

# 甲状腺前景提取报告

肖蔚尔 520030910314

## 1. 完成度（黑色为中期答辩完成，红色为结题新增）

任务要点	完成情况	负责人
先提取甲状腺前景图像	√（使用 nnunet 进行改进）	全体
根据打点标记获取病灶 ROI	√（使用 nnunet 进行改进）	王奕文
根据打点距离信息计算 ROI 大小	√（针对无打点结节也可以进行处理）	王奕文
提取 ROI 中甲状腺结节的边缘、分辨边缘的规则程度、清晰程度	√	王奕文
识别结节内质的基本特性（囊性、实性、囊实性）；	√	童楚炎
判断结节内部是否存在结晶或者钙化	√	倪申越
在甲状腺内寻找钙化位置；	√	倪申越
依照图像右下角位置信息和图像呈现的内容判断当前图像的取材位置信息（切面横/纵、甲状腺左/右叶）。	√	肖蔚尔
无小图标图片获得取材信息、结节相对位置信息（表面/背侧、内/外侧、上/中/下）、最长轴垂直/平行皮肤。	√	肖蔚尔
判断结节的边缘是光滑还是毛刺状、结节形状的细节特征	√	王奕文
判断结节的声晕厚度	√	童楚炎
对结节内质进行更细致的分类，并判断结节内是否有钙化；	√	倪申越
对甲状腺内钙化位置的情况进行分类。	√	倪申越
Ui 框架编写	√	童楚炎
功能整合	√	童楚炎、王奕文

## 2. 取材信息

### 2.1. 有小图标

1. 加载模板图像提取小图标，使用 `cv2.matchTemplate()` 的 `TM_CCOEFF_NORMED` 方法进行图像匹配，过滤匹配度低的区域

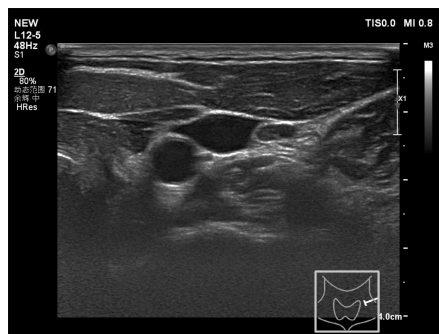


预处理得到的模板图片

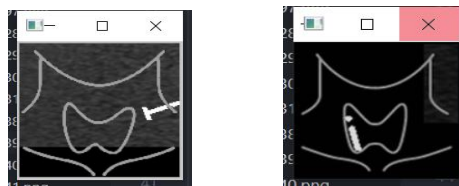
```
# 设定阈值，用于过滤匹配度低的区域
threshold_icon = 0.6
# 过滤匹配度低的区域
loc1 = np.where(res1 >= threshold_icon)
```

过滤

匹配结果如下：



2. 绘制边框并剪裁小图标得到如下结果：



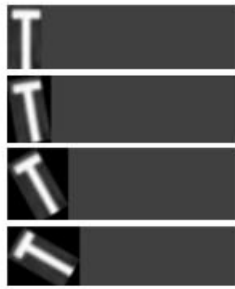
### 3. 处理小图标

① 定义旋转角度和旋转中心

② 计算旋转后的图像大小，并计算平移矩阵（防止旋转后超出原图区域的部分被遮盖）

```
M = cv2.getRotationMatrix2D(center, angle, 1)
cos = np.abs(M[0, 0])
sin = np.abs(M[0, 1])
new_width = int((focus.shape[0] * sin) + (focus.shape[1] * cos))
new_height = int((focus.shape[0] * cos) + (focus.shape[1] * sin))
M[0, 2] += (new_width / 2) - center[0]
M[1, 2] += (new_height / 2) - center[1]

# 进行旋转变换
foces_rotated = cv2.warpAffine(focus, M, (new_width, new_height))
```



