

HW5 report

姓名:徐有慶 學號:R05922162

1. 作業要求:

實作灰階圖片的 Dilation, Erosion, Opening, Closing, and Hit-and-Miss。

程式語言: Java

執行環境: Eclipse

2. 程式設計:

(a) Dilaton:

Step1. 建立一 3-5-5-5-3, value 為 0 的 kernel

Step2. 根據公式 $(f \oplus b)(s,t) = \max\{f(s-x, t-y) + b(x,y) \mid (s-x, t-y) \in A, (x,y) \in B\}$

做 dilation, 其中, A 為原圖像素所成的集合, B 為 kernel 像素所成的集合, #A =

512*512, #B = 3+5+5+5+3, f(x,y) 即為原圖 A 在 (x,y) 的 graylevel, b(x,y) 即為 kernel B

在 (x,y) 的 graylevel。

(b) Erosion:

Step1. 建立一 3-5-5-5-3, value 為 0 的 kernel

Step2. 根據公式 $(f \ominus b)(s,t) = \min\{f(s+x, t+y) - b(x,y) \mid (s+x, t+y) \in A, (x,y) \in B\}$

做 erosion, 其中, A 為原圖像素所成的集合, B 為 kernel 像素所成的集合, #A =

512*512, #B = 3+5+5+5+3, f(x,y) 即為原圖 A 在 (x,y) 的 graylevel, b(x,y) 即為 kernel B

在 (x,y) 的 graylevel。

(c) Opening:

Step1. 將原圖做完 erosion 後再做 dilation。

(d)Closing:

Step1. 將原圖做完 dilation 後再做 erosion。

3. 主要程式：

(a)Dilation:

```
public static int[][] dilation(int[][] grayImg, Set<Point> bSet) {  
  
    int[][] dilationImg = new int[HEIGHT][WIDTH];  
  
    for (int i = 0; i < HEIGHT; i++) {  
  
        for (int j = 0; j < WIDTH; j++) {  
  
            int max = 0;  
  
            for (Point point : bSet) {  
  
                int x = point.getX();  
  
                int y = point.getY();  
  
                int value = point.getValue();  
  
                if (i - x > 0 && j - y > 0 && i - x < HEIGHT && j - y < WIDTH) {  
  
                    int grayValue = grayImg[i - x][j - y] + value;  
  
                    if (grayValue > max) {  
  
                        max = grayValue;  
  
                    }  
  
                }  
  
            }  
  
            dilationImg[i][j] = max << 16 | max << 8 | max;  

```

```

    }

}

return dilationImg;

}

```

(b) Erosion:

```

public static int[][] erosion(int[][] grayImg, Set<Point> bSet) {

    int[][] erosImg = new int[HEIGHT][WIDTH];

    for (int i = 0; i < HEIGHT; i++) {

        for (int j = 0; j < WIDTH; j++) {

            int min = 255;

            for (Point point : bSet) {

                int x = point.getX();

                int y = point.getY();

                int value = point.getValue();

                if (i + x > 0 && j + y > 0 && i + x < HEIGHT && j + y < WIDTH) {

                    int grayValue = grayImg[i + x][j + y] - value;

                    if (grayValue < min) {

                        min = grayValue;

                    }

                }

            }

        }

    }

}

```

```

        erosImg[i][j] = min << 16 | min << 8 | min;

    }

}

return erosImg;

}

```

(c) Opening:

```

public static int[][] opening(int[][] grayImg, Set<Point> bSet) {

    int[][] erosionImgArray = erosion(grayImg, bSet);

    return dilation(convertToGrayImg(erosionImgArray), bSet);

}

```

(d) Closing:

```

public static int[][] closing(int[][] grayImg, Set<Point> bSet) {

    int[][] dilationImgArray = dilation(grayImg, bSet);

    return erosion(convertToGrayImg(dilationImgArray), bSet);

}

```

4. 執行結果：

Dilation



Erosion



Opening



Closing



5. 如何執行

執行./source code/ CV-HW5/HW5.jar 即可產生作業要求的兩種圖片(lena.bmp 需要和 HW5.jar 在同一個資料夾底下)