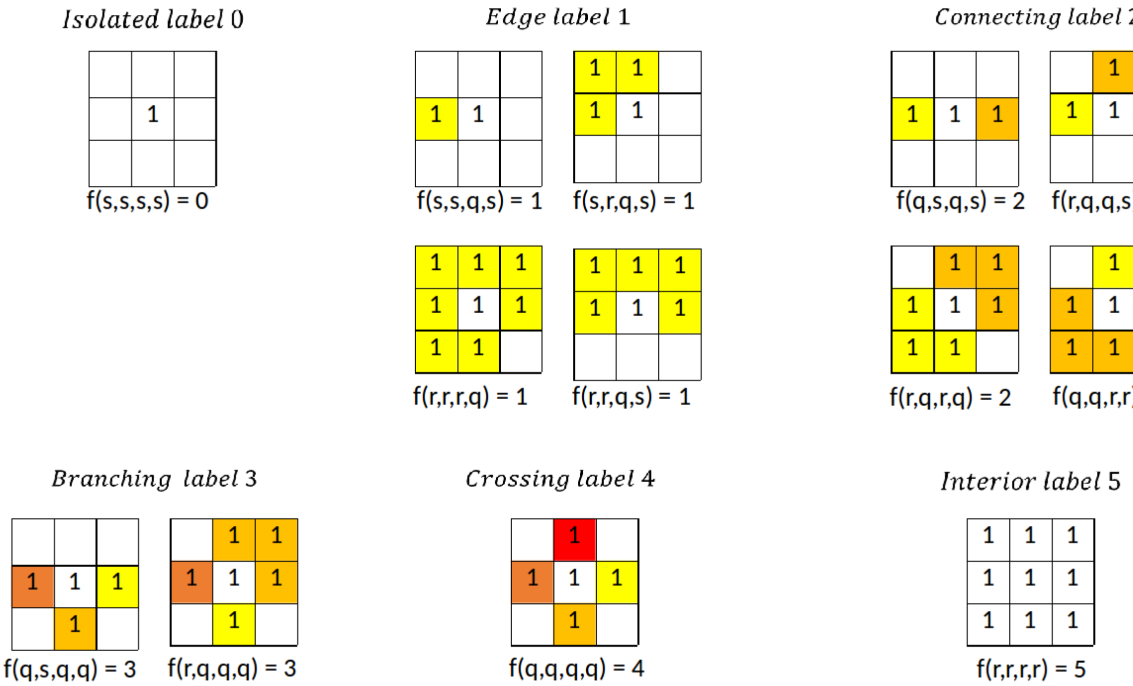
Computer vision Hw6

R07921010 余奇安

1.Yokoi connectivity number

Use Yokoi connectivity number to recognize the property of every pixel

1:edge label
2:connecting label
3:branching label
4:crossing label
5:interior label
0:isolated label



Step: Binary gray scale lena at 128
downside from 512x512 to 64x64 (choose the top left right pixel in 8x8 grid)
do the yokoi algorithm for 4 connectivity

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

a1=h(b,c,d,e)=h(x0,x1,x6,x2)
a2=h(b,c,d,e)=h(x0,x2,x7,x3)
a3=h(b,c,d,e)=h(x0,x3,x8,x4)
a4=h(b,c,d,e)=h(x0,x4,x5,x1)

x_7	x_2	x_6
x_3	x_0	x_1
x_8	x_4	x_5

define f function:

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

to determine the number of label.

