GrabCut

Ma Xinxian   121090408@link.cuhk.edu.cn

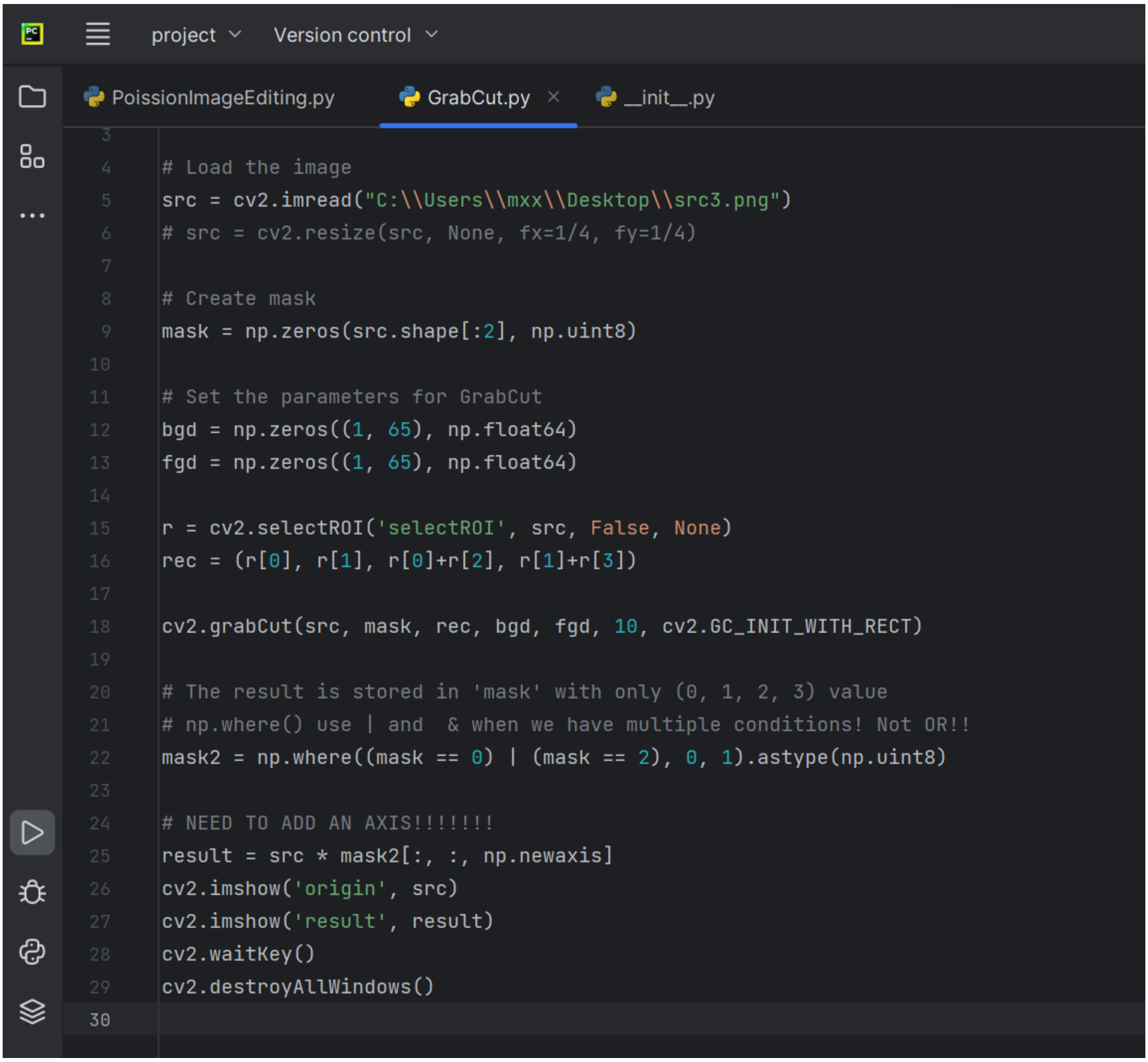
2023 年 7 月 24 日

目录

1 SIMPLE TASK

Let’s see with rectangular mode, which is easier. You definitely can set the whole image as the targeted rectangle. Since that is too easy, we omit it and try to ask user to input the targeted rectangle.

