Cong Cong

Z3414050

COMP9517

Assignment 1

# Stage1: Image filtering and thresholding

**1.1**

For global thresholding, I tried three different methods with **threshold value** and same maximum value 255:

* Binary thresholding
  + If the pixel value is larger than **threshold value**, it will be set to 255 otherwise it will set to 0.
* Truncate thresholding
  + If the pixel value is larger than **threshold value**, it will be changed to **threshold value** otherwise it will not be changed.
* Threshold to Zero
  + If the pixel value is larger than **threshold value**, it will be set to 0 otherwise it will not be changed.

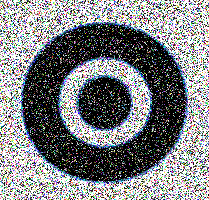
For Otsu thresholding, the thresholding value is not explicitly provided, it is calculated using Otsu algorithm.

For adaptive thresholding, the threshold value is decided for different regions of the image, as a result of adaptive thresholding, different regions of the same image can end up having different threshold value for different regions:

* Adaptive threshold:
  + Mean: threshold value is the mean of the selected region
  + Gaussian: threshold value is the weighted sum of the selected region

In our case, I have tried different value of the pixel neighbourhood size, and found out that size=5with subtracted constant equals to 2 gives the best results for adaptive threshold (mean/ Gaussian).

Original images and histograms:

1. A screenshot of a social media post

Description automatically generated2.A close up of a keyboard

Description automatically generated A screenshot of a social media post

Description automatically generated

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Global | | | | | Otsu | Adaptive |
| Threshold(60) | Threshold(127) | | Threshold(180) | |  |  |
| Binary | | Binary  A close up of a logo  Description automatically generated | | Binary | A close up of a womans face  Description automatically generated | Mean |
| Truncate | | Truncate | | Truncate | Gaussian |
| To Zero | | To Zero  A close up of a logo  Description automatically generated | | To Zero |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Global | | | Otsu | Adaptive |
| Threshold(60) | Threshold(127) | Threshold(180) |  |  |
| Binary  A close up of a logo  Description automatically generated | Binary  A screenshot of a cell phone  Description automatically generated | Binary  A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated | Mean  A close up of a logo  Description automatically generated |
| Truncate  A screenshot of a cell phone  Description automatically generated | Truncate | Truncate  A picture containing object  Description automatically generated | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A picture containing object  Description automatically generated | To Zero | To Zero  A picture containing monitor  Description automatically generated |  |

* [img1] In my opinion, global thresholding algorithms performs generally better than the other two. As the results produced by global thresholding show less noise and clearer separation between foreground and background. For Global thresholding, it can be found that when threshold value equals to 127, the results are relatively better. For example, for binary thresholding, small threshold values can’t remove noise well because most of the pixels will be set to intensity of 255 (brighter) and larger threshold values can’t show very clear separation between foreground and background because most of the pixels will be set to intensity of 0 (darker). Otsu thresholding shows a similar result as Global thresholding with threshold value equals to 127, as the histogram indicates, Otsu thresholding will find a value in the middle of range[0-255]. Moreover, I have run a code to check the threshold value chosen by Otsu thresholding algorithm, the result is 127. Furthermore, both approaches of Adaptive threshold do not perform very well, I think the main reason is Adaptive threshold try to choose a threshold value based on regions, the pixels in the black circle area can be affected by the outside grey or white colour background.
* [img2] In my opinion, global thresholding algorithms performs generally better than the other two. As the results produced by global thresholding show less noise and clearer separation between foreground and background for both the dice and its reflections. Whereas, the result produced by Otsu don’t show the reflections and the results produced by Adaptive thresholding have a few noise. For global thresholding, it can be observed that the best result is produced by Truncate global thresholding algorithm with thresholding value of 127. Otsu thresholding algorithm finds a “most suitable” threshold value of 135,which is closer to 127 and this is the reason why its result is similar to binary threshold algorithm with thresholding value of 127. For adaptive thresholding, it shows clear separation between foreground and background, but there are a few noise, I think the main reason is the participation of the dark background in the selected region which influence the mean or gaussian calculation.
* Overall, based on the shape of histograms of both images, it is reasonable to select a thresholding value in the middle of the intensity range in order to obtain a better performance. This can also explain why does Otsu thresholding algorithm behave similarly to global thresholding with threshold value equals to 127 and why does global thresholding with threshold value equals to 127 performs better than other thresholding algorithms for both images. However, global thresholding (truncate) works better on image 2, because it sets the pixel value to the threshold instead of setting to 256.

