

《计算机视觉》实验报告

姓名：冯俊佳 学号：23122721

实验 2

一. 任务 1

a) 核心代码:

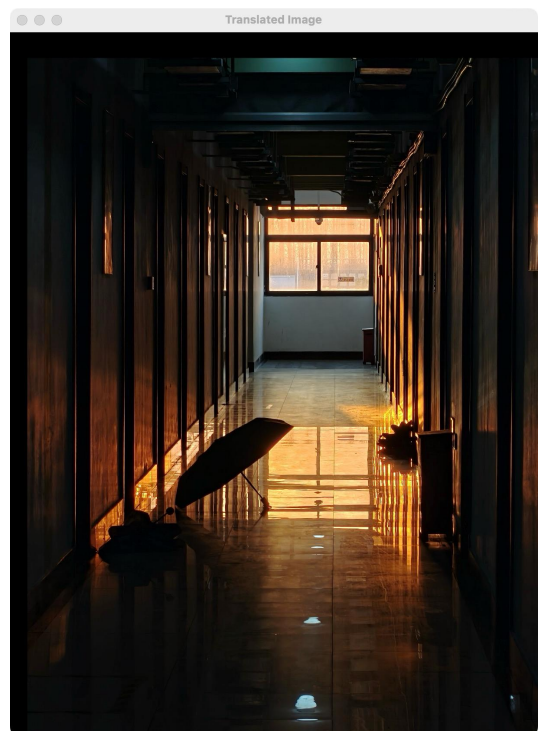
```
1.# 读取图片
2.my_photo = cv2.imread('/Users/feng/Desktop/ 计 算 机 视 觉
   /Week2/W2_Experiment/Corridor.jpg', 1)
3.# 获取高和宽 得到图片的大小
4.height, width = my_photo.shape[:2]
5.
6.# -----平移-----
7.# tx 向右 ty 向上
8.tx, ty = 100, 150
9.# 旋转矩阵构造
10.translation_matrix = np.array([
11.    [1, 0, tx],
12.    [0, 1, ty]
13.], dtype=np.float32)
14.# 将旋转矩阵应用于图像
15.translated_image = cv2.warpAffine(src=my_photo, M=translation_
   matrix, dsize=(width, height))
16.
17.# -----缩放-----
18.# 计算比例
19.scale_up_x = 1024 / width
20.scale_up_y = 768 / height
21.# 原始比例 0.6 缩小
22.scale_down = 0.6
23.scaled_f_down = cv2.resize(my_photo, None, fx=scale_down, fy=s
   cale_down, interpolation=cv2.INTER_LINEAR)
24.scaled_f_up = cv2.resize(my_photo, None, fx=scale_up_x, fy=sca
   le_up_y, interpolation=cv2.INTER_LINEAR)
25.
26.# -----翻转-----
27.img_0 = cv2.flip(my_photo, 1) # 水平翻转
```

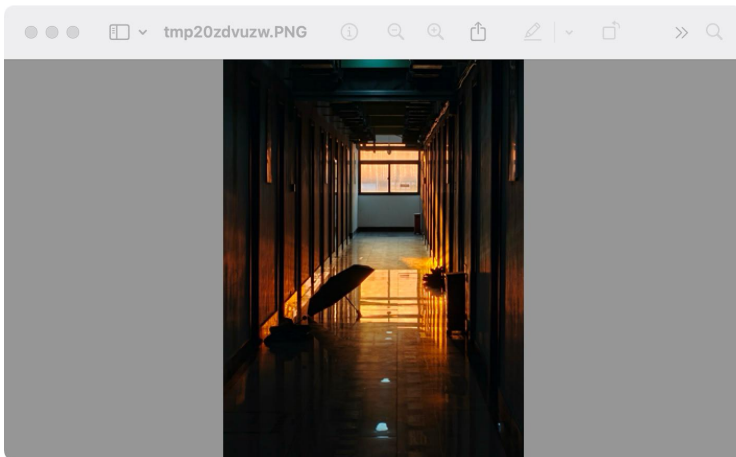
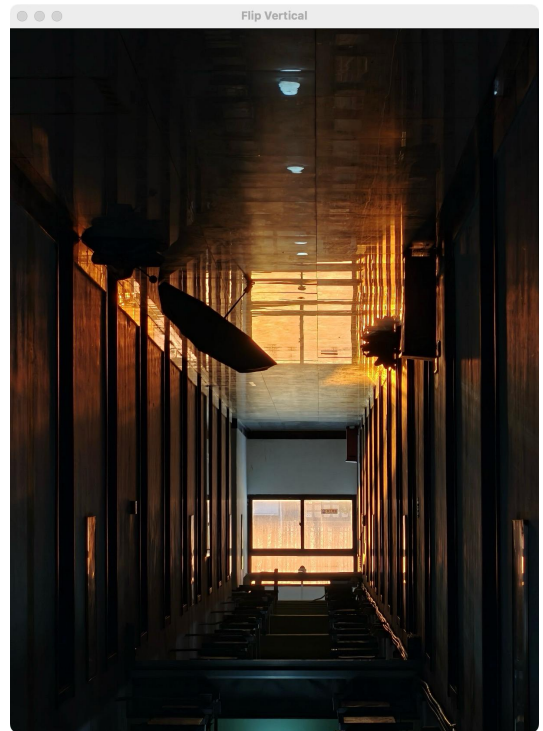
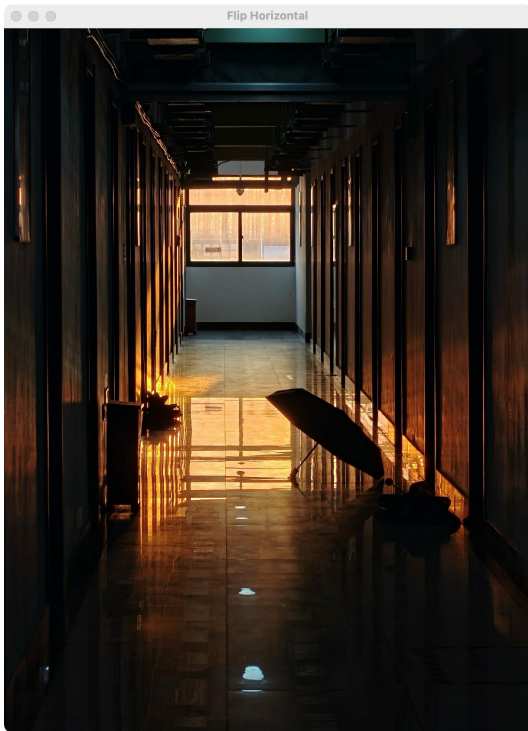
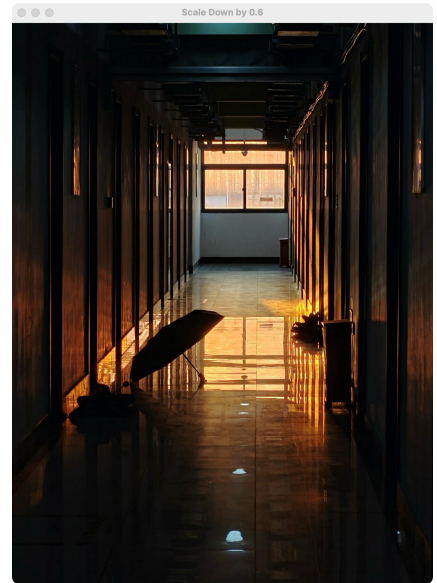
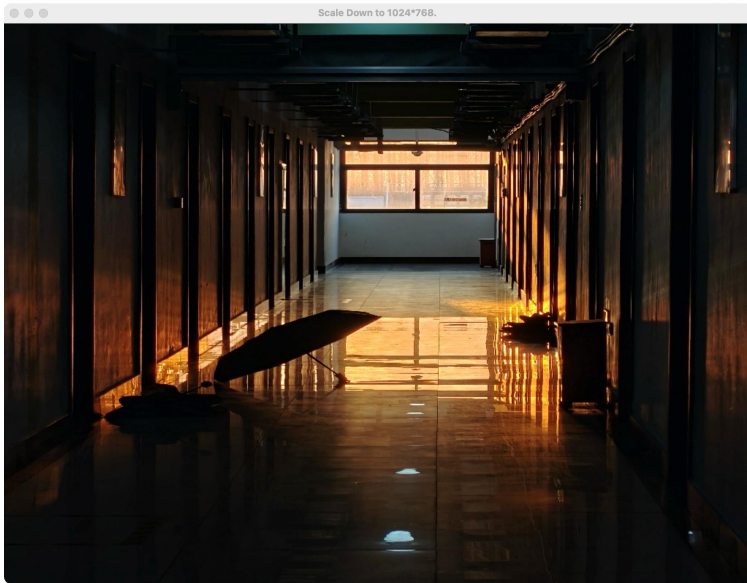
```

28. img_1 = cv2.flip(my_photo, 0) # 垂直翻转
29. img_2 = cv2.flip(my_photo, -1) # 水平+垂直翻转
30.
31. # -----旋转-----
32. # 图片中心
33. center = (width / 2, height / 2)
34. # 旋转 45 度
35. rotate_matrix = cv2.getRotationMatrix2D(center=center, angle=45, scale=1)