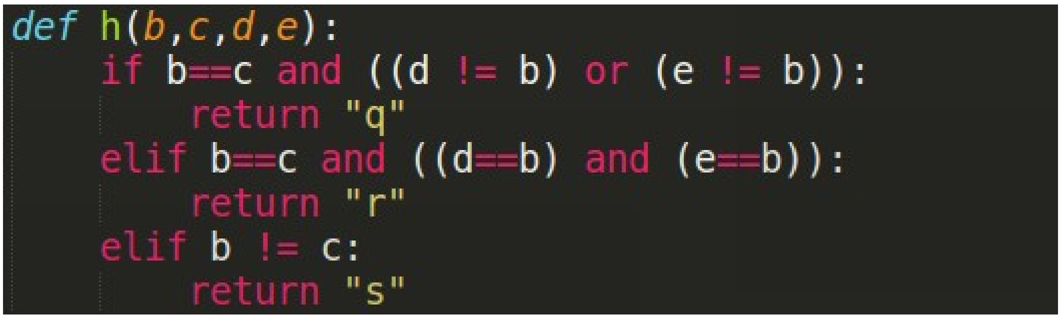
Result:

```
1 1 1 1 1 1 1 1                 1 2 1     1 1 1 1 1 1 1 2 2 3 2 2 2 2 1           1 1 1 1 1 1 1 1 1 1 1 1               0     0
1 5 5 5 5 5 5 1                   1 2 1   1 5 5 5 5 5 1 1   2   1 1     2 1       1 1 5 5 5 5 5 5 5 5 5 1 1
1 5 5 5 5 5 5 1                 1   2 2 2 1 5 5 5 1 1 2     2 1 1 1 2 2 2           1 5 5 5 5 5 5 5 5 5 5 1               0
1 5 5 5 5 5 5 1                 1   2   1 5 5 1 1 2   1     1 1 5 5 1               1 5 5 5 5 5 5 5 5 5 5 1 1           0
1 5 5 5 5 5 5 1                     1   2 1 1 2   1           1 2 1 1     0         1 5 5 5 5 5 5 5 5 5 5 5 1
1 5 5 5 5 5 5 1                   0     1     2 1   1           1                   1 1 5 5 5 5 5 5 5 5 5 5 1 1
1 5 5 5 5 5 5 1                       1     1     1 2 1 1 1 1         1 1   0         1 5 5 5 5 5 5 5 5 5 5 5 1 1
1 5 1 1 1 5 5 1                       2 3 2 2   1 2 5 5 5 5 1 1 1 1                 1 1 5 5 5 5 5 5 5 5 5 5 5 5 1
1 1 1   1 5 5 1                         1     0   1 5 5 5 5 5 5 5 1 1               1 5 5 5 5 5 5 5 5 5 5 5 5 1 1
1 1     1 5 5 1                                   2 1 1 5 5 5 5 5 5 1 1             1 5 5 1 1 1 5 5 5 5 5 5 1 1
2 1     1 5 5 1                                   2   1 5 5 5 5 5 5 5 1 1 1         1 5 5 1   1 1 5 5 5 5 1 1
1       1 5 5 1                                   2   1 5 5 5 5 5 5 5 5 5 1 1       1 5 5 1     1 1 5 5 5 1                   1
        1 5 5 1                               1 1 2 1 1 5 5 5 5 5 5 5 5 5 5 1       1 5 5 1       1 5 5 1 1                 1 2
        1 5 5 1                               1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1     1 1 5 1       1 1 1 1                 1 1 1
        1 5 5 1                 1           2 2 2 1 1 5 5 5 5 5 5 5 5 5 5 5 5 1 1     1 5 1         1 1                 1 1 5 1
        1 5 5 1                 2         1 2   1   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1   1 5 1     1 1 1 1 1               1 5 5 1
        1 5 5 1                 2                 1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1   1 5 1   1 1 5 5 5 1             1 1 5 5 1
        1 5 5 1                 2               1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1 1 5 1 1 1 5 5 5 1 1           1 1 5 5 5 1
        1 5 5 1               1 2             1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1             1 5 5 5 5 1
        1 5 5 1               1 1       0   2 2 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1 2           1 1 5 5 5 5 1
        1 5 5 1               1 1 1       2 2   1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1   1           1 5 5 5 5 5 1
        1 5 5 1               1 5 1 1     1   1 2 5 1 1 2 1 1 1 1 1 2 1 1 1 5 5 5 5 5 5 5 5 5 1 1 1             1 1 5 5 5 5 5 1
