2021/11/20 陳嘉政 r10922171 HW6

程式碼:

1. 先將 **lena** 做二值化，方法和之前相同
2. 二值化過後的 **lena** 做 **downsampling: input** 為二值化影像，另外宣告一個 **64x64** 矩陣，每個**(row,col)**的值為對應 **input (row,col)**為 **8** 的倍數那些點的值。

```
downsampling(Mat input)

Mat resized_image = Mat::zeros(64, 64, CV_8UC1);
for (int i = 0; i < input.rows; i++) {
    for (int j = 0; j < input.cols; j++) {
        if (i % 8 == 0 && j % 8 == 0) {
            resized_image.at<uint8_t>(i / 8, j / 8) = input.at<uint8_t>(i, j);
        }
    }
}
return resized_image;
```

3. 宣告一個 **result** 矩陣並全清成 **0**，然後將 **downsampled** 後的影像每個點做，若值為 **255**，就由此 **pixel** 劃出設好地矩形範圍，進入 **Yokoi function**

```
/*Yokoi connected number*/
Mat result = Mat::zeros(64, 64, CV_8UC1);
for (int i = 0; i < downsampled_image.rows; i++) {
    for (int j = 0; j < downsampled_image.cols; j++) {
        if (downsampled_image.at<uint8_t>(i, j) == 255) {
            int startRow = i > 0 ? 1 : 0;
            int startCol = j > 0 ? 1 : 0;
            int sizeRow = startRow == 1 ? 3 : 2;
            int sizeCol = startCol == 1 ? 3 : 2;
            sizeRow = i == 63 ? 2 : sizeRow;
            sizeCol = j == 63 ? 2 : sizeCol;
            Mat temp = downsampled_image(cv::Rect(j - startCol, i - startRow, sizeCol, sizeRow));
            result.at<uint8_t>(i, j) = Yokoi(temp, i, j);
        }
    }
}
```

4.將每個情況列舉出來，每個情況 **pixel** 都去算他的四個 **corner** 找出 **r,q,s**。如果四個都是 **r** 回傳 **5**，否則計算 **q** 數目並回傳，將回傳數字記錄在 **result** 的對應 **pixel** 中

```
Yokoi(Mat input,int row, int col)

int di[] = {0, -1, 0, 1, 1, -1, -1, 1};
int dj[] = {1, 0, -1, 0, 1, 1, -1, -1};
char a[4] = { 0 };
int n;
```

(1)左上角

```
if (row == 0 && col == 0) {
    int i = 0, j = 0;
    for (n = 2; n < 3; n++) {
        if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 1], j + dj[n + 1])) {
            if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 2], j + dj[n + 2]) && input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[(n + 6) % 8], j + dj[(n + 6) % 8]))
                a[n + 1] = 'r';
            else
                a[n + 1] = 'q';
        }
        else
            a[n + 1] = 's';
    }
    if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[0], j + dj[0]))
        a[0] = 'q';
    int count = 0;
    for (n = 0; n < 4; n++)
        if (a[n] == 'q')
            count++;
    return count;
}
```

(2)右上角

```
else if (row == 0 && col == 63) {
    int i = 0, j = 1;
    for (n = 1; n < 2; n++) {
        if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 1], j + dj[n + 1])) {
            if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 2], j + dj[n + 2]) && input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[(n + 6) % 8], j + dj[(n + 6) % 8]))
                a[n + 1] = 'r';
            else
                a[n + 1] = 'q';
        }
        else
            a[n + 1] = 's';
    }
    if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[3], j + dj[3]))
        a[3] = 'q';
    int count = 0;
    for (n = 0; n < 4; n++)
        if (a[n] == 'q')
            count++;
    return count;
}
```

(3)左下角

```
else if (row == 63 && col == 0) {
    int i = 1, j = 0;
    for (n = -1; n < 0; n++) {
        if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 1], j + dj[n + 1])) {
            if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 2], j + dj[n + 2]) && input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[(n + 6) % 8], j + dj[(n + 6) % 8]))
                a[n + 1] = 'r';
            else
                a[n + 1] = 'q';
        }
        else
            a[n + 1] = 's';
    }
    if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[1], j + dj[1]))
        a[1] = 'q';
    int count = 0;
    for (n = 0; n < 4; n++)
        if (a[n] == 'q')
            count++;
    return count;
}
```

