Howework 7 Report

Student:鞠之浩 ID:P10942A08

Output： 

Code describe：Step1. Binarize lena.bmp

```python
#binarize
def binarize(lena,ans):
    for i in range(lena.shape[0]):
        for j in range(lena.shape[1]):
            if lena[i][j]>=128:
                ans[i][j] = 1
            else:
                ans[i][j] = 0
    return ans
```

Step2. Downsampling to 64X64

```python
#downsampling
def downsampling(img):
    row, col = int(img.shape[0]/8), int(img.shape[1]/8)
    downsampling_img = np.zeros([row,col], np.int)
    for i in range(row):
        for j in range (col):
            downsampling_img[i,j] = img[i*8,j*8]
    return downsampling_img
```

Step3. Write h function

```python
def yokoi_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'
    else:
        return 's'
```

Step4. Write f function

```python
def yokoi_f(a1, a2, a3, a4):
    if (a1 == 'r' and a2 == 'r' and a3 == 'r' and a4 == 'r'):
        return 5
    else:
        n=0
        if a1 == 'q':
            n += 1
        if a2 == 'q':
            n += 1
        if a3 == 'q':
            n += 1
        if a4 == 'q':
            n +=1
        return n
```

Step5. Scan lena.bmp, to get yokoi connectivity number by calling f function.

```python
def yokoi(img):
    row, col = img.shape[0]-2, img.shape[1]-2
    yokoi_matrix = np.zeros([row, col], np.int)
    for i in range(1, row+1):
        for j in range(1, col+1):
            if img[i, j] == 1:
                a1 = yokoi_h(img[i, j], img[i, j+1], img[i-1, j+1], img[i-1, j])
                a2 = yokoi_h(img[i, j], img[i-1, j], img[i-1, j-1], img[i, j-1])
                a3 = yokoi_h(img[i, j], img[i, j-1], img[i+1, j-1], img[i+1, j])
                a4 = yokoi_h(img[i, j], img[i+1, j], img[i+1, j+1], img[i, j+1])
                yokoi_matrix[i-1, j-1] = yokoi_f(a1, a2, a3, a4)
    return yokoi_matrix
```

Step6. Write h function for shrink

```python
def shrink_h(b, c, d, e):
    if b == c and (b != d or b != e):
        return 1
    else:
        return 0
```

Step7. Write f function for shrink

```python
def shrink_f(a1, a2, a3, a4):
    cnt = 0
    if a1 == 1:
        cnt += 1
    if a2 == 1:
        cnt += 1
    if a3 == 1:
        cnt += 1
    if a4 == 1:
        cnt += 1
    if cnt == 1:
        return 1
    else:
        return 0
```

Step8. Write pair relationship function

```python
def pair_relationship(yokoi_matrix):
    row, col = yokoi_matrix.shape[0]-2, yokoi_matrix.shape[1]-2
    pair_relationship_matirx = np.zeros([row, col], np.str)
    for i in range(1, row+1):
        for j in range(1, col+1):
            if (yokoi_matrix[i, j] != 1) or (yokoi_matrix[i-1, j] != 1 and yokoi_matrix[i, j-1] != 1 and yokoi_matrix[i+1, j] != 1 and yokoi_matrix[i, j+1] != 1):
                pair_relationship_matirx[i-1, j-1] = 'q'
            elif (yokoi_matrix[i, j] == 1) and (yokoi_matrix[i-1, j] == 1 or yokoi_matrix[i, j-1] == 1 or yokoi_matrix[i+1, j] == 1 or yokoi_matrix[i, j+1] != 1):
                pair_relationship_matirx[i-1, j-1] = 'p'
    return pair_relationship_matirx
```

Step9. Shrink operation

```python
def shrink(pair_relationship_matrix, img):
    row, col = img.shape[0]-2, img.shape[1]-2
    for i in range(1, row+1):
        for j in range(1, col+1):
            if img[i, j] == 1:
                a1 = shrink_h(img[i, j], img[i, j+1], img[i-1, j+1], img[i-1, j])
                a2 = shrink_h(img[i, j], img[i-1, j], img[i-1, j-1], img[i, j-1])
                a3 = shrink_h(img[i, j], img[i, j-1], img[i+1, j-1], img[i+1, j])
                a4 = shrink_h(img[i, j], img[i+1, j], img[i+1, j+1], img[i, j+1])
                if pair_relationship_matrix[i-1, j-1] == 'p':
                    if shrink_f(a1, a2, a3, a4):
                        img[i, j] = 0
    return img
```

Step10. Repeat step 5,8,9 until the output hasn’ t been changed.