SECV3213-01 ASAS PEMPROSESAN IMEJ (FUNDAMENTAL OF IMAGE PROCESSING)

ASSIGNMENT 3

GROUP 4

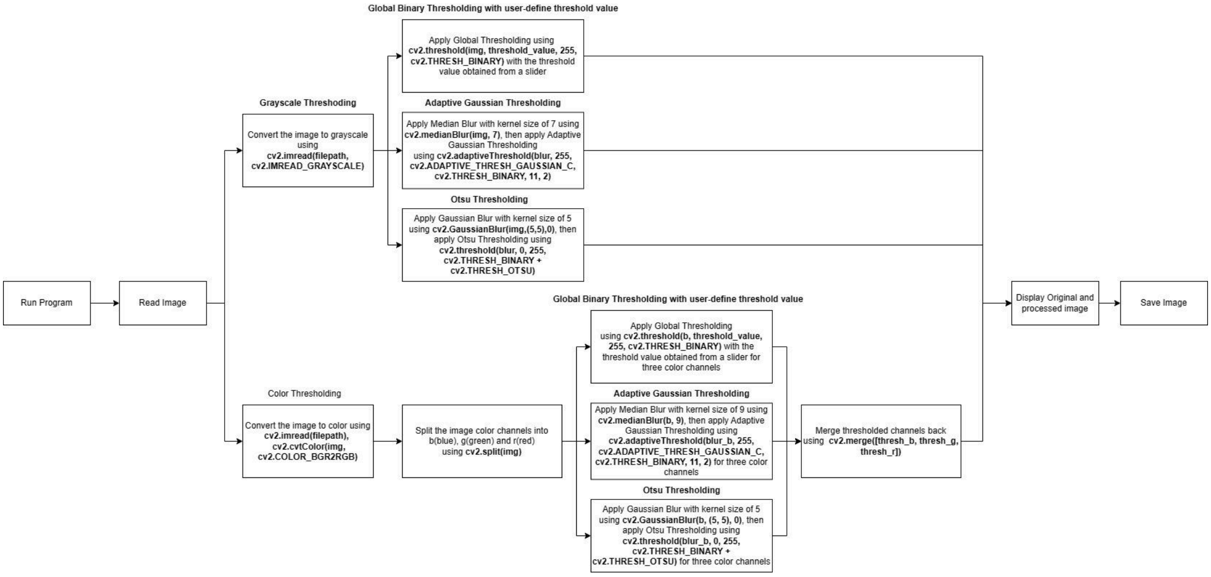
Name	Matric No
Muhammad Akashah Bin Bahar	A21EC0066
Lew Chin Hong	A21EC0044
Yeo Chun Teck	A21EC0148

Video Link: https://youtu.be/_5ShXAdk07s

Lecturer: Dr Md. Sah Bin Hj. Salam

Proposed Solution

Process Flow Diagram


Run Program
Read Image
Grayscale Threshoding
Convert the image to grayscale using cv2.imread(filepath, cv2.IMREAD_GRAYSCALE)
Global Binary Thresholding with user-define threshold value
Apply Global Thresholding using cv2.threshold(img, threshold_value, 255, cv2.THRESH_BINARY) with the threshold value obtained from a slider
Adaptive Gaussian Thresholding
Apply Median Blur with kernel size of 7 using cv2.medianBlur(img, 7), then apply Adaptive Gaussian Thresholding using cv2.adaptiveThreshold(blur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2)
Otsu Thresholding
Apply Gaussian Blur with kernel size of 5 using cv2.GaussianBlur(img,(5,5),0), then apply Otsu Thresholding using cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
Color Thresholding
Convert the image to color using cv2.imread(filepath), cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
Split the image color channels into b(blue), g(green) and r(red) using cv2.split(img)
Global Binary Thresholding with user-define threshold value
Apply Global Thresholding using cv2.threshold(b, threshold_value, 255, cv2.THRESH_BINARY) with the threshold value obtained from a slider for three color channels
Adaptive Gaussian Thresholding
Apply Median Blur with kernel size of 9 using cv2.medianBlur(b, 9), then apply Adaptive Gaussian Thresholding using cv2.adaptiveThreshold(blur_b, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2) for three color channels
Otsu Thresholding
Apply Gaussian Blur with kernel size of 5 using cv2.GaussianBlur(b, (5, 5), 0), then apply Otsu Thresholding using cv2.threshold(blur_b, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU) for three color channels
Merge thresholded channels back using cv2.merge([thresh_b, thresh_g, thresh_r])
Display Original and processed image
Save Image

Proposed Solution Explanation

In our proposed solution, the user would be provided a proper user interface for some application mechanisms, including:

- to choose and open an input image
- to save the output image
- to show the input and output image on the screen
- to control and set the parameter like the threshold value for certain method of image thresholding in the interactive way

To achieve these, we choose to include the Tkinter library and matplotlib library for creating the graphical user interface. In this assignment, the library is used to provide

- Colored threshold button
    - The user can click this button to choose a colored image and apply color image thresholding
    - They would be presented with various options, ranging from global thresholding in color to Otsu thresholding in color
- Colored global thresholding button
    - The user can choose to click this button to apply global thresholding in color
- Colored adaptive thresholding button