1.1 CODE WITH PYTHON

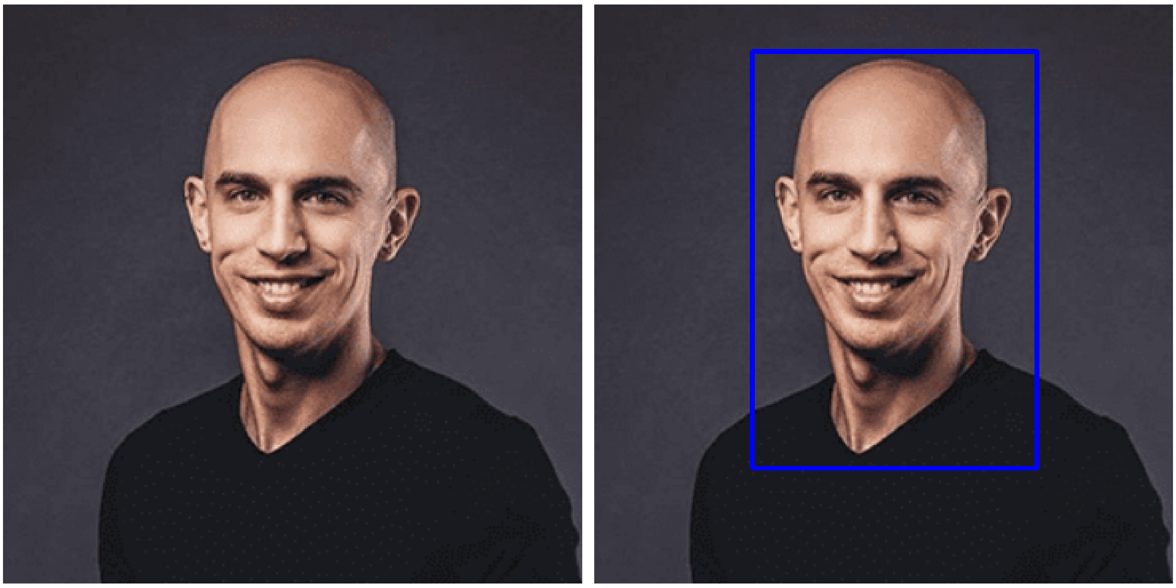


```python

# Load the image
src = cv2.imread("C:\\Users\\mxx\\Desktop\\src3.png")
# src = cv2.resize(src, None, fx=1/4, fy=1/4)

# Create mask
mask = np.zeros(src.shape[:2], np.uint8)

# Set the parameters for GrabCut
bgd = np.zeros((1, 65), np.float64)
fgd = np.zeros((1, 65), np.float64)

r = cv2.selectROI('selectROI', src, False, None)
rec = (r[0], r[1], r[0]+r[2], r[1]+r[3])

cv2.grabCut(src, mask, rec, bgd, fgd, 10, cv2.GC_INIT_WITH_RECT)

# The result is stored in 'mask' with only (0, 1, 2, 3) value
# np.where() use | and  & when we have multiple conditions! Not OR!!
mask2 = np.where((mask == 0) | (mask == 2), 0, 1).astype(np.uint8)

# NEED TO ADD AN AXIS!!!!!!!
result = src * mask2[:, :, np.newaxis]
cv2.imshow('origin', src)
cv2.imshow('result', result)
cv2.waitKey()
cv2.destroyAllWindows()

