threshold-based.py

```
import cv2
import numpy as np
# 读取图像
image = cv2.imread('lung1.jpg', cv2.IMREAD_GRAYSCALE)
# 应用阈值 lung1用的140 lung2用的160
_, binary_image = cv2.threshold(image, 140, 255, cv2.THRESH_BINARY)
#将黑白倒置
inverted_image = cv2.bitwise_not(binary_image)

#构造掩码划分区域
mask = np.zeros_like(inverted_image, dtype=np.uint8)
# #(64,40)到(466,466)的矩形  lung1
cv2.rectangle(mask, (64, 40), (466, 466), (255, 255, 255), -1)  # -1 表示填充矩形
# (150,145)到(1200,700)的矩形 lung2
#cv2.rectangle(mask, (150, 145), (1200, 700), (255, 255, 255), -1)  # -1 表示填充矩形
# # 取反操作，将矩形以外的区域置为黑色
mask_inv = cv2.bitwise_not(mask)
# 将图像与掩膜相与，保留矩形内的区域
result = cv2.bitwise_and(inverted_image, mask)
cv2.imshow('Threshold Segmentation', result)
cv2.imwrite('threshold_based_lung1.jpg',result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

region-based.py

```python
import cv2
import numpy as np

# 读取图像
image = cv2.imread('lung2.jpg')
# 转换为灰度图
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# 使用GrabCut算法
rect = (100,100, 1200, 700) #lung1使用30 30 450 450
mask = np.zeros(image.shape[:2], np.uint8)
bgdModel = np.zeros((1, 65), np.float64)
fgdModel = np.zeros((1, 65), np.float64)
cv2.grabCut(image, mask, rect, bgdModel, fgdModel, 5, cv2.
                                    GC_INIT_WITH_RECT)

# 根据mask提取前景
result = np.where((mask == 2) | (mask == 0), 0, 1).astype('uint8') *
                                    255

# 显示结果
cv2.imshow('Region Segmentation', result)
cv2.imwrite('region_based_lung2.jpg',result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```