正确预处理结果



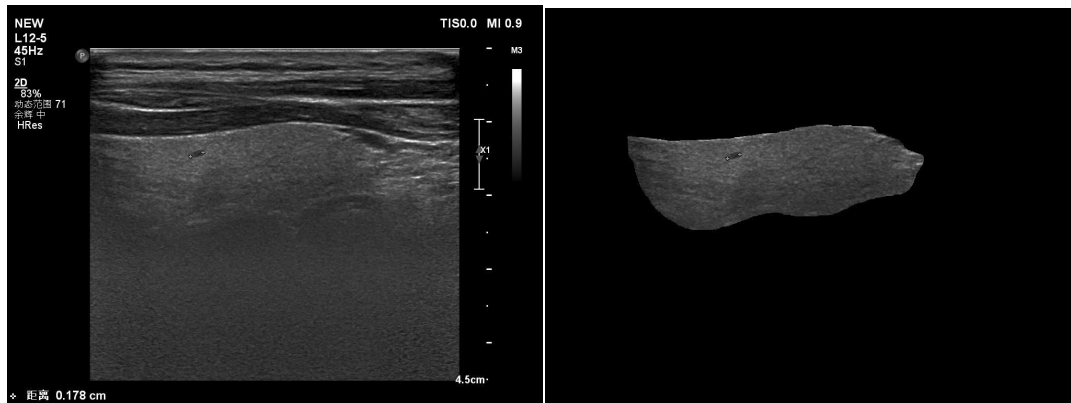
不进行矩阵变换的错误结果

③旋转变换后图像匹配（方法同上），根据匹配成功的图标对应的预处理旋转角度确定切面方向（ $45^{\circ}$  -  $135^{\circ}$ 、 $225^{\circ}$  -  $315^{\circ}$  为纵切）

## 2.2. 没有小图标

用图像匹配识别图标，若无法正确匹配则判定为无图标；

1. 前置操作获取甲状腺前景提取图如下：



2. 高斯滤波平滑处理

```
image = cv2.GaussianBlur(image, (15, 15), 0)
```

3. 使用 `cv2.findContours` 函数进行轮廓检测（`cv2.RETR_EXTERNAL` 指定了轮廓检测的模式为仅检测最外层的轮廓），遍历轮廓并在 `seg` 图中标记：



上图中区域边缘贴合的灰色轮廓不是线条，是轮廓是别出的点绘制出来的。上图是改进后的版本，使用了参数 CHAIN\_APPROX\_NONE，识别出的点较为密集效果较好，而最初使用的参数为 CHAIN\_APPROX\_SIMPLE，其对于平行坐标轴的轮廓只识别并标记线段两端点，得到的轮廓点会出现多端空缺，效果如下：



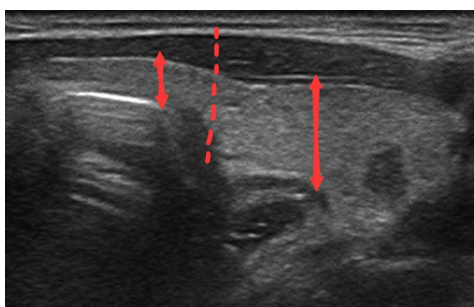
4. 根据识别出的甲状腺像素坐标，取左、中、右三个位置的轮廓点，y 坐标做差得到甲状腺高度差

```
# 筛选符合 x 坐标等于固定值的点的 y 坐标
selected_points = contour[contour[:, :, 0] == i]
while len(selected_points) != 2:
    i += 1
    if i >= x + w:
        print("ERROR:out of range!!!(check_roi_pos)")
        break
    # print("!!!", i)
    selected_points = contour[contour[:, :, 0] == i]
y_coordinates.extend(selected_points[:, 1].tolist())
```

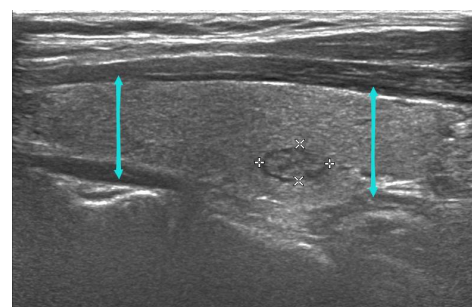
效果如下：



5. 通过判断高度差变化趋势确定横切/纵切、左/右叶



高度差变化明显且向右增大（横切左叶）

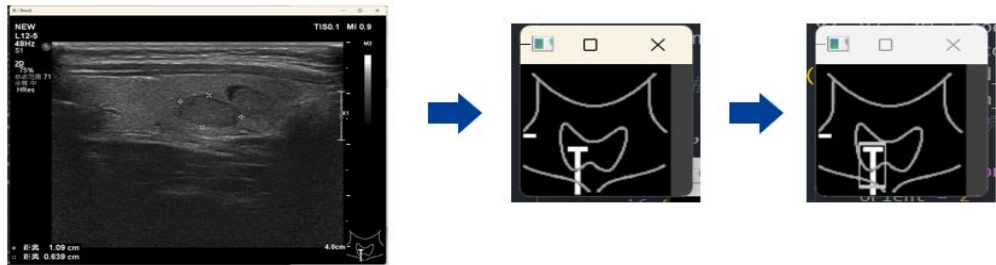


变化不明显（纵切）

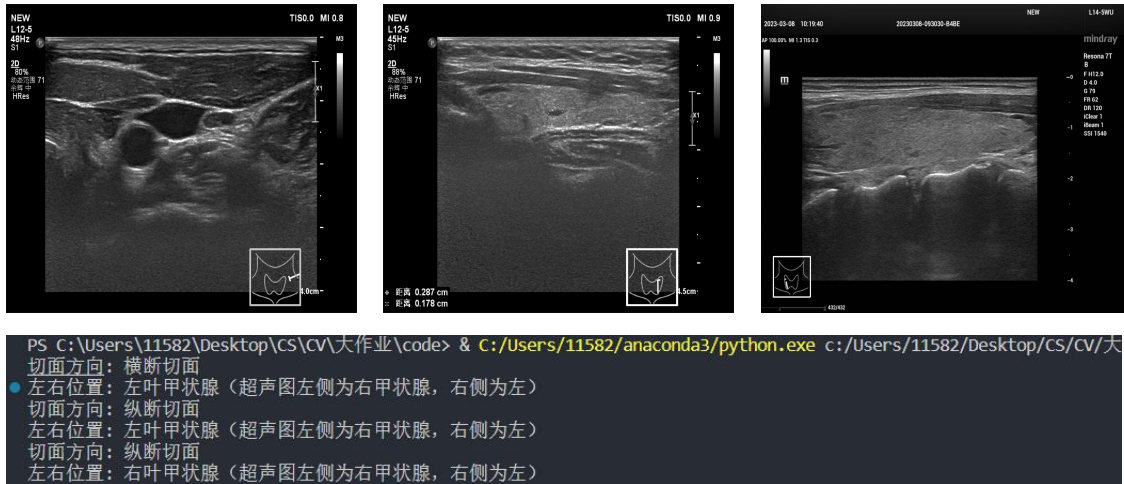
缺陷：如果是没有图标的纵切图则无法根据已有信息判断出左/右叶（原始输入信息有限，该缺陷无法优化除非提供更详细的原始信息）

2.3. 效果展示

2.3.1. 有小图标

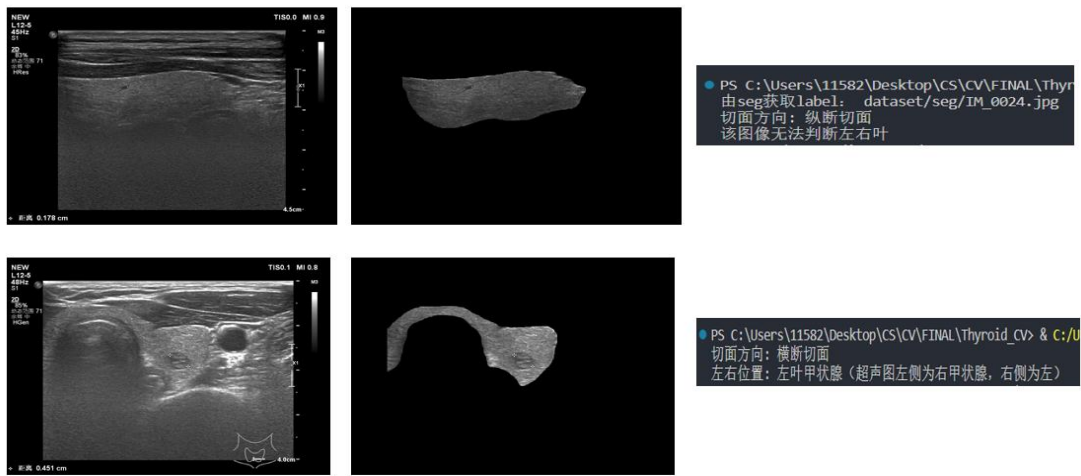


取材镜头提取



处理结果准确

2.3.2. 无小图标



处理结果准确

（下方实例原始图片中有小图标，但此步骤的前置条件是传入没有小图标的原始图像的甲状腺区域，因此只需对单看甲状腺的情况准确分类即可）

### 3. 结节位置信息

【前置条件：甲状腺 seg、ROI\_label、roi\_OCR】

#### 3.1. 获取 roi 绝对位置（高斯滤波+轮廓检测）

```
''' 输入是roi_label的位置, 输出roi中心坐标x,y,w,h '''  
def get_roi(roi_label):  
  
    roi = cv2.GaussianBlur(roi, (9, 9), 0)
```

#### 3.2. 获取 roi 相对位置

获取病灶中心位置坐标和图像处理获得的大致宽、高，输出符合 x 坐标等于固定点 x 且距离目标点位置最近的轮廓点的 y 坐标；筛选符合 x 坐标等于固定值的点的 y 坐标；若同一 x 坐标对应多个轮廓点，则获取距离结节中心点距离最近的上下两个轮廓点。

#### 3.3. 预处理：seg -> label

前置条件获得甲状腺的 seg 图，若要进行后续操作需转为黑白两色的 label 图，步骤如下：

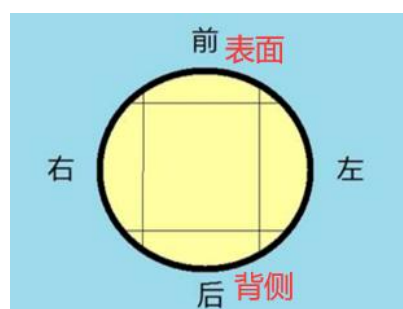
- ①二值化，非黑色区域转白色
- ②膨胀，扩展白色区域
- ③腐蚀，收缩白色区域
- ④生成遮罩 mask，应用遮罩抹去瑕疵斑点
- ⑤对于部分图片，黑色背景区域有微小的白色瑕疵，去除背景内的瑕疵只需将灰度取反再操作一遍即可

#### 3.4. roi\_w, roi\_h, have\_OCR:

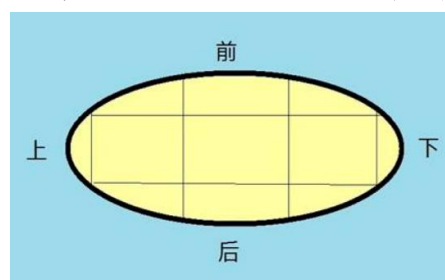
若原图有 roi 两轴大小数据，则直接使用文字识别得到的 roi 轴长；若原图没有打点，则通过轮廓检测获取 roi 边框，由宽、高得到两轴比值