    - The user can choose to click this button to apply adaptive gaussian thresholding in color
- Colored otsu thresholding button
    - The user can choose to click this button to apply otsu thresholding in color
- Grayscale threshold button
    - The user can click this button to choose a colored image and apply grayscale image thresholding, ranging from global thresholding to Otsu thresholding
    - They would be presented with various options, ranging from global thresholding to Otsu thresholding
- Grayscale global thresholding button
    - The user can choose to click this button to apply global thresholding in grayscale
- Grayscale adaptive thresholding button
    - The user can choose to click this button to apply adaptive gaussian thresholding in grayscale
- Grayscale otsu thresholding button
    - The user can choose to click this button to apply otsu thresholding in grayscale
- Save Button
    - The user can click this button save the output image after desired image thresholding
- Subplots
    - displays the original image and color global thresholded image side by side
- Threshold slider
    - serves as a graphical user interface element for the user to interactively adjust the threshold value in the image thresholding process in the case of global thresholding
    - This slider has a value from 0 to 255, which is the range of 8-bits pixel values

Here is the process flow explanation

1. Read Image
    a. The user choose an image to process image thresholding

2. Grayscale Thresholding
    a. The chosen image would be converted to grayscale image
    b. The user would be provided three grayscale thresholding options:
        i. Global Binary Thresholding with user-define threshold value
        ii. Adaptive Gaussian Thresholding
        iii. Otsu Thresholding

3. Color Thresholding
    a. The chosen image would be converted to the colored image
    b. The user would be provided three color thresholding options
        i. Global Binary Thresholding with user-define threshold value

        ii. Adaptive Gaussian Thresholding
        iii. Otsu Thresholding

If the user apply Grayscale Thresholding

4. Global Binary Thresholding with user-define threshold value
    a. A slider would be displayed, it has a range from 0 to 255
    b. Apply Global Thresholding using cv2. threshold(img, threshold_value, 255, cv2.THRESH_BINARY) with the threshold value obtained from a slider
    c. The user could move the slider to adjust the threshold value(from 0 to 255)
    d. When the slider is on moved, the method that is associated to Global Binary Thresholding would be called
    e. 0 is the initial value, the greater the value, the larger the range of pixels value to be filtered out from the original image

5. Adaptive Gaussian Thresholding
    a. Apply Median Blur with kernel size of 7 using cv2.medianBlur(img, 7)
    b. Then apply Adaptive Gaussian Thresholding using cv2.adaptiveThreshold(blur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2. THRESH_BINARY,11,2)
    c. The parameter ‘11’ refers to the block size(11x11) that determines the size of the neighborhood area used for adaptive thresholding
    d. The parameter ‘2’ refers to the constant value that is subtracted from the calculated threshold.
    e. The adaptive thresholding algorithm calculates a threshold for each pixel based on the intensity values of its local neighborhood using a Gaussian-weighted sum

6. Otsu Thresholding
    a. Apply Gaussian Blur with kernel size of 5 using cv2.GaussianBlur(img,(5,5),0)
    b. Then apply Otsu Thresholding using cv2.threshold(blur, 0, 255, cv2.THRESHOLD_BINARY + cv2.THRESHOLD_OTSU)
    c. Otsu Thresholding would find an optimal threshold of the whole image for the image thresholding

If the user apply Color Thresholding

- Three of the color thresholding option is similar to that of the grayscale thresholding, but there are some additional mechanisms to apply on the color image:

    a. Before applying the image thresholding
        i. Split the image color channels into b(blue), g(green), r(red) using cv2.split(img)

    b. After applying the image thresholding

        i. Merge threshold channels back using cv2.merge(threshold_b, threshold_g, threshold_r)

