

1. 如同 hw6，把 512*512 的圖以 8*8 的圖分割成 64*64，再以左上的灰階值

作為 64*64 的灰階值。

```
def ds(img):
    ret = np.zeros((66, 66), np.int)
    for i in range(64):
        for j in range(64):
            if img[i * 8, j * 8] >= 128:
                ret[i + 1, j + 1] = 255
            else:
                ret[i + 1, j + 1] = 0
    return ret
```

2. 跟 hw6 一樣，多加一圈在外面避免邊界問題，一樣做 Yokoi number 計算

```
def yokoi(img):
    check = [[0, 1], [1, 1], [1, 0], [1, -1],
              [0, -1], [-1, -1], [-1, 0], [-1, 1]]
    ret = np.zeros((66, 66), np.int)
    for i in range(1, 65):
        for j in range(1, 65):
            if img[j, j] == 255:
                flag = 0
                count = 0
                cnt = 0
                for k in range(9):
                    if img[i + check[k % 8][0], j + check[k % 8][1]] == 255:
                        cnt += 1
                        if flag == 1 and k == 8:
                            count -= 1
                        if flag == 0 and k % 2 == 0 and k < 8:
                            count += 1
                        flag = 1
                else:
                    if flag == 1:
                        flag = 0
                if cnt == 9:
                    ret[i, j] = 5
                else:
                    ret[i, j] = count
    return ret
```

3. Pair relationship operator

當前 pixel 的 yokoi number 為 1 且 4-connected neighbor 中有大於一個

neighbor 其 yokoi number 也為 1 的話，將此 pixel 設為 p，其餘為 q。

實作中 100 代表 p，99 代表 q。

```
def get_h(arr):
    ret = np.zeros((66, 66))
    for i in range(1, 65):
        for j in range(1, 65):
            if arr[i, j] == 1:
                if (i >= 2 and arr[i - 1, j] == 1) or (i < 64 and arr[i + 1, j] == 1) \
                    or (j >= 2 and arr[i, j - 1] == 1) or (j < 64 and arr[i, j + 1] == 1):
                    ret[i, j] = 100
            elif arr[i, j] != 0:
                ret[i, j] = 99
    return ret
```

4. 類似計算 yokoi number，不同在於當 pixel 的 yokoi number 為 1 且

get_h 算出來為 p 時要將 pixel 改為 0。

```
def thinning(img, img2):
    check = [[0, 1], [1, 1], [1, 0], [1, -1],
              [0, -1], [-1, -1], [-1, 0], [-1, 1]]
    for i in range(1, 65):
        for j in range(1, 65):
            if img[i, j] == 255:
                flag = 0
                num = 0
                cnt = 0
                for k in range(9):
                    if img[i + check[k % 8][0], j + check[k % 8][1]] == 255:
                        cnt += 1
                        if flag == 1 and k == 8:
                            num -= 1
                        if flag == 0 and k % 2 == 0 and k < 8:
                            num += 1
                            flag = 1
                    else:
                        if flag == 1:
                            flag = 0
                if num == 1 and img2[i, j] == 100:
                    img[i, j] = 0
    return img
```

將以上步驟重複做七遍，得到下面這張圖：