36. # 旋转图片
37. rotated_image = cv2.warpAffine(src=my_photo, M=rotate_matrix,
    dsize=(width, height))
38.
39. # -----缩略-----
40. # 使用 PIL
41. # 目标尺寸
42. target_width = 1024 / 2
43. target_height = 768 / 2
44. # 读取图片
45. image = Image.open('/Users/feng/Desktop/ 计 算 机 视 觉
    /Week2/W2_Experiment/Corridor.jpg')
46. # 生成缩略图
47. image.thumbnail((target_width, target_height))

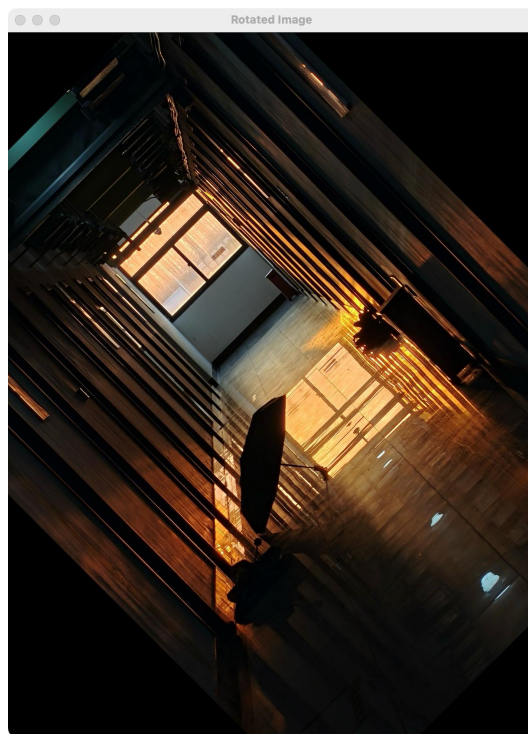
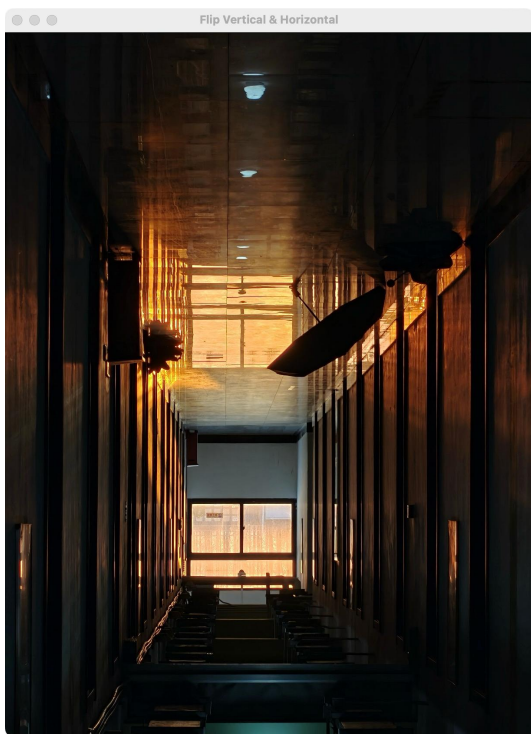
```

b) 实验结果截图





缩略图



c) 实验小结

本实验涵盖了图像变换中的一些基本操作，如平移、缩放、翻转、旋转和缩略，这些操作在图像预处理和计算机视觉中非常常见。通过这些操作，我们可以灵活地调整图像的形态，为后续的图像分析和处理打下基础。

二. 任务 2

a) 核心代码:

```
1.# 读取灰度图片
2.my_photo = cv2.imread('/Users/feng/Desktop/ 计 算 机 视 觉
   /Week2/W2_Experiment/Leopard.png', 0)
3.# 获取高和宽得到图片的大小
4.height, width = my_photo.shape[:2]
5.# -----缩放正方形-----
6.# 计算比例
7.scale_up_x = 600/width
8.scale_up_y = 600/height
9.image = cv2.resize(my_photo, None, fx=scale_up_x, fy=scale_up_y,
   interpolation=cv2.INTER_LINEAR)
10.# -----切片-----
11.# 修改长宽
12.width = 600
13.height = 600
14.# 创建一个与图像大小相同的黑色背景
```

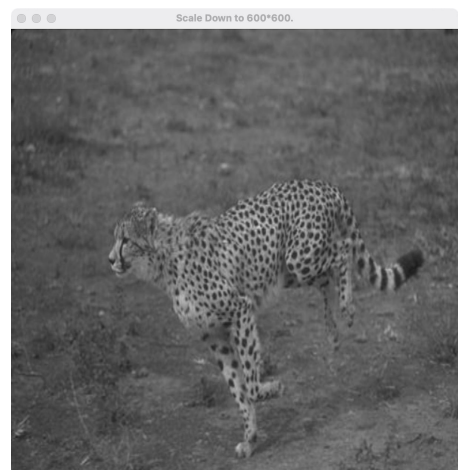


```

15.mask = np.zeros((height, width), dtype=np.uint8)
16. # 定义圆心坐标和半径 (这里假设圆心在图像的中心)
17.center = (width // 2, height // 2)
18.radius = min(width, height) // 2
19. # 在掩膜上绘制白色圆形
20.cv2.circle(mask, center, radius, 255, -1)
21. # 对图像应用掩膜
22.masked_image = cv2.bitwise_and(image, image, mask=mask)
23. # 显示原始图像和掩膜图像
24.cv2.imshow('Original Image', image)
25.cv2.imshow('Masked Image', masked_image)
26.cv2.waitKey(0)
27.cv2.destroyAllWindows()
28. # 保存图片
29.cv2.imwrite('Masked_Image.jpg', masked_image)

```

b) 实验结果截图



```

v Week2
v W2_Experiment
  Corridor.jpg
  Leopard.png
  Masked_Image.jpg
  T1.py
  T2.py
  23122721_冯俊佳_实验2.doc

```

c) 实验小结

本次实验主要对灰度图像进行了缩放和掩膜处理，展示了如何对灰度图像进行缩放操作并使用掩膜进行区域切割。这些操作常用于图像预处理和特定区域的提取，能够为进一步的图像分析提供必要的基础处理。