7. Display Original and processed image

8. Save image
    a. Save the processed image with the desired name.

Results of the solution

1. Grayscale Global Thresholding
   a. When Global Threshold value: 0

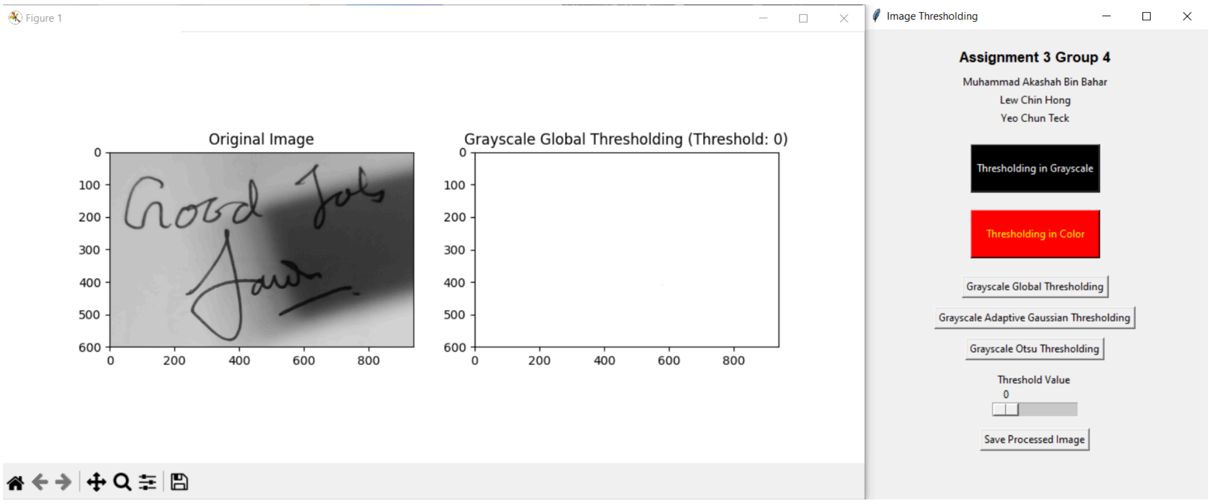


Figure 1
Original Image
Grayscale Global Thresholding (Threshold: 0)
0
100
200
300
400
500
600
0
200
400
600
800
Image Thresholding
Assignment 3 Group 4
Muhammad Akashah Bin Bahar
Lew Chin Hong
Yeo Chun Teck
Thresholding in Grayscale
Thresholding in Color
Grayscale Global Thresholding
Grayscale Adaptive Gaussian Thresholding
Grayscale Otsu Thresholding
Save Processed Image

Threshold Value
0

   b. When Global Threshold value: 120

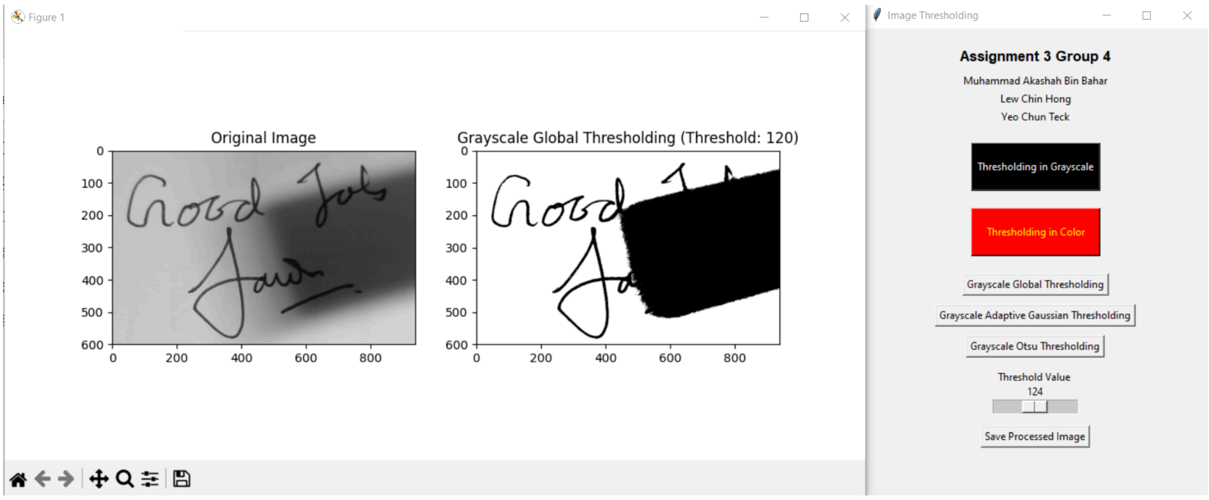


Figure 1
Original Image
Grayscale Global Thresholding (Threshold: 120)
0
100
200
300
400
500
600
0
200
400
600
800
Image Thresholding
Assignment 3 Group 4
Muhammad Akashah Bin Bahar
Lew Chin Hong
Yeo Chun Teck
Thresholding in Grayscale
Thresholding in Color
Grayscale Global Thresholding
Grayscale Adaptive Gaussian Thresholding
Grayscale Otsu Thresholding
Save Processed Image

Threshold Value
124

c. When Global Threshold value: 188

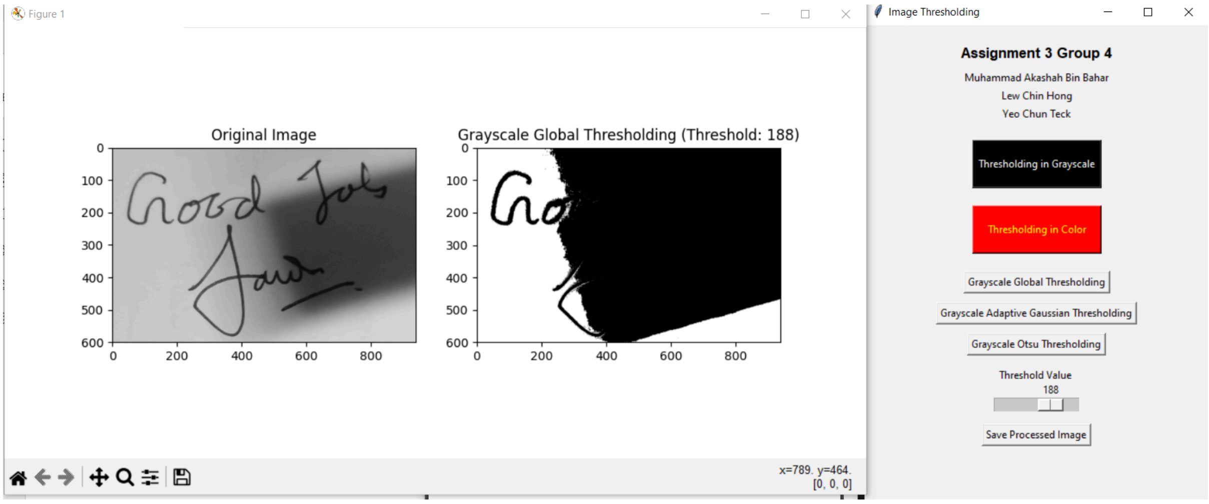


Figure 1
Original Image
Grayscale Global Thresholding (Threshold: 188)
0
100
200
300
400
500
600
0
200
400
600
800
x=789, y=464.
[0, 0, 0]
Image Thresholding
Thresholding in Grayscale
Thresholding in Color
Grayscale Global Thresholding
Grayscale Adaptive Gaussian Thresholding
Grayscale Otsu Thresholding
Save Processed Image