**1.2**

Based on the conclusion from part1.1, for the rest of part1.2 and part1.3, I have chosen threshold value of 127 for global thresholding algorithms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 3 | A screenshot of a cell phone  Description automatically generatedA close up of a logo  Description automatically generated | Binary  A close up of a logo  Description automatically generated | A close up of a mans face  Description automatically generated | Mean  A close up of a logo  Description automatically generated |
| Truncate | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A close up of a logo  Description automatically generated |  |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 5 | A screenshot of a cell phone  Description automatically generated  A close up of a logo  Description automatically generated | Binary  A close up of a mans face  Description automatically generated | A close up of a mans face  Description automatically generated | Mean |
| Truncate | Gaussian |
| To Zero |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 7 | A screenshot of a cell phone  Description automatically generated  A close up of a logo  Description automatically generated | Binary  A close up of a mans face  Description automatically generated |  | Mean |
| Truncate  A close up of a mans face  Description automatically generated | Gaussian |
| To Zero |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 11 | A screenshot of a cell phone  Description automatically generated  A close up of a mans face  Description automatically generated | Binary  A close up of a logo  Description automatically generated | A close up of a logo  Description automatically generated | Mean  A screenshot of a cell phone  Description automatically generated |
| Truncate  A close up of a mans face  Description automatically generated | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A close up of a logo  Description automatically generated |  |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 29 | A screenshot of a social media post  Description automatically generated  A close up of a logo  Description automatically generated | Binary  A close up of a logo  Description automatically generated | A close up of a logo  Description automatically generated | Mean  A close up of a logo  Description automatically generated |
| Truncate  A screenshot of a cell phone  Description automatically generated | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A close up of a logo  Description automatically generated |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 3 | A screenshot of a social media post  Description automatically generated  A close up of a keyboard  Description automatically generated | Binary  A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated | Mean  A close up of a logo  Description automatically generated |
| Truncate | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A close up of a black background  Description automatically generated |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 5 | A screenshot of a cell phone  Description automatically generated  A close up of a keyboard  Description automatically generated | Binary  A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated | Mean  A close up of a logo  Description automatically generated |
| Truncate | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A screen shot of a computer  Description automatically generated |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 9 | A screenshot of a cell phone  Description automatically generated  A screen shot of a smart phone  Description automatically generated | Binary  A picture containing screenshot  Description automatically generated | A screen shot of a smart phone  Description automatically generated | Mean  A close up of a logo  Description automatically generated |
| Truncate  A screen shot of a smart phone  Description automatically generated | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A picture containing monitor, indoor, wall, sitting  Description automatically generated |  |

For median filtering, I tried different values of kernel size, and it can be found that:

* For image1,
  + when kernel size equals to 3, global thresholding (binary) works better than others, this can be shown from the histogram shape, it is reasonable to select a threshold value in the middle (around 127). Moreover, the result of global thresholding (binary) has least amount of noise
  + As kernel size increases, it can be noticed that the amount of noise drops and Otsu thresholding performs similarly well or even better than global thresholding (binary), this can be shown from the histogram, selecting a thresholding value in the middle might not be the best choice, a value smaller than 127 might give a better result (Otsu algorithms chooses 117).
  + When kernel size increases up to 31, the original picture becomes excessively blurred and the results become out of shape.
  + Overall, with kernel size in the range of [5-27], Otsu and global thresholding both work well with the blurred image. However, results are likely to have noise with smaller with kernel size and results are likely to be out of shape with larger with kernel size, thus, it is reasonable to choose a kernel size around 11.
* For image2,
  + when kernel size equals to 3, global thresholding (truncate) works better. Other global thresholding algorithms and Otsu can’t show the image in the reflection well. Moreover, Adaptive thresholding algorithms show clear segmentation between the foreground and background but contain a few noise.
  + As kernel size increases, the result becomes worse, I think the reason behind is that unlike image1, where the foreground contains uniform colour, image2’s foreground contains multiple colours. After blurring the image, it becomes hard for thresholding algorithms to show clear segmentation between the black letters and their white background.
  + Overall, it can be seen that with kernel size set to 3, global thresholding (truncate) performs better.

