

# Computer Vision hw\_6

By R01922124 許彥彬

Predefine Three parameter Q, R, and S as 1, 0 and 2.

This work includes four new functions and some functions that defined in previous homework.

1. A resize function to resize the lena.bmp. (The default setting in my code is 64x64)

codes:

```
void resize(Mat *p, int rows, int cols){
    Mat res(rows, cols, p->type());
    int row_scale = p->rows/rows;
    int col_scale = p->cols/cols;
    for(int i=0;i<rows;i++){
        for(int j=0;j<cols;j++){
            pixel_set(&res,i,j,pixel_get(p,i*row_scale,j*col_scale));
        }
    }
    *p=res.clone();
}
```

2. Formula 1:

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

Codes:

```
int H4(int b,int c, int d, int e){
    if(b==c && ( d!=b || e!=b )){
        return Q;
    }
    if(b==c && d==b && e==b){
        return R;
    }
    if(b!=c){
        return S;
    }
    return -1;
}
```

3. Formula 2:

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

Codes:

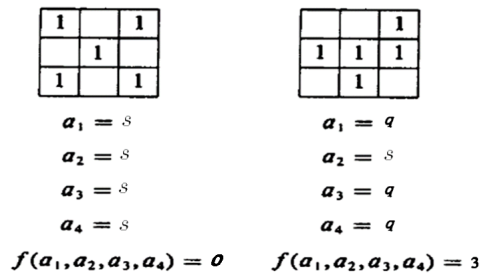
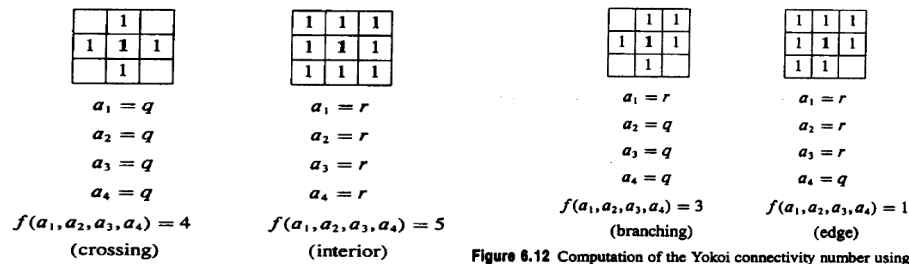
```

int F4(int a1, int a2, int a3, int a4){
    if (a1 == a2 && a1 == a3 && a1==a4 && a1==R){
        return 5;
    }
    int n=0;
    if (a1 == Q) n++;
    if (a2 == Q) n++;
    if (a3 == Q) n++;
    if (a4 == Q) n++;
    return n;
}

```

#### 4. Yokoi Connectivity Number

idea:



Codes:

```

void YokoiConnectivityNumber(Mat *image , vector<string>& res){
    Mat t_image;
    image->copyTo(t_image);
    binary_image(&t_image,128);
    int mv[8][2] = { 0,1, -1,0, 0,-1, 1,0, 1,1,-1,1, -1,-1, 1,-1 };
    int x[9],a[4];
    int rows = t_image.rows;
    int cols = t_image.cols;
    int tr,tc;

    res.resize(rows,string(cols, ' '));

    for (int i=0;i<rows;i++){
        for (int j=0;j<cols;j++){
            x[0] = pixel_get(&t_image,i,j)/255;
            if (x[0] == 0){
                continue;
            }
            for (int k=0;k<8;k++){
                tr = i+mv[k][0];
                tc = j+mv[k][1];
                if (tr< 0 || tr>=rows || tc<0 || tc>=cols){
                    x[k+1] = 0;
                }
                else{
                    x[k+1] = pixel_get(&t_image,tr,tc)/255;
                }
            }
            a[0] = H4(x[0],x[1],x[6],x[2]);
            a[1] = H4(x[0],x[2],x[7],x[3]);
            a[2] = H4(x[0],x[3],x[8],x[4]);
            a[3] = H4(x[0],x[4],x[5],x[1]);
            res[i][j] = F4(a[0],a[1],a[2],a[3])+ '0';
        }
    }
}

```

Result:

11111111	121111111111122322221	111111111111	0 0
15555551	115555555511 2 11 11	1155555555511	0
15555551	1 2115555112 21112221	155555555551	21
15555551	1 2 155112 22221511	1555555555511	1
15555551	22 2112 22 121 0 0	1555555555511	0
15555551	1 2 21 2 1 1	1555555555551	0
15555551	12 1 121111 1321	15555555555511	
15111551	1322 1155551111	15555555555551	
111 1551	1 12155555511	155555555555511	
11 1551	2115555511	1551115555511	
21 1551	2 15555555111	1551 11555511	
1 1551	2 15555555511	1551 115551	1
1551	11211555555551	1551 15511	12
1551	155555555555511	1551 1111	111
1551	1 22211555555555511	1151 11	1151
1551	2 22 1 155555555555511	151 11111	1551
1551	2 1 1155555555555551	151 115551	11551
1551	2 115555555555555111511155511		115551
1551	12 1155555555555555555551		155551
1551	11 0 22155555555555555555555112		1155551
1551	111 22 1555555555555555555551	1	1555551
1551	1511 1 1251121111121115555555511		11555551
1551	15521 1 121 1 11 1 155555511	0	15555551
1551	1151 132 2 1155555111	0	115555551
1551	151 0 322 115555111	121	155555551
1551	1221 2 1555551	131	1155555551
1551	2 0 1 115555511	1	1155555551
1551	2 0 0 1155555551	0	1 155555551
1551	2 1155555551		2115555551
1551	1 0 11555555551		1555555551
1551	1 11511115555521	1	11555555551
1551	1 1 11111 1155511	2	15555555551
1551	131 111 15111	2	15555555551
1551	121 0 1121 1 111 1	2	115555555551
1551	11 111 1 221 11 1	2	155555555551
1551	12 0 1 21 121 11 1111	2	155555555551
1551	1 12 22 151111111551	2	1155555555551
1551	1 2 1555551115511	1	1555555555551
1551	2 0 0 22 12555551 15551	1	1555555555551
1551	1 1 1555511 11511	2	11555555555551
1551	0 0 21 155551 1 151	2	15555555555551
1551	2 15555112 151	2	15555555555551
1551	1 1 1 1155555511111	2	15555555555551
1551	2 22 111511111212		211555555555551
1551	0 1 12 151 2 1		1555555511155551
1551	0 0 0 1111 121		155555551 1555551
1551	0 11111111		155555551 1555551
1551	0 115551		155555551 1555511
1551	15551		211111111 155511
11521	1 12 122155511	2	11 115511
1 151 0	1 1 155555111	2111	15511
22 1511	1 15555555111	155111	1511
22 1511	1 1555555551	155551	1151
2 151	0 1 1115555555511	155511	1511
2 1521	0 1 15555555555511	15551 12151	
2 151	121 15555555555551	155511 1551	
2 1511	0 15555555555551	115551 1511	
21 1511	11 15555555555551	111111151	
11 151	0 1155555555555511	111511	
11 151	0 1555555555555551	151	
11 151	0 11555555555555551	211	
11 151	0 11555555555555511	1	
11 151	0 15555555555555551		
11 111	0 1211111111111111111		

## 5. Appendix

- i. build\_all.sh  
"sh build\_all.sh" will automatically compile the code in terminal.
- ii. R01922124\_HW6.cpp  
source code
- iii. lena.bmp  
original lena image
- iv. result.txt  
results for this homework
- v. R01922124\_HW6.pdf  
report