Assignment 3 Group 4

Muhammad Akashah Bin Bahar
Lew Chin Hong
Yeo Chun Teck

Threshold Value
188

d. When Global Threshold value : 255

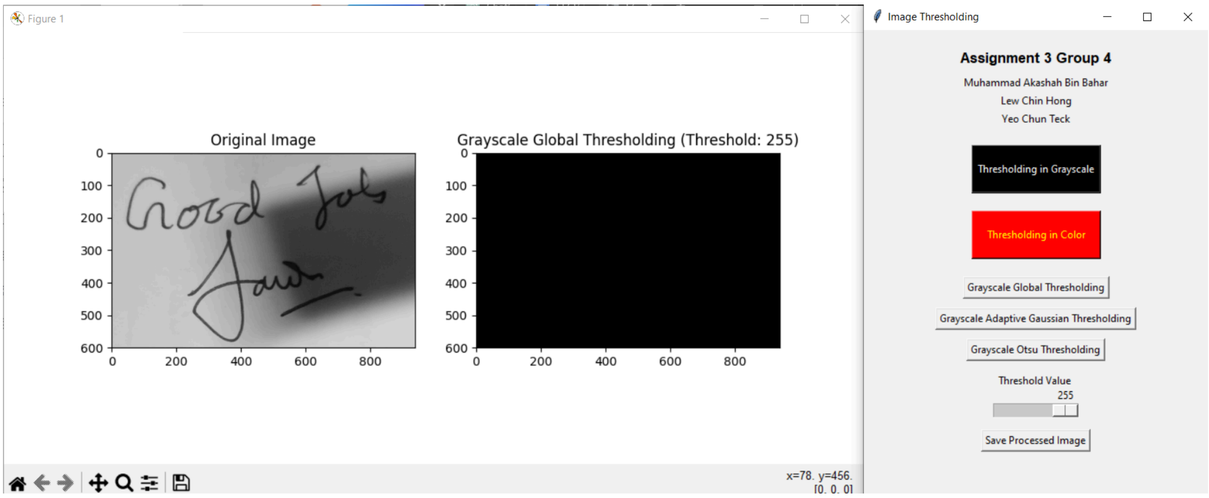


Figure 1
Original Image
Grayscale Global Thresholding (Threshold: 255)
0
100
200
300
400
500
600
0
200
400
600
800
x=78, y=456.
[0, 0, 0]
Image Thresholding
Thresholding in Grayscale
Thresholding in Color
Grayscale Global Thresholding
Grayscale Adaptive Gaussian Thresholding
Grayscale Otsu Thresholding
Save Processed Image

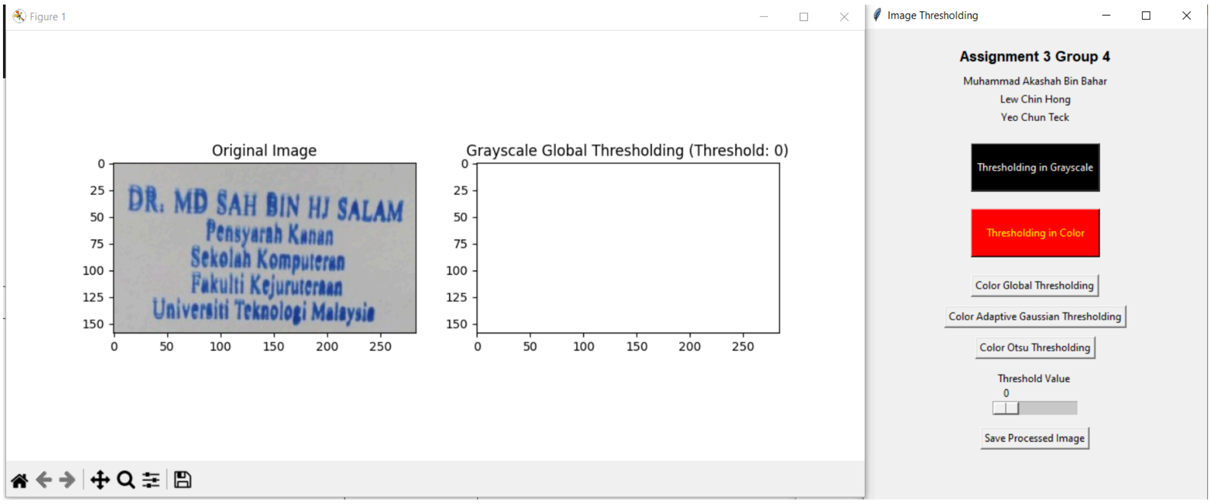
Assignment 3 Group 4

Muhammad Akashah Bin Bahar
Lew Chin Hong
Yeo Chun Teck

Threshold Value
255

2. Color Global Thresholding

a. When Global Threshold: 0

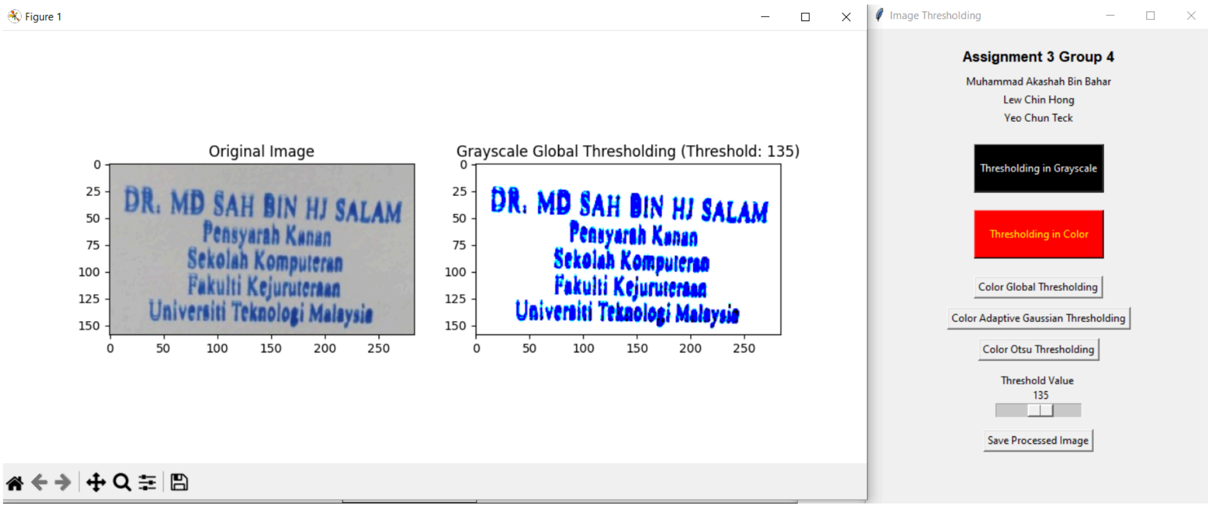


Assignment 3 Group 4

Muhammad Akashah Bin Bahar
Lew Chin Hong
Yeo Chun Teck

Threshold Value
0

b. When Global Threshold: 135

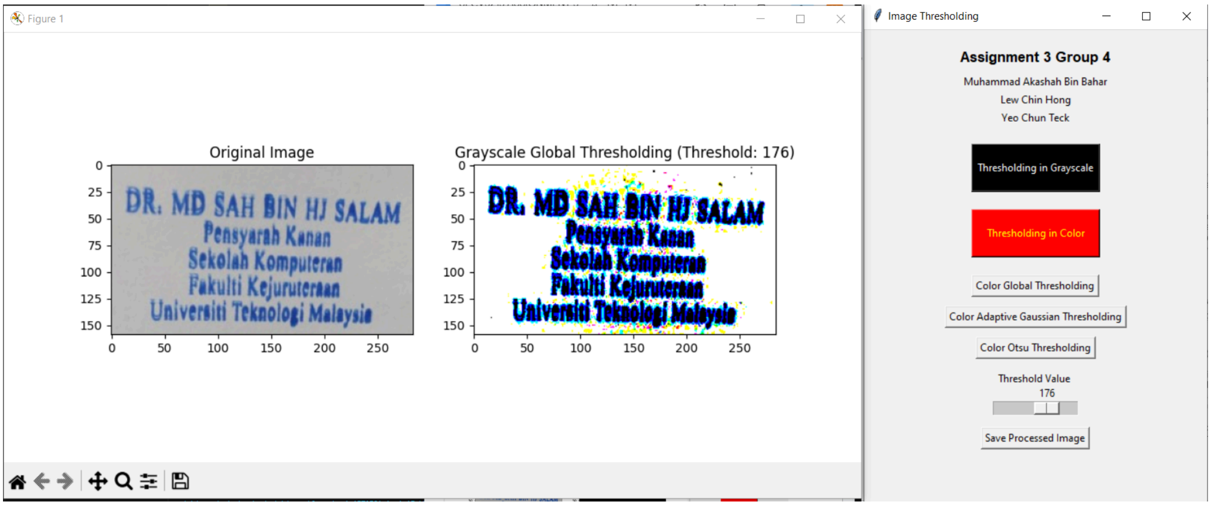


Assignment 3 Group 4

Muhammad Akashah Bin Bahar
Lew Chin Hong
Yeo Chun Teck

Threshold Value
135

c. When Global Threshold: 176

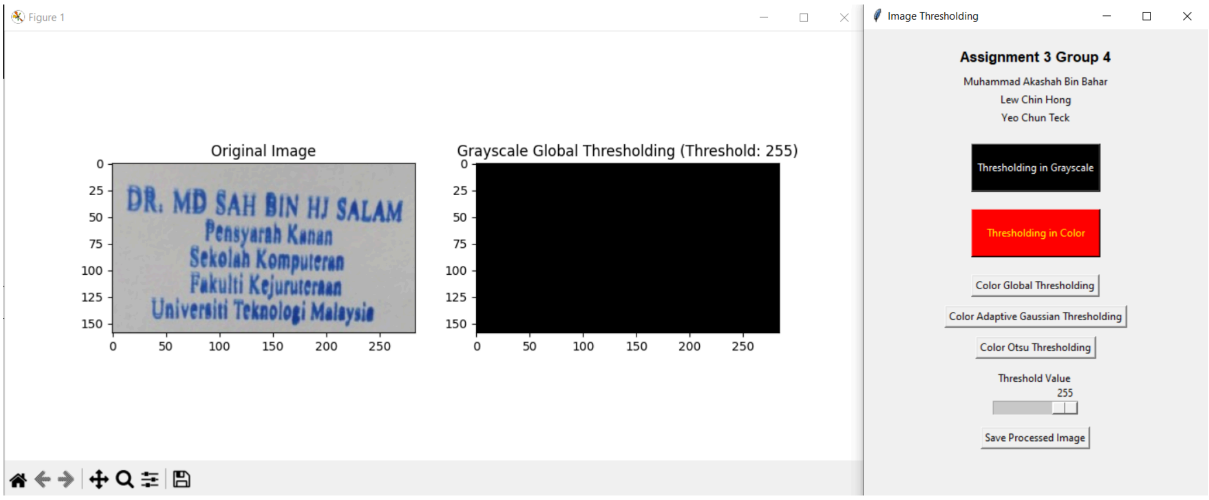


Assignment 3 Group 4

Muhammad Akashah Bin Bahar
Lew Chin Hong
Yeo Chun Teck

Threshold Value
176

d. When Global Threshold: 255



Discussion

- Grayscale Global Thresholding

When the global threshold value is set to 0, the pixel where its value is higher than and equal to 0 would be converted to 255(white). Hence, the processed image would be a white image because the foreground (calligraphy segment) of the image had been turn to 255(white)

When the global threshold value is set to 120, the whole calligraphy segment in the input image is finally shown completely, because the value of all pixels in that segment is less than the threshold value. While the background of the input image is totally white, because the pixel values in the background are more than the threshold value. This enhances the details of the calligraphy segment and enhances the brightness of the background and makes the calligraphy segment seem like it was filtered out from the input image. Threshold value = 120 seem likely to be in the optimal range of the threshold value

When the global threshold value is set to 188, We could see that some part of the processed image in a clearly white background started to turn to black. These areas turn black because their pixel values are less than 188. With this, we could say that the threshold value is out of the optimal threshold range, because the processed image start to lose some details