```

FIGURE 1: NORMAL-CLONE

See more information on my **GitHub.GrabCut**

1.2 RESULT



(a) source

(b) selectROI

FIGURE 2: ORIGIN

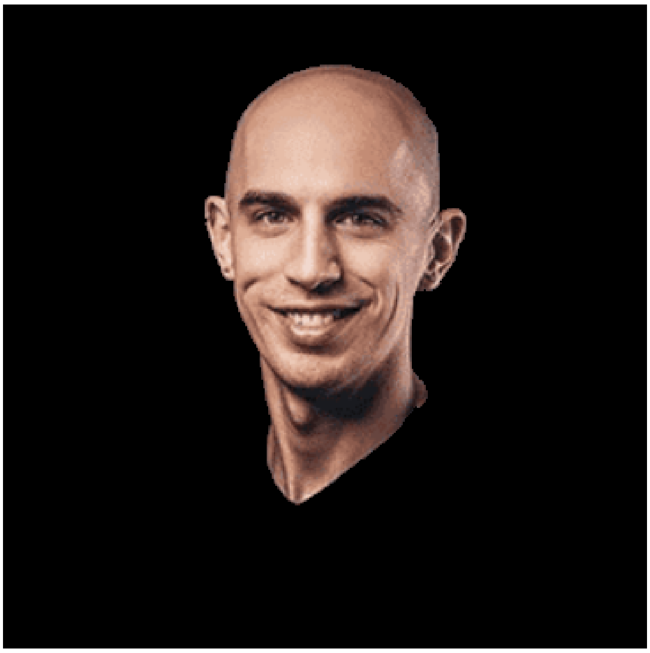
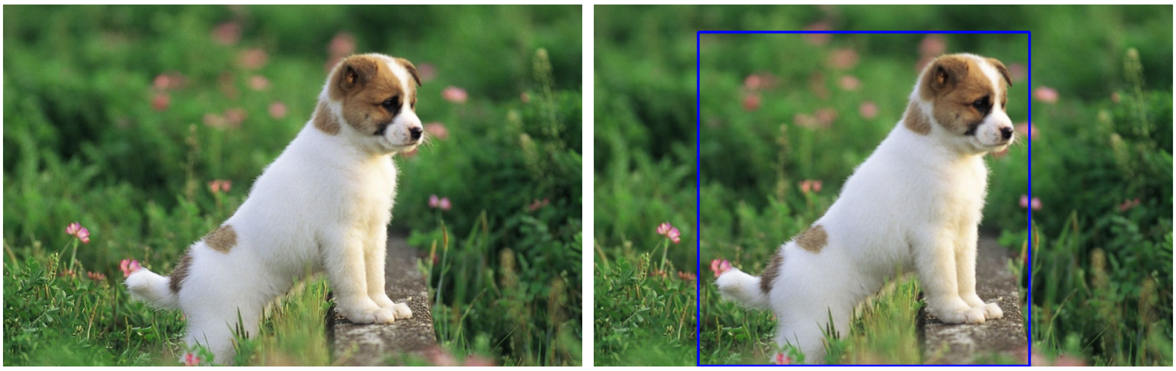


FIGURE 3: RESULT

2 MORE COMPLICATED TASK

But the previous one will meet some problems in reality, especially the background image is too noisy. The following is an example:



