Computer vision
Homework6

B06902091 資工四 羅寶瑩

Yokoi Connectivity Number

Description

Write a program which counts the Yokoi connectivity number on a downsampled image(lena.bmp).

(1) Binarize the benchmark image lena

Use the binarizing function in hw2 to obtain a binarized image of Lena.

```
void binarize(Mat img) {
    for (int i = 0; i < img_rows; i++) {
        for (int j = 0; j < img_cols; j++) {
            if (img.at<uchar>(i, j) < 128) {//0~127
                img.at<uchar>(i, j) = 0;
            }
            else {
                img.at<uchar>(i, j) = 255;
            }
        }
    }
}
```

(2) Downsampling Lena from 512x512 to 64x64

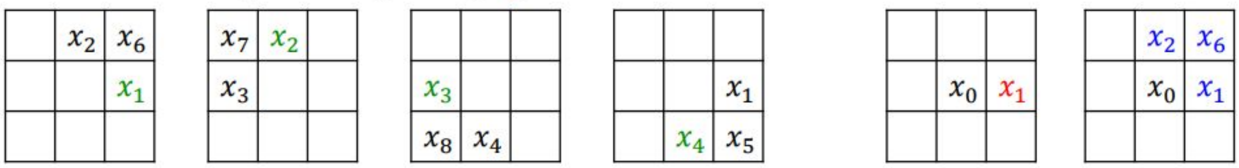
Create a matrix in size 64x64. For each pixel in the new matrix take the topmost-left pixel as the downsampled data.

```
void downsample(Mat ori, Mat down) {
    for (int i = 0; i < 64; i++) {
        for (int j = 0; j < 64; j++) {
            down.at<uchar>(i, j) = ori.at<uchar>(i * 8, j * 8);
        }
    }
}
```

(3) Count the Yokoi connectivity number on a downsampled lena using 4-connected

i. Obtain the value of x0~x8.

Corner Neighborhood
(for corresponding x_i)



ii. Calculate the value of a1~a4 using the function mentioned in the lecture.

$$h(b,c,d,e) = \begin{cases} q & if\ b = c\ and\ (d \neq b \vee e \neq b) \\ r & if\ b = c\ and\ (d = b \wedge e = b) \\ s & if\ b \neq c \end{cases}$$

```
char h(int b, int c, int d, int e) {
    if (b == c) {
        if (d != b || e != b) {
            return 'q';
        }
        else if (d == b && e == b) {
            return 'r';
        }
    }
    else {
        return 's';
    }
}
```

iii. Calculate the connectivity number of the current pixel the function mentioned in the lecture and store it in the output matrix

$$f(a_1, a_2, a_3, a_4) = \begin{cases} 5\ \ if\ a_1 = a_2 = a_3 = a_4 = r \\ n\ \ where\ n = numberof\{a_k | a_k = q\}, otherwise \end{cases}$$

```
int f(int a1, int a2, int a3, int a4) {
    if (a1 == 'r' && a2 == 'r' && a3 == 'r' && a4 == 'r') {
        return 5;
    }
    else {
        int count = 0;
        if (a1 == 'q') {
            count++;
        }
        if (a2 == 'q') {
            count++;
        }
        if (a3 == 'q') {
            count++;
        }
        if (a4 == 'q') {
            count++;
        }
        return count;
    }
}
```