Until the global threshold value is set to 255, the processed image turns to black completely because all of the pixels are less than 255. Different to the situation where the global threshold value is set to 0, all of the pixels including foreground in the image are turned to 0(black)

- Color Global Thresholding

When the global threshold value is set to 0, the processed image is split into

3 color channels (red, green, blue). Similar to the situation in grayscale thresholding where the threshold value is set to 0, for each channel, the pixel where its value is more than and equal to 0 would be set to 255. After combine back those separated channels, the image would finally turn out in white (red+green+blue = white)

When the global threshold value is set to 135, similar to the situation in grayscale thresholding where the threshold value is set to 120, the foreground of the input image has enhancement in its details and the background is totally white(255). This is because the background pixels values are more than the threshold and the foreground pixels values are less than the threshold. The foreground segment can be viewed clearly, and this showed that Threshold value = 135 seems to be in the range of optimal threshold value because the foreground was like filtered successfully out of the input image

When the global threshold value is set to 176, we could observe the foreground part darkened and some color spot in the background. This is because the pixel value of some pixels in the foreground is less than the threshold value, then they turn to black (0). For the background, it had some color spots observed because the pixel value in certain channels were less than the threshold value and they turned to 0. Hence the background is not white like we see initially, because the pixel is not 255 in every color channel. Threshold value = 176 could be said out of the optimal threshold range because the processed image is not clear as what we observed when the threshold value = 135

When the global threshold value is set to 255, it is similar to the situation of grayscale thresholding where the threshold value is also set to 255. There is no pixel whose value is more than the threshold value in every channel. Hence, the whole image turned to black.