(a) source

(b) selectROI

FIGURE 4: ORIGIN

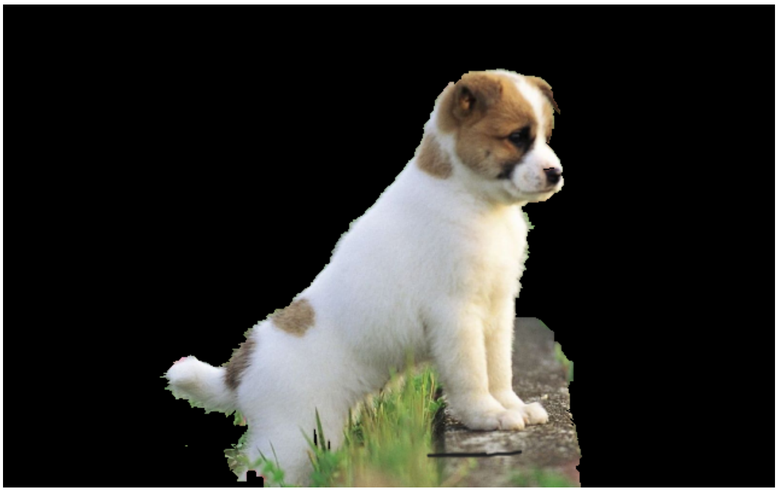
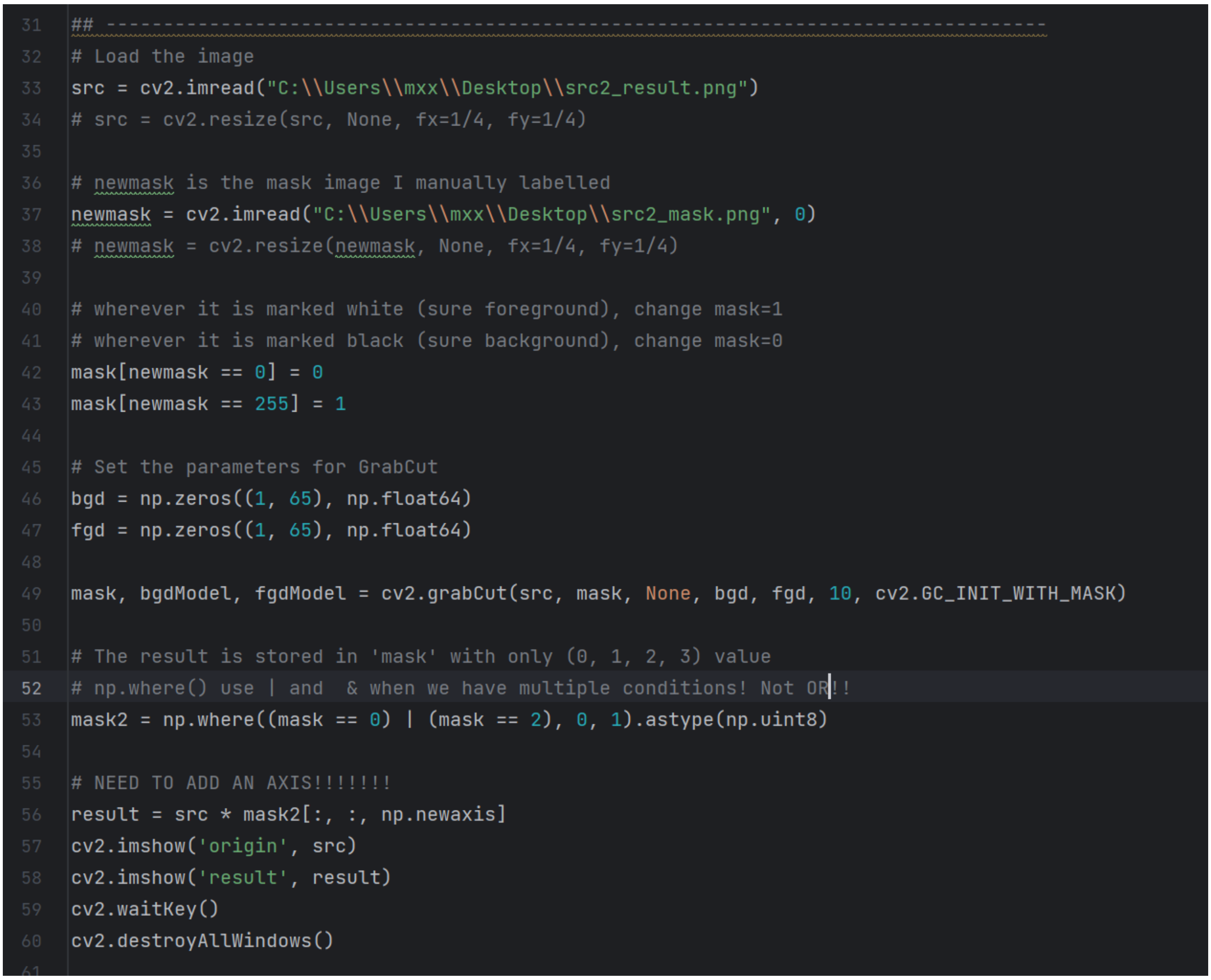


FIGURE 5: RESULT

As you see in the result, some part of ground and grass has come to picture which we don't want. We need to remove them. This is why complete GRABCUT algorithm also need an constantly correction from the users.