(4)右下角

```
else if(row == 63 && col == 63) {
    int i = 1, j = 1;
    for (n = 0; n < 1; n++) {
        if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 1], j + dj[n + 1])) {
            if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 2], j + dj[n + 2]) && input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[(n + 6) % 8], j + dj[(n + 6) % 8]))
                a[n + 1] = 'r';
            else
                a[n + 1] = 'q';
        }
        else
            a[n + 1] = 's';
    }
    if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[2], j + dj[2]))
        a[2] = 'q';
    int count = 0;
    for (n = 0; n < 4; n++)
        if (a[n] == 'q')
            count++;
    return count;
}
```

(5)上方

```
else if (row == 0) {
    int i = 0, j = 1;
    for (n = 1; n < 3; n++) {
        if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 1], j + dj[n + 1])) {
            if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 2], j + dj[n + 2]) && input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[(n + 6) % 8], j + dj[(n + 6) % 8]))
                a[n + 1] = 'r';
            else
                a[n + 1] = 'q';
        }
        else
            a[n + 1] = 's';
    }
    if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[0], j + dj[0]))
        a[0] = 'q';
    int count = 0;
    for (n = 0; n < 4; n++)
        if (a[n] == 'q')
            count++;
    return count;
}
```

(6)下方

```
else if (row == 63) {
    int i = 1, j = 1;
    for (n = -1; n < 1; n++) {
        if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 1], j + dj[n + 1])) {
            if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 2], j + dj[n + 2]) && input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[(n + 6) % 8], j + dj[(n + 6) % 8]))
                a[n + 1] = 'r';
            else
                a[n + 1] = 'q';
        }
        else
            a[n + 1] = 's';
    }
    if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[2], j + dj[2]))
        a[2] = 'q';
    int count = 0;
    for (n = 0; n < 4; n++)
        if (a[n] == 'q')
            count++;
    return count;
}
```

(7)左方

```
else if (col == 0) {
    int i = 1, j = 0;
    for (n = -1; n < 3; n+=3) {
        if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 1], j + dj[n + 1])) {
            if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 2], j + dj[n + 2]) && input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[(n + 6) % 8], j + dj[(n + 6) % 8]))
                a[n + 1] = 'r';
            else
                a[n + 1] = 'q';
        }
        else
            a[n + 1] = 's';
    }
    if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[1], j + dj[1]))
        a[1] = 'q';
    int count = 0;
    for (n = 0; n < 4; n++)
        if (a[n] == 'q')
            count++;
    return count;
}
```

(8)右方

```
else if (col == 63) {
    int i = 1, j = 1;
    for (n = 0; n < 2; n++) {
        if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 1], j + dj[n + 1])) {
            if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 2], j + dj[n + 2]) && input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[(n + 6) % 8], j + dj[(n + 6) % 8]))
                a[n + 1] = 'r';
            else
                a[n + 1] = 'q';
        }
        else
            a[n + 1] = 's';
    }
    if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[3], j + dj[3]))
        a[3] = 'q';
    int count = 0;
    for (n = 0; n < 4; n++)
        if (a[n] == 'q')
            count++;
    return count;
}
```

(9)正常情況

```
else {
    int i = 1, j = 1;
    for (n = -1; n < 3; n++) {
        if (input.at<uint8_t>(i, j) == input.at<uint8_t>(i + di[n + 1], j + dj[n + 1])) {
            if (input.at<uint8_t>(i, j) != input.at<uint8_t>(i + di[n + 2], j + dj[n + 2]) || input.at<uint8_t>(i, j) != input.at<uint8_t>(i + di[(n + 6) % 8], j + dj[(n + 6) % 8]))
                a[n + 1] = 'q';
            else
                a[n + 1] = 'r';
        }
        else
            a[n + 1] = 's';
    }
    if (a[0] == 'r' && a[1] == 'r' && a[2] == 'r' && a[3] == 'r')
        return 5;
    else {
        int count = 0;
        for (n = 0; n < 4; n++)
            if (a[n] == 'q')
                count++;
        return count;
    }
}
```

5.利用 txt 檔輸出矩陣，並截圖調整長寬

```
fstream file("matrix.txt", ios::out);
for (int i = 0; i < result.rows; i++) {
    for (int j = 0; j < result.cols; j++) {
        if (result.at<uint8_t>(i, j) != 0)
            file << unsigned(result.at<uint8_t>(i, j));
        else
            file << " ";
    }
    file << endl;
}
file.close();
```

結果: