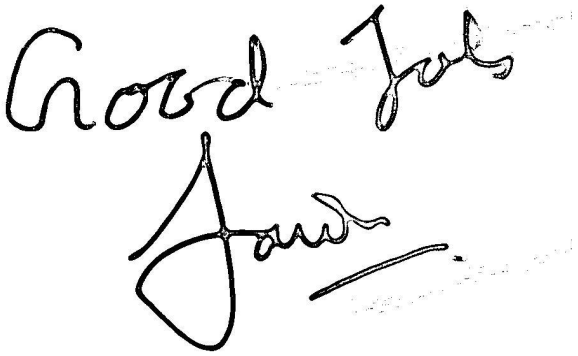
In the user interface, we provided two textboxes for the user to change the kernel size and the threshold value that will be used in black and white image thresholding and colored image thresholding respectively. For the black and white images, we have set a value of 255 as the default value for the adaptive thresholding and cannot be changed. For the median blur, we have set 7 as the default kernel size to reduce the noise in the image and it can be changed by the user using the provided text box and save button. The user can only change the kernel size to reduce the noise of the black and white image and the threshold value cannot be changed. The cv function used for black and white image is as follow:

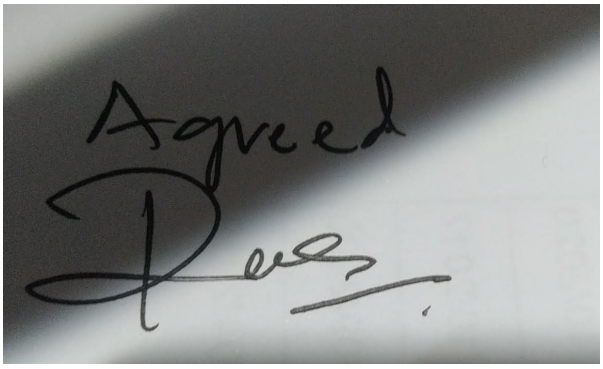
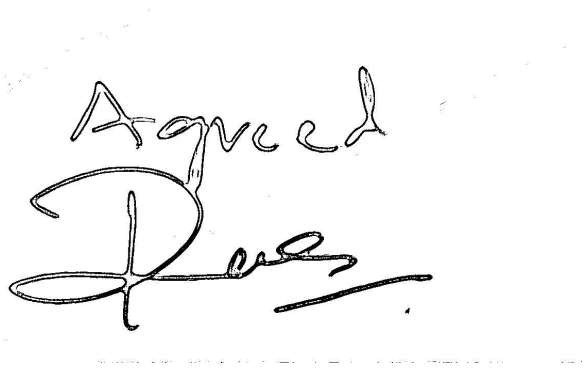
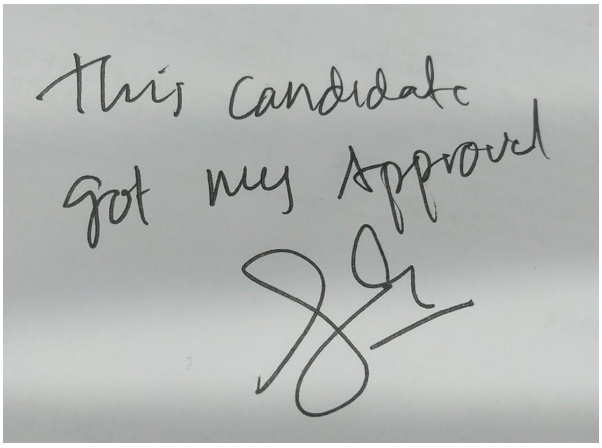
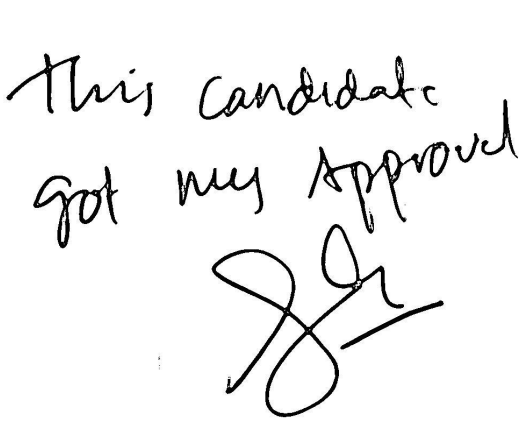
- cv.adaptiveThreshold(src, value, method1, method2, pixel neighborhood size, c)
    - src: blurred image
    - value: the threshold value
    - method1: thresholding method. In this solution, we used `cv.ADAPTIVE_THRESH_GAUSSIAN_C`
    - method2: thresholding method, almost the same as the global threshold. In this solution, we used `cv.THRESH_BINARY_INV`
    - pixel neighborhood size: neighborhood size to compute the threshold value
    - c: constant to improve the result