So we modify our resulting mask in previous case as we told now. It is called **MASK mode**. What I actually did is that, I opened input image in paint application and added another layer to the image. Using brush tool in the paint, I marked missed foreground (like hair, shoes, ball etc) with white and unwanted background (like grass, ground etc) with black on this new layer. Then filled remaining background with gray. Then loaded that mask image in OpenCV, edited original mask image we got with corresponding values in newly added mask image.

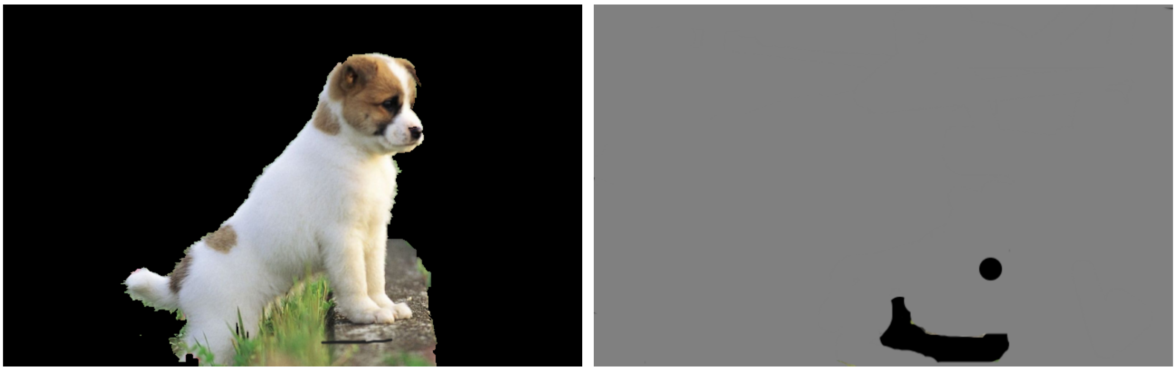
2.1 CODE WITH PYTHON



```
## ----------------------------------------------------------------------------------
# Load the image
src = cv2.imread("C:\\Users\\mxx\\Desktop\\src2_result.png")
# src = cv2.resize(src, None, fx=1/4, fy=1/4)

# newmask is the mask image I manually labelled
newmask = cv2.imread("C:\\Users\\mxx\\Desktop\\src2_mask.png", 0)
# newmask = cv2.resize(newmask, None, fx=1/4, fy=1/4)

# wherever it is marked white (sure foreground), change mask=1
# wherever it is marked black (sure background), change mask=0
mask[newmask == 0] = 0
mask[newmask == 255] = 1

# Set the parameters for GrabCut
bgd = np.zeros((1, 65), np.float64)
fgd = np.zeros((1, 65), np.float64)

mask, bgdModel, fgdModel = cv2.grabCut(src, mask, None, bgd, fgd, 10, cv2.GC_INIT_WITH_MASK)

# The result is stored in 'mask' with only (0, 1, 2, 3) value
# np.where() use | and  & when we have multiple conditions! Not OR!!
mask2 = np.where((mask == 0) | (mask == 2), 0, 1).astype(np.uint8)

# NEED TO ADD AN AXIS!!!!!!!
result = src * mask2[:, :, np.newaxis]
cv2.imshow('origin', src)
cv2.imshow('result', result)
cv2.waitKey()
cv2.destroyAllWindows()
```