3. Grayscale Adaptive Gaussian Thresholding

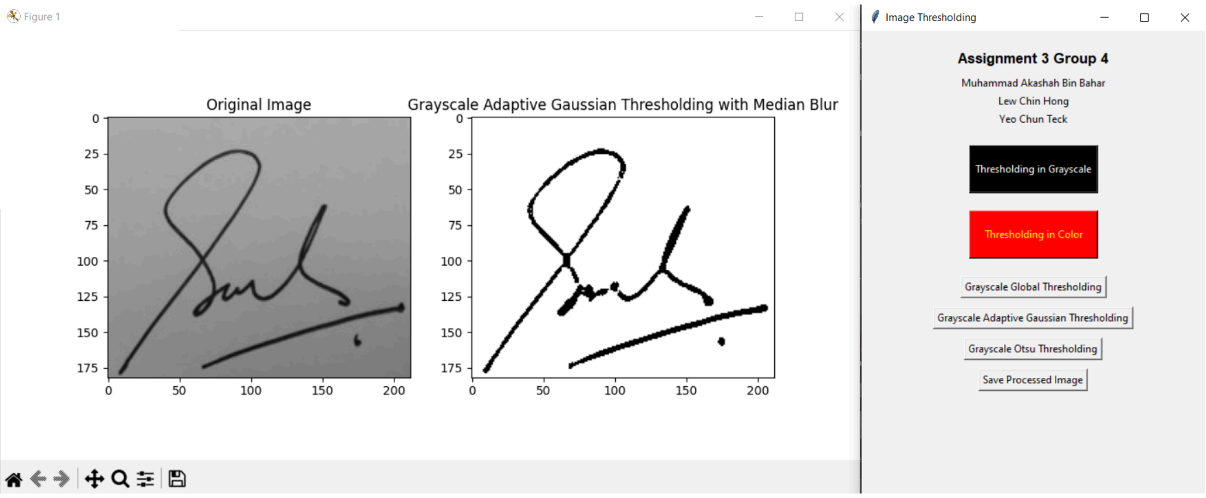
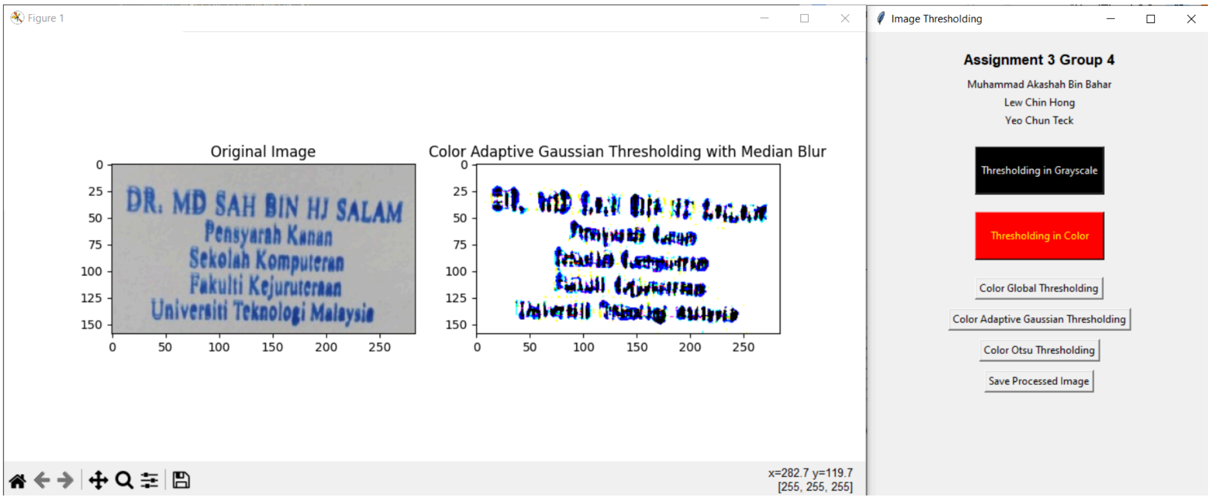


Figure 1
Original Image
Grayscale Adaptive Gaussian Thresholding with Median Blur
0
25
50
75
100
125
150
175
0
50
100
150
200
Image Thresholding
Assignment 3 Group 4
Muhammad Akashah Bin Bahar
Lew Chin Hong
Yeo Chun Teck
Thresholding in Grayscale
Thresholding in Color
Grayscale Global Thresholding
Grayscale Adaptive Gaussian Thresholding
Grayscale Otsu Thresholding
Save Processed Image

4. Color Adaptive Gaussian Thresholding



Discussion

- Grayscale Adaptive Gaussian Thresholding

The median blur is applied to the image to enhance the clarity of the background in the input image. In Grayscale Adaptive Gaussian Thresholding, the input image is partitioned into multiple blocks of the same size (11 x 11). The thresholds for different neighborhoods (blocks) depend on the Gaussian-weighted mean of each neighborhood. Consequently, different neighborhoods have distinct thresholds for local thresholding. For specific neighborhoods, local pixels with values less than the threshold are set to 0 (black), while those with values greater than the threshold are set to 255 (white). The effectiveness of the thresholding relies on factors such as the block size, the values of local pixels, and the constant subtracted from the calculated threshold. Moreover, manual adjustment, as required by the slider in Global Thresholding, is unnecessary. The algorithm automatically computes the threshold for each small region of the input image. To achieve the desired thresholding performance, parameters like block size and subtracted constant can be adjusted.

- Color Adaptive Gaussian Thresholding

Color Adaptive Gaussian Thresholding is similar to Grayscale Adaptive Gaussian Thresholding. The difference lies in the separation of color channels (Red, Green, Blue) from the input image. Adaptive Gaussian Thresholding is applied independently to each color channel of the input image, after which these channels are merged back into a color image.

