

《计算机视觉》实验报告

姓名：刘远航 学号：22121883

实验二

一. 任务 1

a) 实验内容

1. 平移：x 轴平移 100 像素，y 轴平移 150 像素
2. 缩放：缩放到 1024*768；按比例缩小（60%）
3. 翻转：水平翻转，垂直翻转，水平+垂直翻转
4. 旋转：给出旋转中心，旋转角度，对图片旋转
5. 缩略：将图片缩小，放到原图的左上角

b) 核心代码：

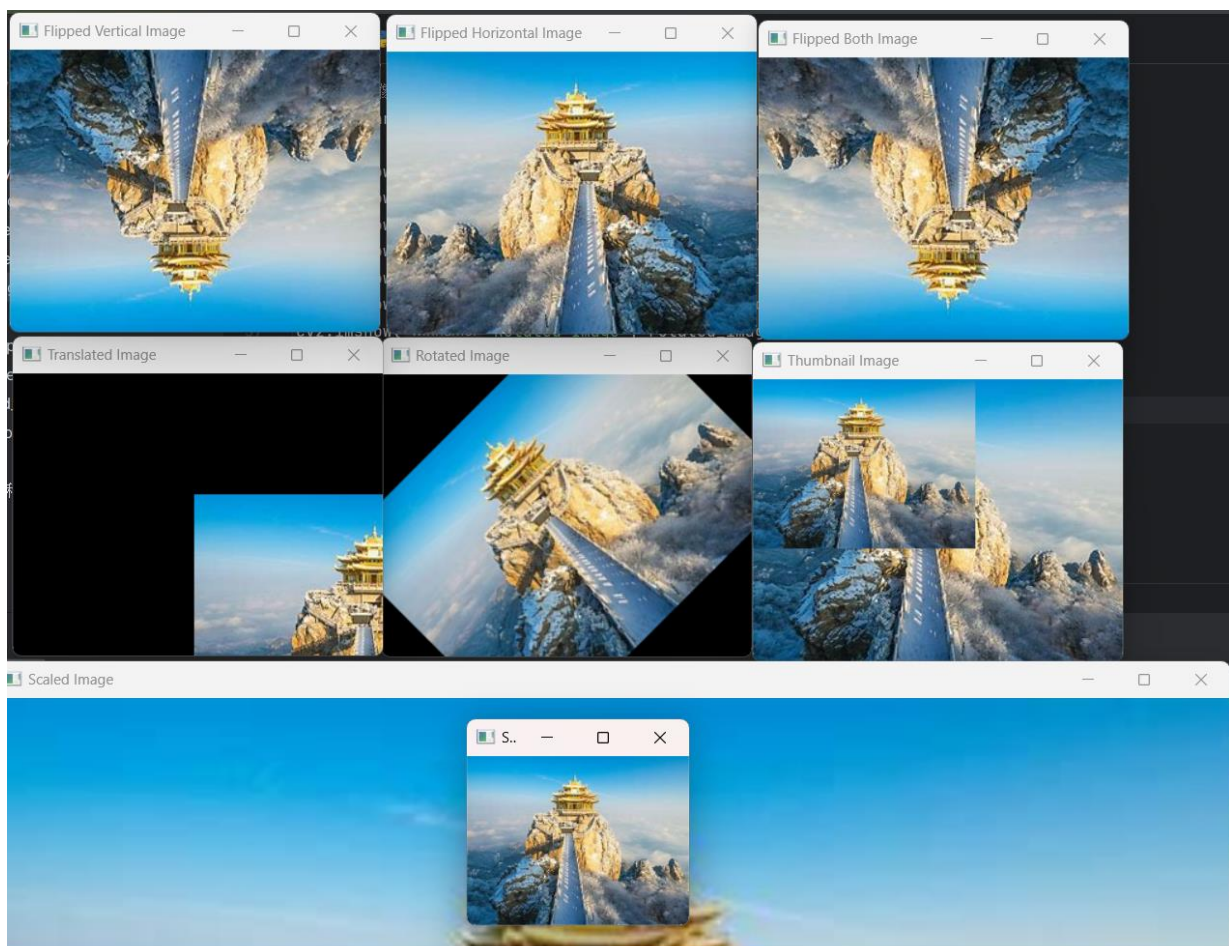
```
# 平移
trans_x = 150
trans_y = 100
rows, cols = image.shape[:2]
M = np.float32([[1, 0, trans_x], [0, 1, trans_y]])
translated_image = cv2.warpAffine(image, M, (cols, rows))
# 缩放
scaled_image = cv2.resize(image, (1024, 768))
scaled_down_image = cv2.resize(image, None, fx=0.6, fy=0.6)
# 翻转
flipped_horizontal_image = cv2.flip(image, 1)
flipped_vertical_image = cv2.flip(image, 0)
flipped_both_image = cv2.flip(image, -1)
# 旋转
rotation_center = (cols // 2, rows // 2)
rotation_angle = 45 # 旋转角度（度）
M = cv2.getRotationMatrix2D(rotation_center, rotation_angle, 1)
rotated_image = cv2.warpAffine(image, M, (cols, rows))
# 缩略
scaledimage = cv2.resize(image, None, fx=0.6, fy=0.6)
```

```

# 获取原始图像和缩小后图像的尺寸
original_height, original_width = image.shape[:2]
scaled_height, scaled_width = scaledimage.shape[:2]
# 将缩小后的图像放置在原始图像的左上角
start_x = 0
start_y = 0
end_x = scaled_width
end_y = scaled_height
# 确保放置的区域不超出原始图像的范围
if end_x > original_width:
    end_x = original_width
if end_y > original_height:
    end_y = original_height
# 在原始图像上放置缩小后的图像
image[start_y:end_y, start_x:end_x] = scaledimage[:end_y-start_y, :end_x-start_x]

```

c) 实验结果截图



d) 实验小结

任务一主要是用 `opencv` 的各种函数来对图片进行平移，缩放，旋转，缩略等操作，通过调节一个个参数可以对图片进行调整，对计算机视觉有了进一步了解。

二. 任务 2

a) 实验内容

- 1、读取一张新的图片，将其转换为灰度图片；
- 2、调整灰度图片为正方形（边长不小于 500 像素）；
- 3、用圆形掩膜对图片进行切片，并保存切片后的图像（如图所示）

b) 核心代码

```
# 读取图片并转换为灰度图像
image = cv2.imread("image2.jpg")
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
rows, cols = image.shape[:2]
print(rows, cols)
gray_image = cv2.resize(gray_image, (500, 500))
# 创建圆形掩膜
size=500
mask = np.zeros((size, size), dtype=np.uint8)
center = (size // 2, size // 2)
radius = size // 2
cv2.circle(mask, center, radius, (255, 255, 255), -1)

# 对图像进行掩膜切片
masked_image = cv2.bitwise_and(gray_image, gray_image, mask=mask)
# 显示和保存切片后的图像
cv2.imshow("Masked Image", masked_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
cv2.imwrite("masked_image.jpg", ma_image)
```

c) 实验结果截图



d) 实验小结

通过这个实验学习了一些函数的用法，了解了一些关于掩膜，切片的知识并用python 进行实现，提高了我对这门课的兴趣。