**1.3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 3 | A close up of a logo  Description automatically generated | Binary  A close up of a logo  Description automatically generated | A close up of a logo  Description automatically generated | Mean |
| Truncate  A close up of a mans face  Description automatically generated | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A close up of a mans face  Description automatically generated |  |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 5 |  | Binary  A close up of a logo  Description automatically generated | A close up of a logo  Description automatically generated | Mean  A close up of a logo  Description automatically generated |
| Truncate  A close up of a mans face  Description automatically generated | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A close up of a mans face  Description automatically generated |  |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 51 | A screenshot of a cell phone  Description automatically generated | Binary  A close up of a logo  Description automatically generated | A close up of a logo  Description automatically generated | Mean |
| Truncate  A close up of a mans face  Description automatically generated | Gaussian  A screenshot of a cell phone  Description automatically generated |
| To Zero |  |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 3 | A screenshot of a social media post  Description automatically generated  A picture containing object, white  Description automatically generated | Binary  A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated | Mean  A close up of a logo  Description automatically generated |
| Truncate  A close up of a keyboard  Description automatically generated | Gaussian  A close up of a logo  Description automatically generated |
| To Zero |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel Size | Original image(blurred)+histogram | Global | Otsu | Adaptive |
| 5 | A screenshot of a social media post  Description automatically generated  A picture containing object  Description automatically generated | Binary  A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated | Mean  A close up of a logo  Description automatically generated |
| Truncate  A close up of a keyboard  Description automatically generated | Gaussian  A close up of a logo  Description automatically generated |
| To Zero  A screen shot of a social media post  Description automatically generated |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Gaussian kernel standard deviation = 3 | Gaussian kernel standard deviation = 5 | Gaussian kernel standard deviation = 7 | Gaussian kernel standard deviation = 9 |
| A close up of a keyboard  Description automatically generated | A close up of a clock  Description automatically generated | A close up of a clock  Description automatically generated | A close up of a clock  Description automatically generated |

For gaussian filtering, I tried different values of kernel sizes and standard deviations, it can be found that:

* For image1,
  + From the shape of the histogram, it can be found that for global thresholding algorithms, a threshold value should be chosen around 127. In this case, I have chosen 127 as my threshold value, and the output of global thresholding(binary) works well. For Otsu thresholding, the algorithm has calculated 113 as the most suitable threshold value. Both Otsu thresholding and global thresholding(binary) perform better than other thresholding algorithms with different kernel sizes.
  + I have tried various values of kernel sizes (range from 3 to 51) and it can be found that the histogram shape is not influenced much with the increase of kernel size. Moreover, the result of global thresholding(binary) and Otsu thresholding performs similarly well in all different kernel sizes. Whereas, Adaptive thresholding algorithm’s outputs become blurry and they don’t show very clear and smooth edges with large kernel size.
  + Overall, global thresholding(binary) and Otsu thresholding both work well. When kernel size = 5, all three thresholding algorithms perform generally well over other tested kernel sizes.
* For image2,
  + It can be found that when kernel size = 3, global thresholding(truncate) performs better than others. With larger kernel sizes, the original image becomes excessively blurry and the outputs after thresholding don’t work well.
* For gaussian filtering, another important factor is the standard deviation, I tried different standard deviations and found that increasing standard deviation reduces noise but also attenuates high frequency details(edges) and make image blurry. Thus, in this case, I chose standard deviation = 3.

# Stage 2: Use filtering and thresholding for segmentation

A screenshot of a cell phone

Description automatically generated

For this part, firstly, from plotted the histogram, it can be found that most of the grey and black rounded dots are smaller than a value in between 150 and 200. Thus, it is reasonable to set the thresholding value as some number in this range.

|  |  |  |  |
| --- | --- | --- | --- |
| Kernel size=3 | Global | Otsu | Adaptive |
| Median filter    A close up of a logo  Description automatically generated | Binary  A close up of a logo  Description automatically generated | A close up of a logo  Description automatically generated | Mean  A close up of a piece of paper  Description automatically generated |
| Truncate | Gaussian  A close up of a piece of paper  Description automatically generated |
| To Zero  A screenshot of a cell phone  Description automatically generated |  |
| Gaussian filter    A close up of a logo  Description automatically generated | Binary  A close up of a logo  Description automatically generated | A close up of a logo  Description automatically generated | Mean  A close up of a piece of paper  Description automatically generated |
| Truncate | Gaussian  A close up of a piece of paper  Description automatically generated |
| To Zero  A screenshot of a cell phone  Description automatically generated |  |

1. **Kernel size = 3**
   * Median filtering + thresholding
     + Kernel size = 3, thresholding = 150 Kernel size = 3, thresholding = 160

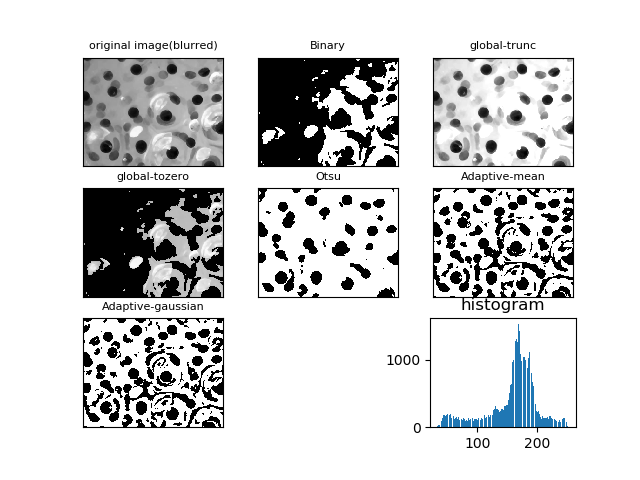
A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

Kernel size = 3, thresholding = 170 Kernel size = 3, thresholding = 180

A screenshot of a cell phone

Description automatically generated 

If median filtering with kernel size equals = 3 is used, output of global thresholding algorithm(truncate) with thresholding value = 170 has clearer separation separates more cells from the background comparing with other algorithms.

* + Gaussian filtering + thresholding
    - Kernel size = 3, thresholding = 150 Kernel size = 3, thresholding = 160

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Kernel size = 3, thresholding = 170 Kernel size = 3, thresholding = 180

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

If gaussian filtering with kernel size equals = 3 is used, output of global thresholding algorithm(truncate) with thresholding value = 170 s has clearer separation separates more cells from the background comparing with other algorithms.

Overall, for both filtering method with kernel size =3, output of global thresholding algorithm(truncate) with thresholding value = 170 s has clearer separation separates more cells from the background comparing with other algorithms, this algorithm performs equally well after applying different filtering approach on this image.

1. **Kernel size = 5**
   * + - Median filtering + thresholding

Kernel size = 5 thresholding = 150 Kernel size = 5, thresholding = 160

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Kernel size = 5, thresholding = 170 Kernel size = 5, thresholding = 180

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

If median filtering with kernel size equals = 5 is used, output of global thresholding algorithm(truncate) with thresholding value = 170 has clearer separation and separates more cells from the background and comparing with other algorithms.

* + - * Gaussian filtering + thresholding

Kernel size = 5 thresholding = 150 Kernel size = 5, thresholding = 160

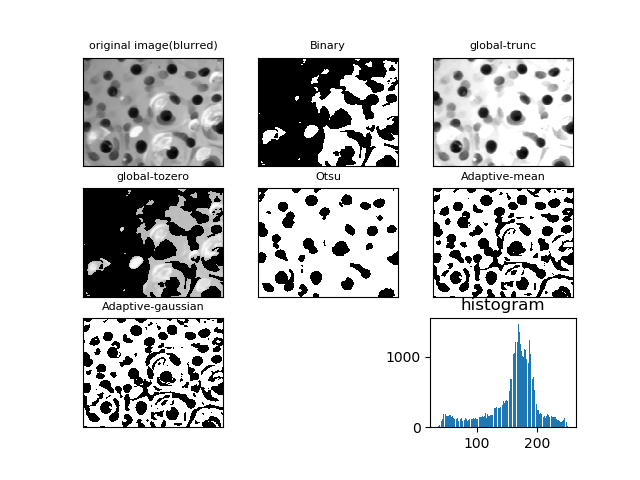
A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Kernel size = 5 thresholding = 170 Kernel size = 5, thresholding = 180

A screenshot of a cell phone

Description automatically generated

If gaussian filtering with kernel size equals = 5 is used, output of global thresholding algorithm(truncate) with thresholding value = 170 s has clearer separation separates more cells from the background comparing with other algorithms.

Best of Median filtering with kernel size = 5 Best of Gaussian filtering with kernel size = 5

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Overall, for both filtering method with kernel size =5, output of global thresholding algorithm(truncate) with thresholding value = 170 s has clearer separation separates more cells from the background comparing with other algorithms. However, it can also be noticed that global thresholding algorithm(truncate) produces less noise with median filtering.

1. **Kernel size = 7**
   * + - Median filtering + thresholding

Kernel size = 7 thresholding = 150 Kernel size = 7, thresholding = 160

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Kernel size = 7 thresholding = 170 Kernel size = 7, thresholding = 180

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

If median filtering with kernel size equals = 7 is used, output of global thresholding algorithm(truncate) with thresholding value = 170 separates more cells from the background comparing with other algorithms.

* + - * Gaussian filtering + thresholding

Kernel size = 7 thresholding = 150 Kernel size = 7, thresholding = 160

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Kernel size = 7 thresholding = 170 Kernel size = 7, thresholding = 180

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

If gaussian filtering with kernel size equals = 7 is used, output of global thresholding algorithm(truncate) with thresholding value = 170 separates more cells from the background comparing with other algorithms.

Best of Median filtering with kernel size = 7 Best of Gaussian filtering with kernel size = 7

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Overall, for both filtering method with kernel size =7, output of global thresholding algorithm(truncate) with thresholding value = 170 s has clearer separation separates more cells from the background comparing with other algorithms. However, it can also be noticed that global thresholding algorithm(truncate) produces less noise with median filtering.

Best of kernel size = 3 Best of kernel size = 5

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Best of kernel size = 7

A screenshot of a cell phone

Description automatically generated

Here, I have chosen the best performances of different thresholding Algorithms with different kernel sizes. It can be notices that when kernel size equals to 3, the outputs still contains some noise. Whereas, when kernel size grows to 7, the number of noise is reduced but the overall quality of the image is too blurry. Thus, by setting the kernel size equals to 5 and set threshold value equals to 170, it is expected to obtained a relatively better segmentation by global thresholding algorithm(Truncate).

Segmentation process:

1. Take input of kernel size.
2. Run median or gaussian filtering based on input kernel size.
3. Run different thresholding algorithms on blurred images
4. Compare results, choose the most suitable one based on the amount of noise, clearness of edges

In conclusion, for image3, from the histogram it can be found that by setting a threshold value in the range of [150~200] is more likely to give a better segmentation result. Because intensity value larger than 200 are likely to be the bright reflection(noise). Otsu algorithm tends to find a threshold value smaller than 150 and Adaptive thresholding is dependent on the choice of region blocks. I think the reason why median filtering works better than gaussian filtering is that median filtering removes noise while keeping edges sharp, whereas gaussian filtering removes noise and reduce contrast.