5. Grayscale Otsu Thresholding

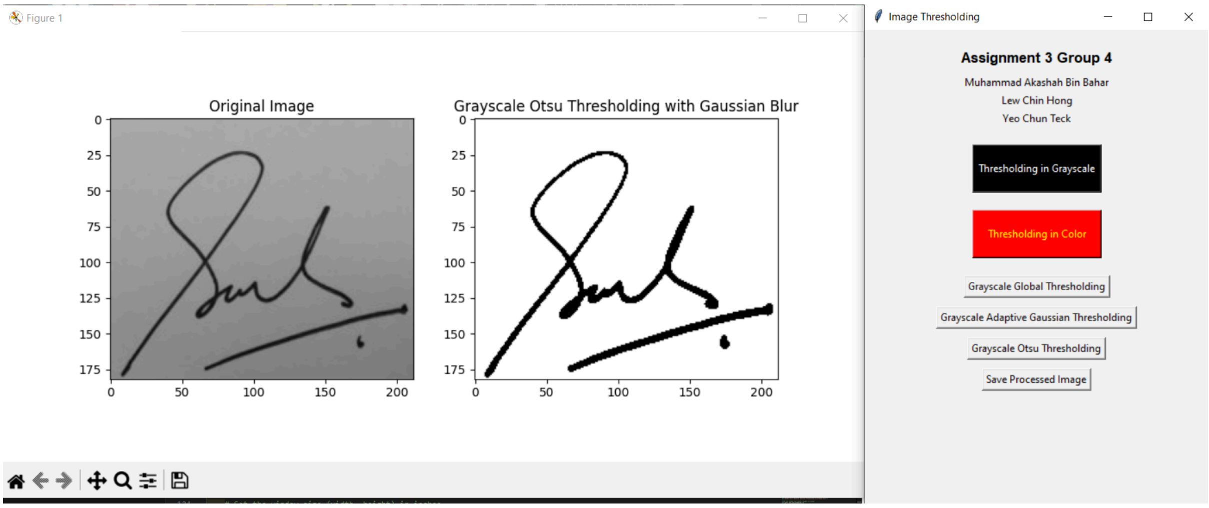
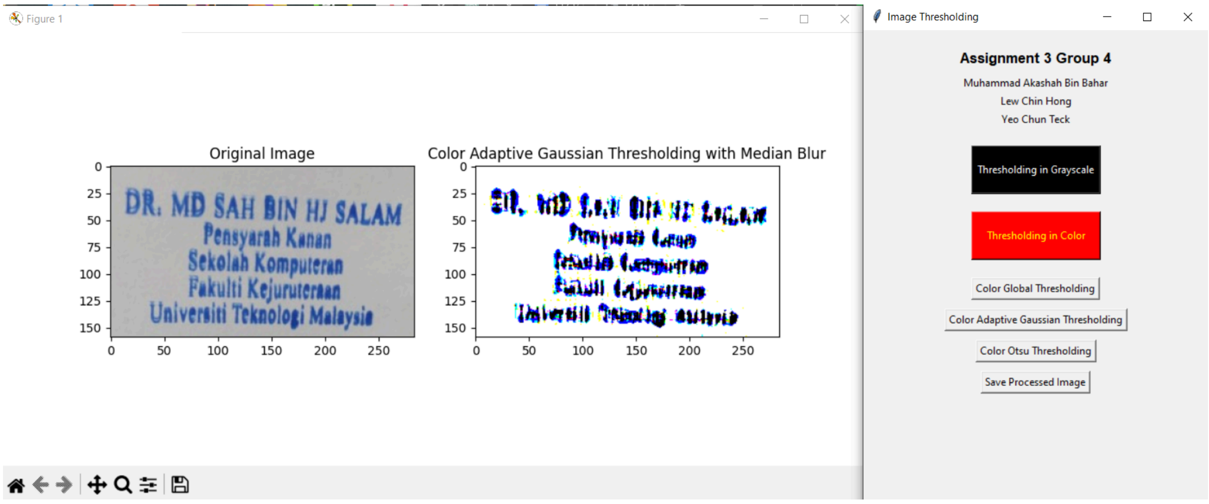


Figure 1
Original Image
Grayscale Otsu Thresholding with Gaussian Blur
0
25
50
75
100
125
150
175
0
50
100
150
200
Image Thresholding
Assignment 3 Group 4
Muhammad Akashah Bin Bahar
Lew Chin Hong
Yeo Chun Teck
Thresholding in Grayscale
Thresholding in Color
Grayscale Global Thresholding
Grayscale Adaptive Gaussian Thresholding
Grayscale Otsu Thresholding
Save Processed Image

6. Color Otsu Thresholding



Discussion

- Grayscale Otsu Thresholding

Gaussian blur is applied to the image to enhance the clarity of the background in the input image. In Grayscale Otsu Thresholding, the algorithm automatically computes the optimal threshold for the entire image by utilizing the maximum and minimum values of pixels in the entire image. Pixels with values less than the threshold are set to 0 (black), while those with values greater than the threshold are set to 255 (white). The effectiveness of the thresholding is not always guaranteed, as it directly considers the entire image for calculation, resulting in every pixel sharing the same threshold value. The larger the calculation scope, the lower the effectiveness in image thresholding, especially in varying light conditions where it may not perform optimally. Similar to Adaptive Gaussian

Thresholding, manual adjustment via a slider is not required because the algorithm automatically computes the threshold.

- Color Otsu Thresholding

  Color Otsu Thresholding is akin to Grayscale Otsu Thresholding. The distinction lies in the separation of color channels (Red, Green, Blue) from the input image. Otsu Thresholding is independently applied to each color channel of the input image, after which these channels are merged back into a color image.

Workload distribution among members

Member	**Tasks**				**Task percentage**
	System Planning	**Code Contribution**	**Report Writing**	**Video Making**	
Muhammad Akashah Bin Bahar	33%	33%	33%	33%	33
Lew Chin Hong	33%	33%	33%	33%	33
Yeo Chun Teck	34%	34%	34%	34%	34