HW7
r10944013 網媒所碩一 馮啟倫

HW7:
目標：做Thinning operation

演算法：
1. 先做downsampling (512*512) -> (64*64)
2. 對圖片計算Yokoi, 再對圖片做 pair relation operator（透過Yokoi所得到的數值計算，得到p or q值）
3. 最後透過pair relation得到的p,q圖，當原圖的pixel有值且對應的pair relation的值是q，那就把這一個點給shrink掉，然後再recursive的重複1~3步驟，直接步驟一的圖片跟步驟三output的圖片完全相等

Code segment

#yokoi

```
def cal_score(x0,x1,x2,x3,x4,x5,x6,x7,x8):
    condition = []
    #first dimension
    if x1==x0:
        if x2==x0 and x6==x0:
            condition.append("r")
        else:
            condition.append("q")
    else:
        condition.append("s")

    #second dimension
    if x2==x0:
        if x7==x0 and x3==x0:
            condition.append("r")
        else:
            condition.append("q")
    else:
        condition.append("s")

    #third dimension
    if x3==x0:
        if x8==x0 and x4==x0:
            condition.append("r")
        else:
            condition.append("q")
    else:
        condition.append("s")

    #third dimension
    if x4==x0:
        if x1==x0 and x5==x0:
            condition.append("r")
        else:
            condition.append("q")
    else:
        condition.append("s")
    #Check for q and r
    if condition.count("r") == 4:
        return 5
    else:
        return condition.count("q")

def cal_yokoi(img_downsample):
    #padding
    img_pad = np.zeros((66,66))
    for row in range(img_downsample.shape[0]):
        for col in range(img_downsample.shape[1]):
            img_pad[row+1][col+1] = img_downsample[row][col]

    img_yokoi = np.zeros((64,64))
    for row in range(img_downsample.shape[0]):
        for col in range(img_downsample.shape[1]):
            center_row = row+1
            center_col = col+1
            x0 = img_pad[center_row][center_col]
            x1 = img_pad[center_row][center_col+1]
            x2 = img_pad[center_row-1][center_col]
            x3 = img_pad[center_row][center_col-1]
            x4 = img_pad[center_row+1][center_col]
            x5 = img_pad[center_row+1][center_col+1]
            x6 = img_pad[center_row-1][center_col+1]
            x7 = img_pad[center_row-1][center_col-1]
            x8 = img_pad[center_row+1][center_col-1]
            if x0 != 0:
                score = cal_score(x0,x1,x2,x3,x4,x5,x6,x7,x8)
                img_yokoi[row][col] = score
    return img_yokoi
```

pair operation 與最後的thin operation

```python
def cal_pair(img):
    #input: yokoi image

    #padding
    img_pad = np.zeros((66,66))
    for row in range(img.shape[0]):
        for col in range(img.shape[1]):
            img_pad[row+1][col+1] = img[row][col]

    img_pair = np.zeros((64,64), dtype="object")
    for row in range(img.shape[0]):
        for col in range(img.shape[1]):
            center_row = row+1
            center_col = col+1
            x0 = img_pad[center_row][center_col]
            x1 = img_pad[center_row][center_col+1]
            x2 = img_pad[center_row-1][center_col]
            x3 = img_pad[center_row][center_col-1]
            x4 = img_pad[center_row+1][center_col]
            if x0 != 0:
                if x0 != 1:
                    img_pair[row][col] = "q"
                    continue
                #確保底下都是1了
                temp_score = 0
                for i in [x1,x2,x3,x4]:
                    if i == 1:
                        temp_score +=1

                if temp_score  >= 1:
                    img_pair[row][col] = "p"
                else:
                    img_pair[row][col] = "q"
    return img_pair
```

```python
def cal_thin(img, img_pair):
    #padding
    img_pad = np.zeros((66,66))
    for row in range(img.shape[0]):
        for col in range(img.shape[1]):
            img_pad[row+1][col+1] = img[row][col]

    img_thin = np.zeros((64,64), dtype="object")
    for row in range(img.shape[0]):
        for col in range(img.shape[1]):
            center_row = row+1
            center_col = col+1
            x0 = img_pad[center_row][center_col]
            x1 = img_pad[center_row][center_col+1]
            x2 = img_pad[center_row-1][center_col]
            x3 = img_pad[center_row][center_col-1]
            x4 = img_pad[center_row+1][center_col]
            x5 = img_pad[center_row+1][center_col+1]
            x6 = img_pad[center_row-1][center_col+1]
            x7 = img_pad[center_row-1][center_col-1]
            x8 = img_pad[center_row+1][center_col-1]

            if x0 != 0 and img_pair[row][col] == "p": #符合shrink的檢驗條件
                score = cal_score(x0,x1,x2,x3,x4,x5,x6,x7,x8)
                if score == 1: #同時修改我最後的output跟正在計算的matrix
                    img_thin[row][col] = 0
                    img_pad[center_row][center_col] = 0
                else:
                    img_thin[row][col] = img_pad[center_row][center_col]
            else:
                img_thin[row][col] = img_pad[center_row][center_col]
    return img_thin
```

#main function

```python
while True:
    img_downsample_old = copy.deepcopy(img_downsample)

    img_yokoi = cal_yokoi(img_downsample)
    img_pair = cal_pair(img_yokoi)
    #img_shrink = cal_shrink(img_downsample)
    img_downsample = cal_thin(img_downsample, img_pair)

    #remove point
    #for row in range(64):
    #    for col in range(64):
    #        if img_shrink[row][col] == 'g':
    #            if img_pair[row][col] == 'p':
    #                img_downsample[row][col] = 0

    #check similarity
    if np.sum(img_downsample == img_downsample_old) == 64*64:
        break

#還原回255
img_ans = np.zeros((64,64))
for i in range(img_downsample.shape[0]):
    for j in range(img_downsample.shape[1]):
        if img_downsample[i][j] == 1:
            img_ans[i][j] = 255
```

Result picture: