

1. 將 Lena 從 512x512 降採樣到 64x64。
2. 將基準圖像 lena 與 HW2 一樣二值化，然後以 8x8 塊為單位，將最左上角的像素作為降採樣數據。
3. 使用 4-connected 計算下採樣 lena 上的 Yokoi 連接數。
4. 此分配的結果是一個 64x64 矩陣。
5. 利用YOKOI OPERATOR 及pair relationship operator，產生 marked image
6. 再利用connect shrink operator來產生shrink 後的結果，
7. 並且每次與唱次的結果作比較，當不再變化時，才跳出迴圈

```

padding=np.zeros((66,66),dtype=int)
for i in range(64):
    for j in range(64):
        padding[i+1][j+1]=Downsampling[i][j]
yokoi=np.zeros((64,64),dtype=int)
for i in range(66):
    for j in range(66):
        if(padding[i][j]==255 and i-1>=0 and i+1<=65 and j-1>=0 and j+1<=65):
            x0=padding[i][j]
            x1=padding[i][j+1]
            x2=padding[i-1][j]
            x3=padding[i][j-1]
            x4=padding[i+1][j]
            x5=padding[i+1][j+1]
            x6=padding[i-1][j+1]
            x7=padding[i-1][j-1]
            x8=padding[i+1][j-1]
            a1=h(x0,x1,x6,x2)
            a2=h(x0,x2,x7,x3)
            a3=h(x0,x3,x8,x4)
            a4=h(x0,x4,x5,x1)
            yokoi[i-1][j-1]=f(a1,a2,a3,a4)

#PairRelationOperator
for i in range(64):
    for j in range(64):
        padding[i+1][j+1]=yokoi[i][j]
for i in range(1,65):
    for j in range(1,65):
        if padding[i][j]==0:#背景
            thinning[i-1][j-1]=0
        elif padding[i][j]==1:#edge
            if padding[i+1][j]!=1 and padding[i-1][j]!=1 and padding[i][j+1]!=1 and padding[i][j-1]!=1:#
                thinning[i-1][j-1]=2#q
            else:
                thinning[i-1][j-1]=1#p
        else:#yokoi值為2or3or4or5時，設為q
            thinning[i-1][j-1]=2#q

```

1. Pair relation operator:當該點為edge(yokoi值為1)時，若該點的(上下左右的鄰居皆不為edge)，則該點設為q，反之(上下左右的鄰居也存在edge時)，則該點設為P
2. Connect Shrink operator:當Yokoi中計算q的個數為1的話，才把他變成background值；否則維持原樣。