

《数字图像处理》实验报告

姓名： 汪雨卿 学号： 19120191

实验二

一. 任务 1

将 cat.jpg 中猫放置到 table.jpg 中的桌子上
要求编程实现，而不是手工处理

a) 核心代码：

1. 读取图片并获取掩膜

```
# 读取并转换图片格式
cat = cv2.imread('cat.jpg')
hsv = cv2.cvtColor(cat, cv2.COLOR_RGB2HSV)

# 指定绿色范围，60表示绿色，我取的范围是-+10
Gmin = np.array([50, 100, 100])
Gmax = np.array([70, 255, 255])

# 确定绿色范围
mask = cv2.inRange(hsv, Gmin, Gmax)
cv2.imshow('m', mask)

# 确定非绿色范围
mask_not = cv2.bitwise_not(mask)
cv2.imshow('m_n', mask_not)

# 腐蚀膨胀
# 腐蚀操作：将会腐蚀图像中白色像素，以此来消除小斑点，
# 膨胀操作：将使剩余的白色像素扩张并重新增长回去。
erode = cv2.erode(mask, None, iterations=1)
cv2.imshow('erode', erode)
dilate = cv2.dilate(erode, None, iterations=1)
cv2.imshow('dilate', dilate)
```

2. 获取前景图片，即猫的抠图

```
# 通过掩码控制的按位与运算锁定绿色区域
green = cv2.bitwise_and(cat, cat, mask=mask)
cv2.imshow('g', green)
# 通过掩码控制的按位与运算锁定非绿色区域
green_not = cv2.bitwise_and(cat, cat, mask=mask_not)
cv2.imshow('g_n', green_not)

# 拆分为3通道
b, g, r = cv2.split(green_not)

# 合成4通道
bgra = cv2.merge([b, g, r, mask_not])

# 保存带有透明通道的catnew.jpg图片,
cv2.imwrite('catnew.jpg', bgra)
```

3. 在已有的 table 图片上添加上猫的前景图

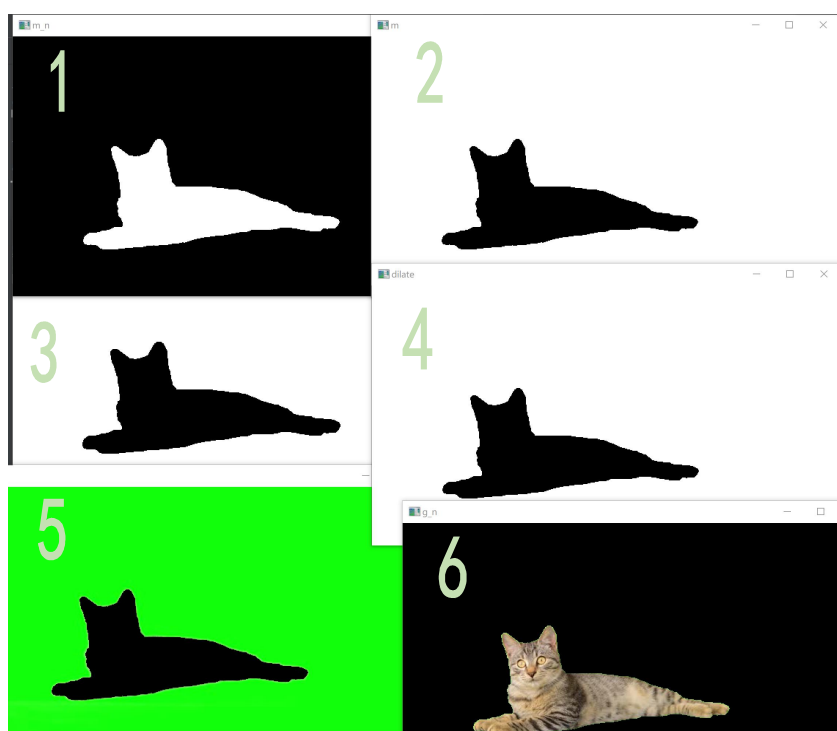
```
# 遍历替换，给图片加上背景
rows, cols, channels = bgra.shape # 1. 获取前景图片的三通道数据，之后用于图片添加
table = cv2.imread('table.jpg') # 2. 读入背景图片
center = [200, 350] # 3. 在新背景图片中的起始位置

for i in range(rows): # 4. 遍历前景图片的行列，找到黑色的点。将原图中对应位置的像素去替换黑点
    for j in range(cols):
        if dilate[i, j] == 0: # 0代表黑色的点
            table[center[0]+i, center[1]+j] = cat[i, j] # 此处替换颜色，为BGR通道
```

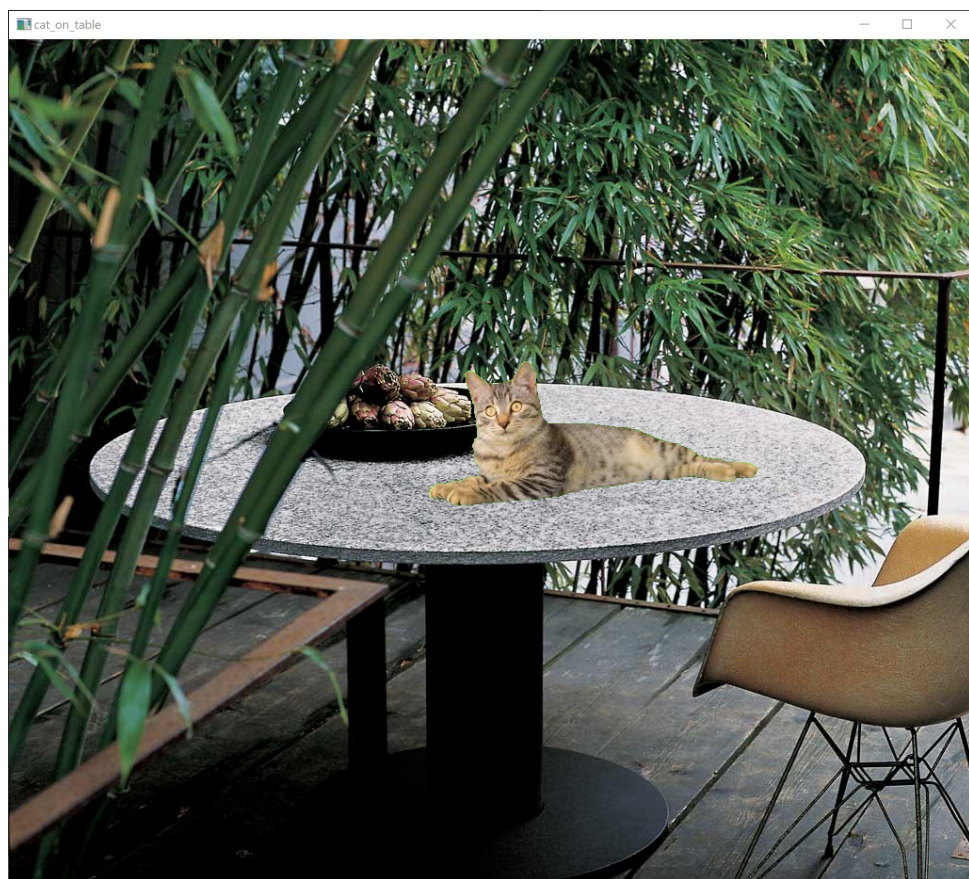
b) 实验结果截图

1. 图 1~6 依次为：

- 1: 获取到的 mask，提取绿色范围
- 2: 获取到的 mask_not，提取非绿色范围
- 3-4: 通过膨胀腐蚀优化得到的图片
- 5: 通过位与锁定绿色范围
- 6: 同 4 得到猫咪的前景图片



2. 最终实现的结果



c) 实验小结

本实验基于上次的实验基础上进行了一定的难度提升。同时结合了课堂上所涉及到的图像的算术运算,从实际操作中体验到了在电影制作过程中,消除绿幕操作的具体过程和实现逻辑。在实验的过程中,获取到图像的 **mask** 和叠加并没有想象中的困难,通过学习 **OpenCv** 手册上的案例,便能够实现在绿幕上抠出猫咪的掩膜图层。同时,也学习查找了一些对于 **mask** 性能的优化的方式,比如膨胀和腐蚀。