

## 2021/11/04 陳嘉政 r10922171 HW5

### 1. grayscale dilation

#### (1) 主要程式碼

參數為 lena 原圖以及在 main 裡宣告好的 kernel。

在此 function 裡利用兩層迴圈跑所有的點，由每個點(i,j)為原點利用 rect 畫出 kernel 大小的框框設給 temp，如果劃出的框框會超出邊界，則用最大的 column 或 row 來減掉 j 或 i 得到新的長或寬。得到 temp 後，將 temp 丟到 findMax function 裡來找出範圍內的最大灰階值，且設給(i,j)當成新的灰階值。

```
Mat grayscaleDilation(Mat input, Mat kernel)
{
    Mat result;
    result = Mat::zeros(input.rows, input.cols, CV_8UC1);
    int kernelSize = kernel.rows;
    for (int i = 0; i < input.rows; i++) {
        for (int j = 0; j < input.cols; j++) {
            Mat temp;
            if ((i + kernelSize) < input.rows && (j + kernelSize) < input.cols) {
                temp = input(cv::Rect(j, i, kernelSize, kernelSize));
            }
            else if ((i + kernelSize) >= input.rows && (j + kernelSize) < input.cols) {
                temp = input(cv::Rect(j, i, kernelSize, input.rows - i));
            }
            else if ((i + kernelSize) < input.rows && (j + kernelSize) >= input.cols) {
                temp = input(cv::Rect(j, i, input.cols - j, kernelSize));
            }
            else {
                temp = input(cv::Rect(j, i, input.cols - j, input.rows - i));
            }
            int max = findMax(temp);
            result.at<uint8_t>(i, j) = max;
        }
    }
    return result;
}
```

```
int findMax(Mat input)
{
    int max = 0;
    for (int i = 0; i < input.rows; i++) {
        for (int j = 0; j < input.cols; j++) {
            if (input.at<uint8_t>(i, j) > max)
                max = input.at<uint8_t>(i, j);
        }
    }
    return max;
}
```

#### (2) 結果



## 2. grayscale erosion

### (1) 主要程式碼

參數為 lena 原圖以及在 main 裡宣告好的 kernel。

方法與 dilation 幾乎相同，差別在於得到的 temp 要丟到 findMin 裡取出最小灰階值，然後設給(i,j)當成新的灰階值

```
Mat grayscaleErosion(Mat input, Mat kernel)
{
    Mat result;
    result = Mat::zeros(input.rows, input.cols, CV_8UC1);
    int kernelSize = kernel.rows;
    for (int i = 0; i < input.rows; i++) {
        for (int j = 0; j < input.cols; j++) {
            Mat temp;
            if ((i + kernelSize) < input.rows && (j + kernelSize) < input.cols) {
                temp = input(cv::Rect(j, i, kernelSize, kernelSize));
            }
            else if ((i + kernelSize) >= input.rows && (j + kernelSize) < input.cols) {
                temp = input(cv::Rect(j, i, kernelSize, input.rows - i));
            }
            else if ((i + kernelSize) < input.rows && (j + kernelSize) >= input.cols) {
                temp = input(cv::Rect(j, i, input.cols - j, kernelSize));
            }
            else {
                temp = input(cv::Rect(j, i, input.cols - j, input.rows - i));
            }
            int min = findMin(temp);
            result.at<uint8_t>(i, j) = min;
        }
    }
    return result;
}
```

```
int findMin(Mat input)
{
    int min = 255;
    for (int i = 0; i < input.rows; i++) {
        for (int j = 0; j < input.cols; j++) {
            if (input.at<uint8_t>(i, j) < min)
                min = input.at<uint8_t>(i, j);
        }
    }
    return min;
}
```

(2) 結果



3. grayscale opening

(1) 主要程式碼

將第二題 erosion 後得出的結果，丟到第一題的 grayscaleDilation function 裡頭，即可得出 opening 後的結果

```
/*grayscale_opening*/  
Mat opening_image = grayscaleDilation(erosion_image, kernel);  
imshow("opening", opening_image);  
waitKey(0);
```

(2) 結果



4. grayscale closing

(1) 主要程式碼

將第一題 dilation 後得出的結果，丟到第二題的 grayscaleErosion function 裡頭，即可得出 closing 後的結果

```
/*grayscale_closing*/  
Mat closing_image = grayscaleErosion(dilation_image, kernel);  
imshow("closing", closing_image);  
waitKey(0);
```

(2) 結果