As for the colored image, the user can change the threshold value used to see which value will produce better output. The user can manipulate the threshold value used for the thresholding by using the text box given and save it using the save button so that the output will be updated using the new given value. For colored image, the threshold function that we used is as follow:

- cv.threshold(src, value, maxval, type)
    - src: source image
    - value: threshold value
    - maxval: maximum value
    - type: type of threshold. In this solution, we used `cv.THRESH_BINARY`

Input & Output Sample

Input	Output
	 Median Blur: 5 Threshold: 255
	 Threshold: 127
	 Median Blur: 13 Threshold: 255

	 Median Blur: 15 Threshold: 255
	 Median Blur: 9 Threshold: 255

Discussion

We have tested the coding using four black and white images and one colored image. From the four black and white images, two of those have shadows issue. The first sample is a sample of using black and white images without noise. For this image, we used kernel size of 5 because there isn't much noise in the picture, thus, to avoid losing more details, we used a smaller kernel size than the default one that we declared which is 7. The threshold value for the image used is 255 which is the default value that we assigned in the code. This sample uses adaptive threshold, but the benefit of using adaptive thresholding cannot really be seen in the first sample because the first sample has not much noise. To see the benefit of using a bigger kernel size to remove the noise in the median blur function, we can see the output of the third and fourth sample. It can be seen that using bigger kernel size can overcome the issue of shadows. This is because in the black and white function, adaptive threshold is used rather than Otsu and basic threshold.

Adaptive threshold is a method of taking the neighborhood pixels to assign the threshold value onto the pixels rather than depending on one pixel to assign the threshold value. This method is better than the other because it can overcome the poor lighting and shadow issues in a picture. Implementing adaptive threshold is the best way to apply thresholding onto an image to make sure the issues like poor lighting and shadow issue can be catered.

As for the colored image, the second sample indicates the result of using basic thresholding onto the image. Before applying threshold onto the image, the image is first converted to RGB scale to make sure that the colors are preserved in the result after applying threshold. The sample indicates the value of thresholding used for this sample is 127, which is the default value that we assigned in the code. This value has the most suitable results compared to other values. However, the user can see the difference between different threshold values by utilizing the text box given. The higher the threshold value applied to the colored image, the bolder the output would be.

Conclusion

To conclude, each threshold method gives almost the same output but some are better than others. For instance, the local/adaptive threshold is better than the other two because it is better implemented on images with poor lighting and shadows. It gets a threshold for different regions in the same image which have varying lighting. By applying an adaptive threshold with the input image that has shadow, the shadow will be removed, and the proper output image is achieved.

Video presentation link:

https://youtu.be/F0TcsJ8nhP4

References

GeeksforGeeks. (2020, December 11). *How to Get the Input From Tkinter Text Box*. https://www.geeksforgeeks.org/how-to-get-the-input-from-tkinter-text-box/

GeeksforGeeks. (2021, November 23). *How to change border color in Tkinter widget*. https://www.geeksforgeeks.org/how-to-change-border-color-in-tkinter-widget/

GeeksforGeeks. (2023, January 4). *Python GUI tkinter*. https://www.geeksforgeeks.org/python-gui-tkinter/

How to display an image from a numpy array in tkinter? (2018, November 14). Stack Overflow. https://stackoverflow.com/questions/53308708/how-to-display-an-image-from-a-numpy-array-in-tkinter

How to get the input from the Tkinter Text Widget. (n.d.). https://www.tutorialspoint.com/how-to-get-the-input-from-the-tkinter-text-widget

how to save images with the save button on the tkinter in python. (2019, July 15). Stack Overflow. https://stackoverflow.com/questions/57033158/how-to-save-images-with-the-save-button-on-the-tkinter-in-python

Murmu, N. (n.d.). *Basics For Displaying Image In Tkinter Python*. https://www.c-sharpcorner.com/blogs/basics-for-displaying-image-in-tkinter-python

OpenCV: Image Thresholding. (n.d.). https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html

plus2net.com. (2019a, February 5). *Browse Upload & Display Image in Tkinter*. https://www.plus2net.com/python/tkinter-filedialog-upload-display.php

plus2net.com. (2019b, February 5). *Python tkinter image resize*. https://www.plus2net.com/python/tkinter-image-resize.php

Python - Tkinter Text. (n.d.). https://www.tutorialspoint.com/python/tk_text.htm

Rosebrock, A. (2021a, April 17). *How to Display a Matplotlib RGB Image*. PyImageSearch. https://pyimagesearch.com/2014/11/03/display-matplotlib-rgb-image/

Rosebrock, A. (2021b, May 9). *Adaptive Thresholding with OpenCV (cv2.adaptiveThreshold)*. PyImageSearch. https://pyimagesearch.com/2021/05/12/adaptive-thresholding-with-opencv-cv2-adaptivethreshold/

Thresholding – Image Processing with Python. (n.d.). Data Carpentry - Image Processing With Python. https://datacarpentry.org/image-processing/07-thresholding/