#### 3.5. 根据已知规则和数据对 roi 进行位置判断：

- ①通过对前后径方向二等分确定背侧（前）/表面（后）；

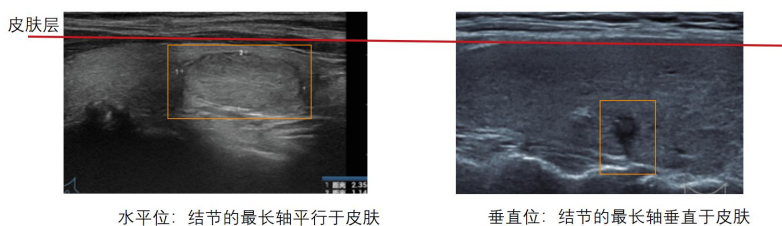


- ②【纵切】通过三等分分割纵切面定位上/中/下；



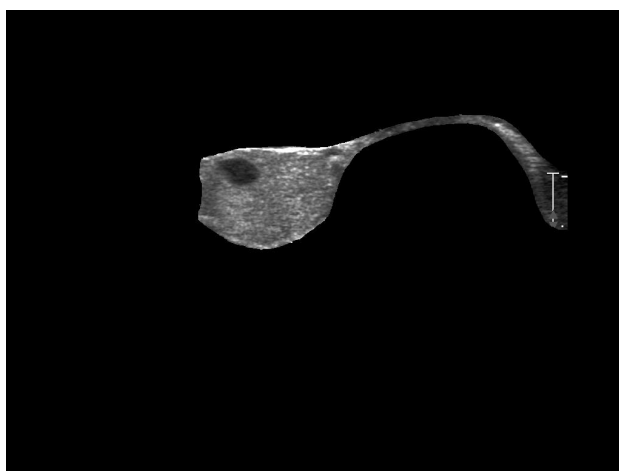


- ③【横切】通过横向二等分结合结节所在左右叶判断内/外侧；
- ④通过前置条件（上下径/前后径、左右径/前后径）确定最长轴平行/垂直皮肤；if(上述二值均<1) 垂直位；else 平行位

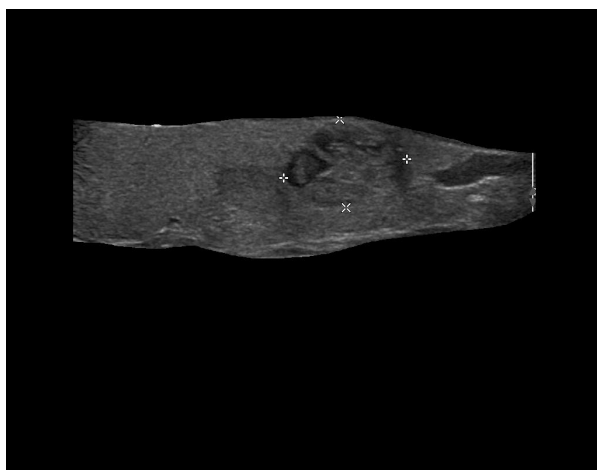


不论是横扫还是竖扫，结节的方向只和皮肤相关。垂直位的结节存在恶性的风险。

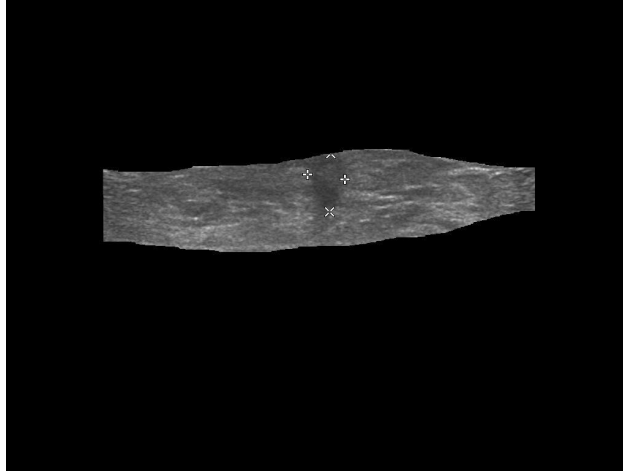
### 3.6. 效果展示：



由seg获取label: dataset/imagesTr/THYROID\_002\_0000.png  
 表面/背面: 表面  
 内侧/外侧: 外侧  
 平行位 (结节的最长轴平行于皮肤) 【无恶性风险】



由seg获取label: dataset/imagesTr/THYROID\_026\_0000.png  
 表面/背面: 表面  
 上/中/下: 中  
 平行位 (结节的最长轴平行于皮肤) 【无恶性风险】



由seg获取label: dataset/imagesTr/THYROID\_131\_0000.png  
表面/背面: 表面  
上/中/下: 中  
垂直位 (结节的最长轴垂直于皮肤) 【有恶性风险】