        1 5 5 1               1 1 1 2       1   1 2 1   1   1 1     1     1 5 5 5 5 5 5 5 1 1 1     0           1 5 5 5 5 5 5 1
        1 5 5 1               1 2   2     1 3 2   2                     1 1 5 5 5 5 5 1 1 1       0           1 1 5 5 5 5 5 5 1
        1 5 5 1                 2 2 2   0     3 2 2                   1 1 5 5 5 5 1 1 1     1 2 1             1 5 5 5 5 5 5 5 1
        1 5 5 1                 1 2   0       1                       1 5 5 5 5 5 1       1 3 1             1 2 5 5 5 5 5 5 5 1
        1 5 5 1                   2     0                           1 1 5 5 5 5 5 1 1       1                 1 5 5 5 5 5 5 5 1
        1 5 5 1                   2       0                       1 1 5 5 5 5 5 5 5 1     0               1   1 5 5 5 5 5 5 5 1
        1 5 5 1                   2                             1 1 5 5 5 5 5 5 5 5 1                     2 1 1 5 5 5 5 5 5 5 1
        1 5 5 1                   1     0                     1 1 5 5 5 5 5 5 5 5 5 1                     1 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                                             1 1 5 1 1 1 1 5 5 5 5 5 2 1     1           1 1 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                 1   0                     1 1 1 1 1     1 1 5 5 5 1 1       2           1 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1               1 3 1                       1 1 1           1 5 1 1 1         2           1 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1             1 2 1                     1 1 2 1       1     1 1 1     1       2         1 1 5 5 5 5 5 5 5 5 1 1 2
        1 5 5 1             1 1                       1 1 1   1     2 2 1   1 1     1       2         1 5 5 5 5 5 5 5 5 5 1   2
        1 5 5 1         1 2     0         1           2 1   1 2 1     1 1   1 1 1 1         2         1 5 5 5 5 5 5 5 5 5 1 1 2
        1 5 5 1           1             1 2         2 2     1 5 1 1 1 1 1 1 1 5 5 1         2       1 1 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1       1                             2       1 5 5 5 5 5 1 1 1 5 5 1 1       1       1 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1       2         0           0     2 2     1 2 5 5 5 5 5 1   1 5 5 5 1         1   1 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1       1                           1         1 5 5 5 5 1 1   1 1 5 1 1         2   1 1 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1             0   0             2 1           1 5 5 5 5 1   1   1 5 1           2   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                               2             1 5 5 5 5 1 1 2   1 5 1           2   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                   1       1   1           1 2 5 5 5 5 5 5 1 1 1 1 1           2   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                   2     2 2                 1 1 5 1 1 1 1 1 2 1 2             2 1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1     0             1   1 2                     1 1 1         2   1             1 5 5 5 5 5 5 5 1 1 1 5 5 5 5 5 1
