

《数字图像处理》实验报告

姓名： 汪雨卿 学号： 19120191

实验三

一. 任务 1

将 wukong.jpg 中的内容，通过几何变换贴到 board1.jpg 和 board2.jpg 中的某张海报墙上
自己选一张图片，重复如上操作
要求编程实现，而不是手工处理

a) 核心代码:

1. 读取 wukong.jpg 并且修改它的长和宽

```
# 对post进行大小变换;cv.INTER_LINEAR双线性插值法
height, width = post0.shape[:2]
post0 = cv.resize(post0, (740, 850), interpolation=cv.INTER_LINEAR) # 宽, 高
```

2. 对 wukong.jpg 进行放射变换，并且指定一个填充色

```
# 进行放射变换,保存至dst
rows, cols = post0.shape[:2]
pts1 = np.float32([[0, 0], [0, 850], [740, 0], [740, 850]])
pts2 = np.float32([[0, 140], [0, 800], [740, 0], [740, 850]])
M = cv.getPerspectiveTransform(pts1, pts2)
dst = cv.warpPerspective(post0, M, (cols, rows), 0.6, borderMode=cv.BORDER_CONSTANT, borderValue=(50, 50, 50, 255))
# 将边长设置成一个自定义的颜色

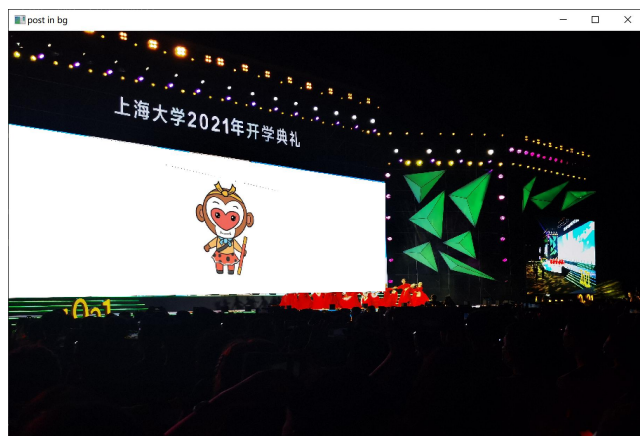
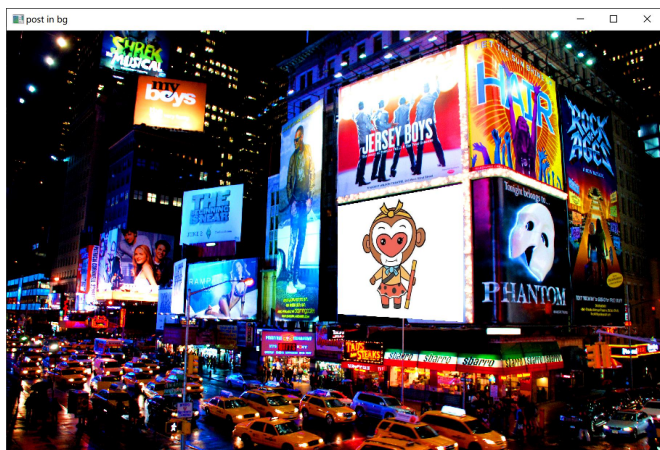
plt.figure(figsize=(10, 8), dpi=80)
```

3. 把 wukong.jpg 叠加到 board1.jpg 的指定位置上。同时，对于边缘的填充色，不予叠加。

```
# 把图片变形之后，贴到背景上
rows_b, cols_b, channels_b = dst.shape
center = [938, 1950] # 高, 横

for i in range(rows_b):
    for j in range(cols_b):
        if not all(dst[i, j] == [50, 50, 50, 255]): # 通过判断，当像素点的颜色和填充色相同的时候，不进行叠加，实现黑边的去除。
            background[center[0] + i, center[1] + j] = dst[i, j]
cv.namedWindow('post in bg', cv.WINDOW_NORMAL)
```

b) 实验结果截图



c) 实验小结

本实验中主要应用了上课所学习到的对于图像的变换操作。分析图像和海报的关系，得出利用仿射变换的方式能够最有效的实现本实验的任务。本实验最大的难点在于如何去除变形后填充的黑边。我受到实验二去除绿幕的启发，将填充的背景色更改为一个原图没有的颜色，然后在覆盖的时候，对于出现该颜色的像素点屏蔽，从而实现黑边的去除。