Source code

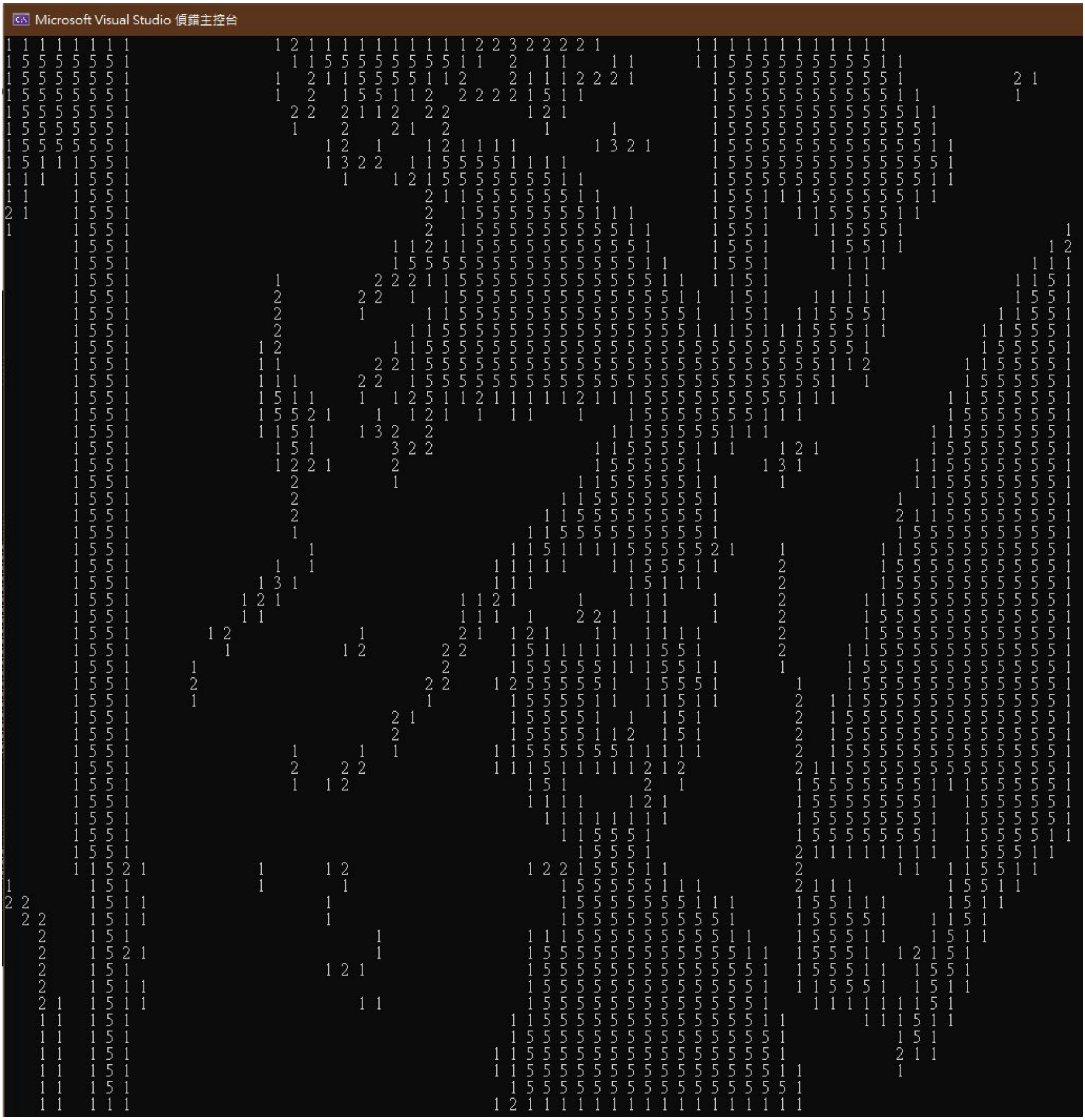
```
void yokoi(Mat down) {
    for (int i = 0; i < 64; i++) {
        for (int j = 0; j < 64; j++) {
            if (down.at<uchar>(i, j) == 0) {
                matrix[i][j] = 0;
                continue;
            }

            //i. Obtain the value of x0~x8.
            int x[9] = { 0,0,0,0,0,0,0,0,0 };
            x[0] = down.at<uchar>(i, j);
            if (j != 63) {
                x[1] = down.at<uchar>(i, j + 1);
            }
            if (j != 0) {
                x[3] = down.at<uchar>(i, j - 1);
            }
            if (i != 0) {
                x[2] = down.at<uchar>(i - 1, j);
            }
            if (i != 63) {
                x[4] = down.at<uchar>(i + 1, j);
            }
            if (i != 63 && j != 63) {
                x[5] = down.at<uchar>(i + 1, j + 1);
            }
            if (i != 0 && j != 0) {
                x[7] = down.at<uchar>(i - 1, j - 1);
            }
            if (i != 0 && j != 63) {
                x[6] = down.at<uchar>(i - 1, j + 1);
            }
            if (i != 63 && j != 0) {
                x[8] = down.at<uchar>(i + 1, j - 1);
            }

            //ii. Calculate the value of a1~a4
            char a[4];
            a[0] = h(x[0], x[1], x[6], x[2]);
            a[1] = h(x[0], x[2], x[7], x[3]);
            a[2] = h(x[0], x[3], x[8], x[4]);
            a[3] = h(x[0], x[4], x[5], x[1]);

            //iii. Calculate the connectivity number
            matrix[i][j] = f(a[0], a[1], a[2], a[3]);;
        }
    }
}
```

Result



Reference:

1. *Lecture slide (CV1_CH6_2020) - p.61~66*