HW7 Report

Computer Vision
Section 02
21800181 Kim Jisu

HW 7

- Purpose

Image segmentation using Threshold. the letters or fingerprints of the image are black and the background is white.

- Principle

double cv::threshold(InputArray src, OutputArray dst, double thresh, double maxval, int type)

Applies a fixed-level threshold to each array element.

void cv::adaptiveThreshold(InputArray src, OutputArray dst, double maxValue, int adaptiveMethod, int thresholdType, int blockSize, double C)

Applies an adaptive threshold to an array.

- Process

fingerprint image:

The threshold function was used to make the fingerprint part black and the background part white. and use same threshold for each pixel.

128 (gray) was given as threshold, 255(white) is max value. fingerprint part 0 (black) and the background part 255 (white).

adaptive image:

If the lighting is not constant in the original image or has multiple background colors, it is difficult to create a clear binary image with one threshold. Therefore, after dividing the image into several areas, adaptiveThreshold was used to obtain and apply a threshold value using only the surrounding pixel value. Therefore the adaptiveThreshold function was used to set character region to black and background region to white.

box size = 7

C = 10

 The current value is the best value ever tried.

character region to 0 and background region to 255.

adaptive_1:

If the lighting is not constant in the original image or has multiple background colors, it is difficult to create a clear binary image with one threshold. Therefore, after dividing the image into several areas, adaptiveThreshold was used to obtain and apply a threshold value using only the surrounding pixel value. Therefore the adaptiveThreshold function was used to set character region to black and background region to white.

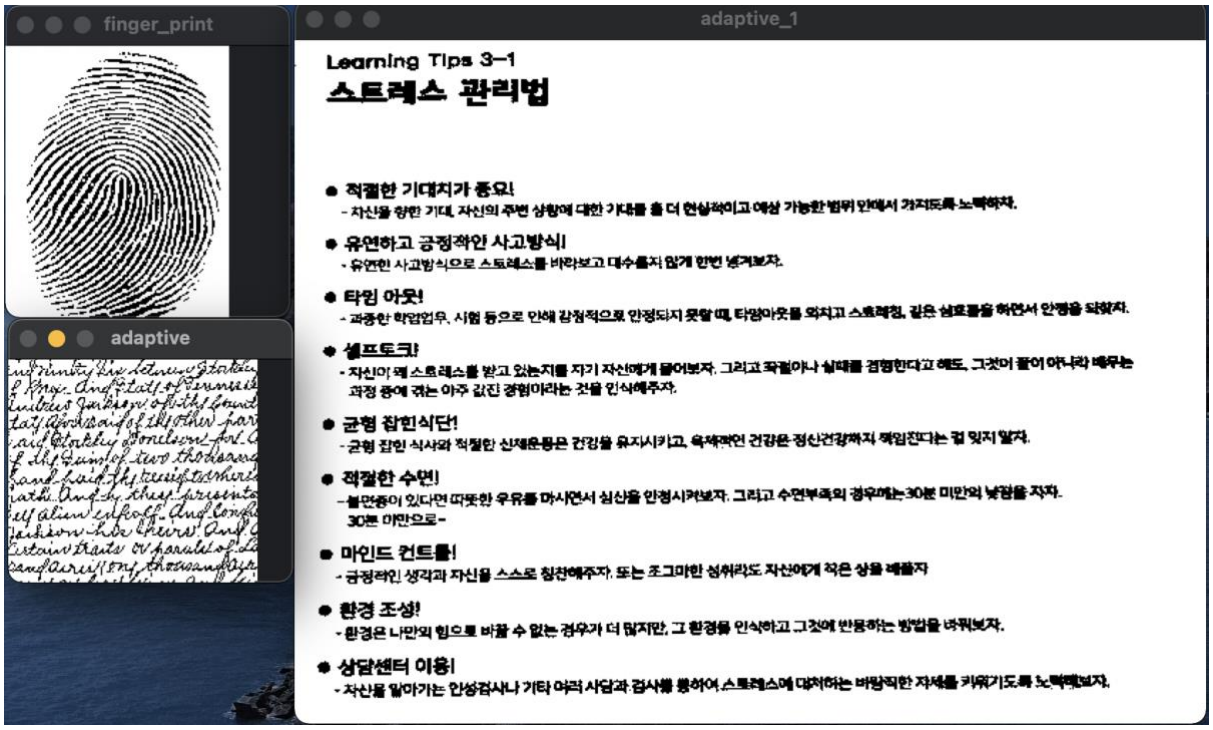
box size = 115

C = 12

□ The current value is the best value ever tried.

character region to 0 and background region to 255.

- Result



- Code

```
#include "opencv2/opencv.hpp"
#include <iostream>

using namespace std;
using namespace cv;
```

```
int main(){
	Mat finger_print,adaptive_1,adaptive;
	finger_print = imread("finger_print.png",0);
	adaptive_1 = imread("adaptive_1.jpg",0);
	adaptive = imread("adaptive.png",0);

	threshold(finger_print, finger_print, 128, 255, THRESH_BINARY);
	adaptiveThreshold(adaptive, adaptive, 255, ADAPTIVE_THRESH_MEAN_C, THRESH_BINARY, 7, 10);
	adaptiveThreshold(adaptive_1, adaptive_1, 255, ADAPTIVE_THRESH_MEAN_C, THRESH_BINARY, 115,
12);

	imshow("finger_print",finger_print);
	imshow("adaptive",adaptive);
	imshow("adaptive_1",adaptive_1);
	waitKey();
}
```

l References

https://docs.opencv.org/master/