        1 5 5 1               0     0     0                     1 1 1     1 2 1               1 5 5 5 5 5 5 5 1   1 5 5 5 5 5 1
        1 5 5 1                         0                       1 1 1 1 1 1 1 1               1 5 5 5 5 5 5 5 1   1 5 5 5 5 5 1
        1 5 5 1                 0                                 1 1 5 5 5 1                 1 5 5 5 5 5 5 5 1   1 5 5 5 5 1 1
        1 5 5 1                                                     1 5 5 5 1                 2 1 1 1 1 1 1 1 1   1 5 5 5 1 1
        1 1 5 2 1             1       1 2                     0   1 1 5 5 5 1 1               2           1 1   1 1 5 5 1 1
          1 5 1   0           1         1                         1 5 5 5 5 5 1 1 1           2 1 1 1           1 5 5 1 1
1 2       1 5 1 1                     1                           1 5 5 5 5 5 5 5 1 1 1       1 5 5 1 1 1       1 5 1 1
  2 2     1 5 1 1                     1                           1 5 5 5 5 5 5 5 5 5 1       1 5 5 5 5 1     1 1 5 1
    2     1 5 1                         0                       1 1 5 5 5 5 5 5 5 5 5 1 1     1 5 5 5 1 1     1 5 1 1
    2     1 5 2 1         0                 0                 1 1 5 5 5 5 5 5 5 5 5 5 5 1 1   1 5 5 5 1   1 2 1 5 1
    2     1 5 1                       1 2 1                   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1   1 5 5 5 1 1   1 5 5 1
    2     1 5 1 1                                       0     1 5 5 5 5 5 5 5 5 5 5 5 5 5 1   1 1 5 5 5 1   1 5 1 1
    2 1   1 5 1 1                           0                 1 5 5 5 5 5 5 5 5 5 5 5 5 5 1     1 1 1 1 1 1 1 5 1
    1 1   1 5 1                     0                       1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1         1 1 1 5 1 1
    1 1   1 5 1                                             1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1             1 5 1
    1 1   1 5 1                                           1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1             2 1 1
    1 1   1 5 1                                           1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1           1
    1 1   1 5 1                                         0   1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1
    1 1   1 1 1                 0                         1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

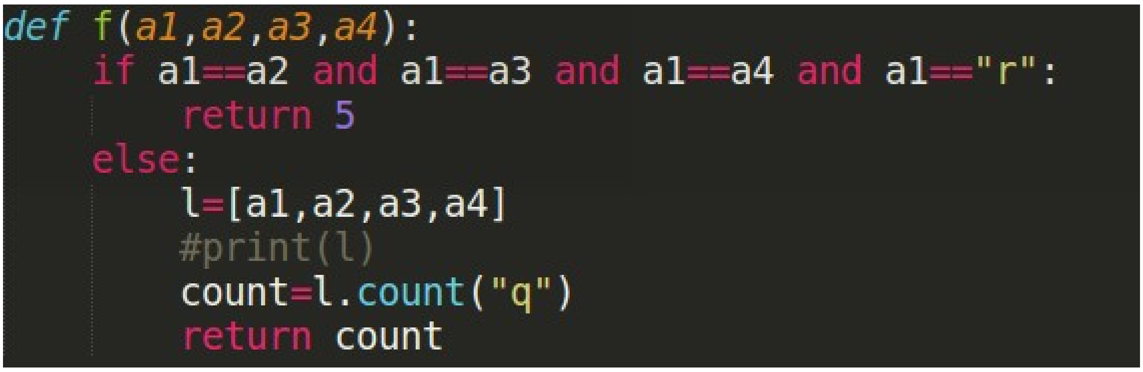
Code:

follow the yokoi algorithm

define h function and f function first

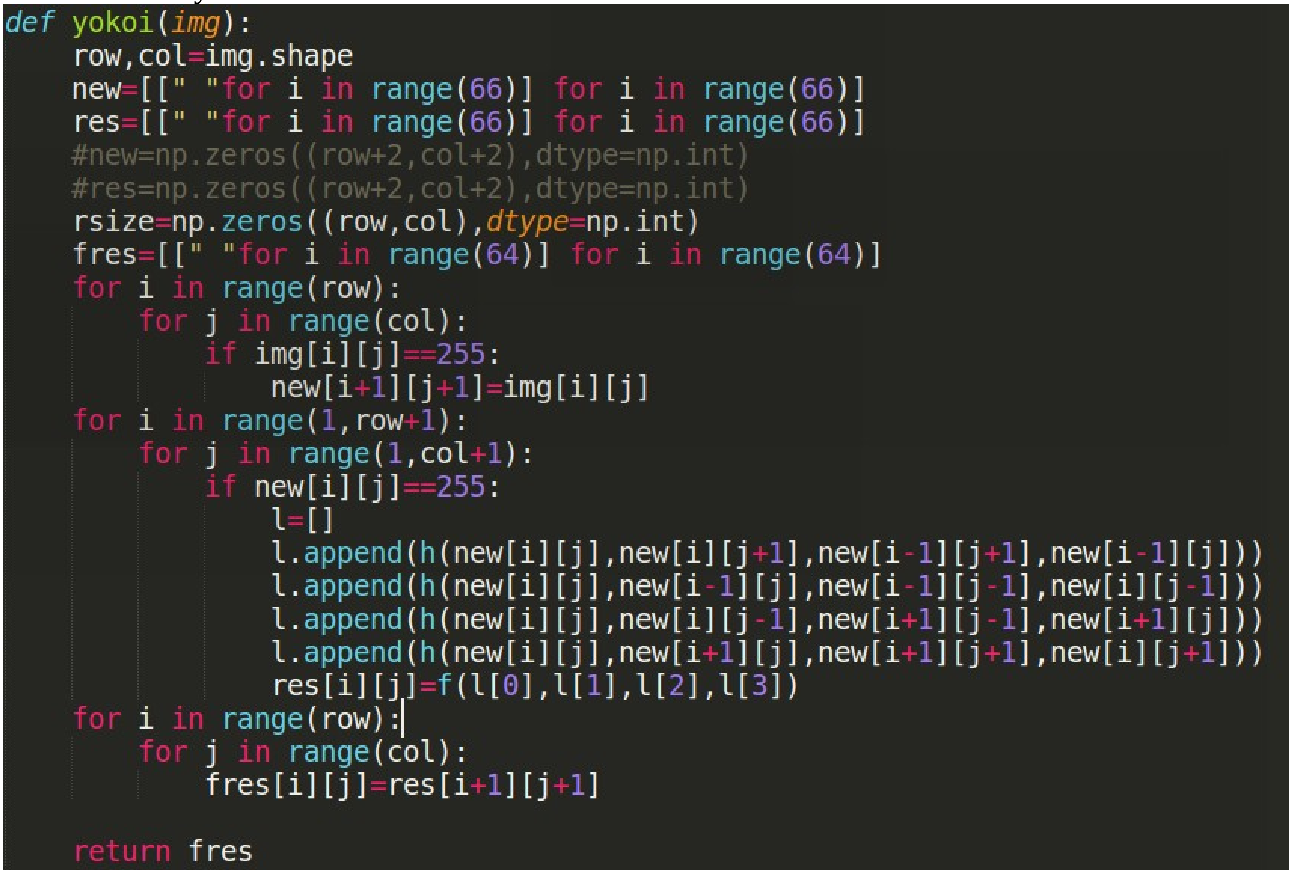


```
def h(b,c,d,e):
    if b==c and ((d != b) or (e != b)):
        return "q"
    elif b==c and ((d==b) and (e==b)):
        return "r"
    elif b != c:
        return "s"
```



```python
def f(a1,a2,a3,a4):
    if a1==a2 and a1==a3 and a1==a4 and a1=="r":
        return 5
    else:
        l=[a1,a2,a3,a4]
        #print(l)
        count=l.count("q")
        return count
```

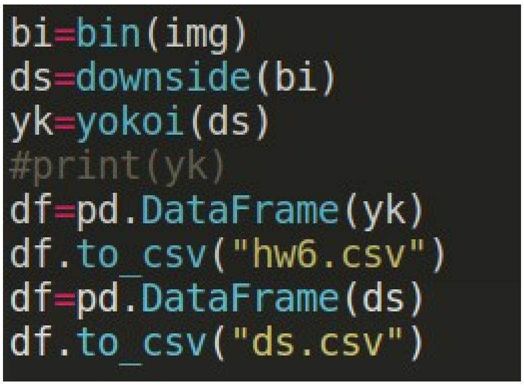
Second define yokoi function



```python
def yokoi(img):
    row,col=img.shape
    new=[[" "for i in range(66)] for i in range(66)]
    res=[[" "for i in range(66)] for i in range(66)]
    #new=np.zeros((row+2,col+2),dtype=np.int)
    #res=np.zeros((row+2,col+2),dtype=np.int)
    rsize=np.zeros((row,col),dtype=np.int)
    fres=[[" "for i in range(64)] for i in range(64)]
    for i in range(row):
        for j in range(col):
            if img[i][j]==255:
                new[i+1][j+1]=img[i][j]
    for i in range(1,row+1):
        for j in range(1,col+1):
            if new[i][j]==255:
                l=[]
                l.append(h(new[i][j],new[i][j+1],new[i-1][j+1],new[i-1][j]))
                l.append(h(new[i][j],new[i-1][j],new[i-1][j-1],new[i][j-1]))
                l.append(h(new[i][j],new[i][j-1],new[i+1][j-1],new[i+1][j]))
                l.append(h(new[i][j],new[i+1][j],new[i+1][j+1],new[i][j+1]))
                res[i][j]=f(l[0],l[1],l[2],l[3])
    for i in range(row):
        for j in range(col):
            fres[i][j]=res[i+1][j+1]

    return fres
```

Finally, use pandas to output the matrix



```python
bi=bin(img)
ds=downside(bi)
yk=yokoi(ds)
#print(yk)
df=pd.DataFrame(yk)
df.to_csv("hw6.csv")
df=pd.DataFrame(ds)
df.to_csv("ds.csv")
```