FIGURE 6: NORMAL-CLONE

2.2 RESULT



(a) result from REC mode

(b) newmask

FIGURE 7: BEFORE MASK

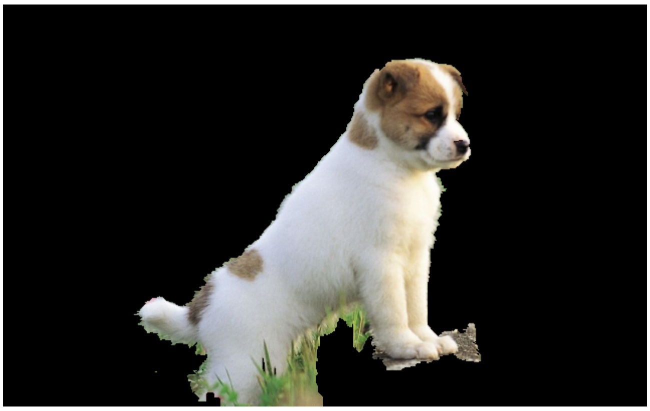
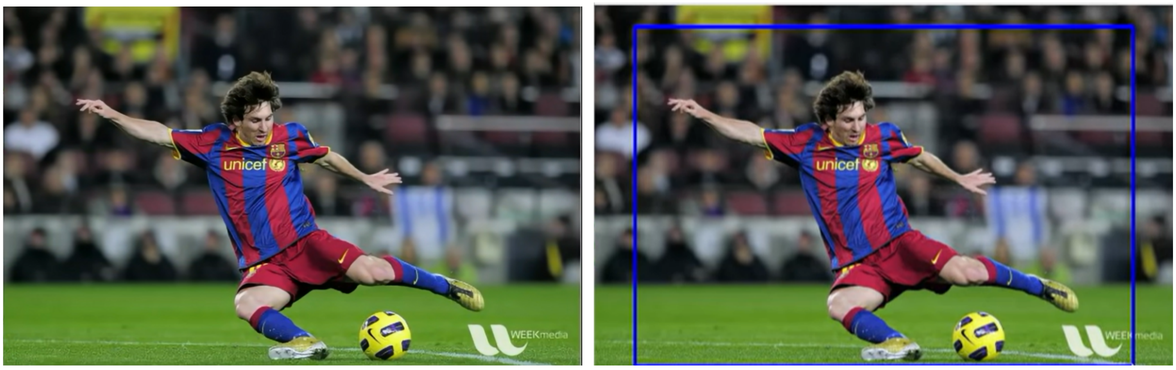


FIGURE 8: RESULT

3 OTHER EXAMPLES



(a) source

(b) ROI

FIGURE 9: ORIGIN

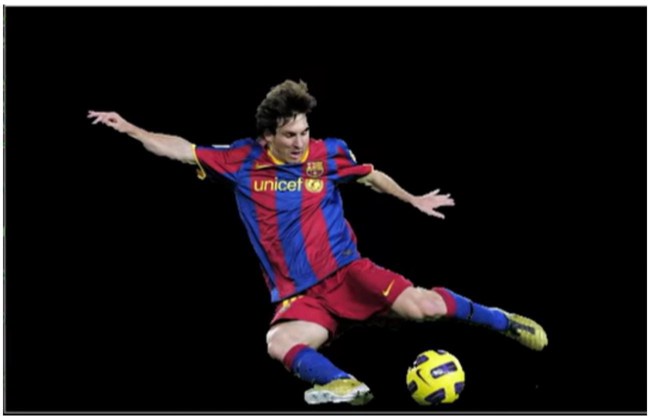


FIGURE 10: RESULT AFTER MASK AND REC

4 EXPERIENCE

尽管仍然不涉及深度学习的知识，GrabCut 算法和上次 Possion 融合一样，是很经典的方法，在深度学习广泛运用已经很好的完成了任务。

这次任务我仍然是先着手于原理的理解。第一个惊讶的点是这个算法融合了我半个大学学习的很多内容，包括机器学习当中的 GMM, 数据结构算法当中的 min-cut 还有统计学的很多知识，再次强调了一些基础学科对于研究发现的重要性。其次，我还惊叹于 GrabCut 在 GraphCut 基础上改进的巧妙：引入 GMM 替代原来灰度图的直方分布；用框选与初始化 mask 的用户交互方法，在不改变便捷度的情况下，还能够获取更多有效数据的数据；以及迭代更新以达到更好效果的算法。

实现的过程中也遇到了一些问题：包括如何进行交互？（采用 selectROI），mask 和图像大小区别带来的 bug，bgd 和 fgd 初始化设置的问题。不过其实 cv2 的 GrabCut 函数已经极大方便了这个任务的实现。当然，完整的 GrabCut 算法还需要用户继续进行标记以不断优化结果，而我这里没有（这也导致了这个算法在别的一些图上效果并不理想，展示的是较为理想的结果）。

很期待接下来的任务！