工具：Jupyter，内核为Python3.12，opencv版本：4.8.1

题目：使用openCV读取一张图片，并对图片做以下操作：

（1）将图片的缩小四分之一

（2）在（1）的基础上将图片旋转135度

（3）将图片向左平移四分之三，向上平移五分之四

这次我们分别封装3个函数，来分别对应三个操作

首先导入我们的opencv库和Numpy函数库，并且导入咱们预先准备好的图片

1. **import** cv2
2. **import** numpy as np
3. img=cv2.imread("D:/1699194637144535.jpg")

定义一个CvShow函数，内部先imshow一下，然后等待操作并且最后摧毁所有视窗

1. **def** CvShow(name,img):
2. cv2.imshow(name,img)
3. cv2.waitKey(0)
4. cv2.destroyAllWindows()

定义一个ImageZoomOut函数，内部就使用一个resize函数，绝对尺寸设为None，表示咱们用后面的比例。题目要求缩小四分之一，即为原来的0.75，最后我们return一下处理好的图片

1. **def** ImageZoomOut(img):
2. ret=cv2.resize(img,None,fx=0.75,fy=0.75)
3. **return** ret

定义一个PictureRotation函数，内部我们先获取下图片的高和宽

1. **def** PictureRotation(img):
2. h,w=img.shape[:2]

接着我们用getRotationMatrix2D函数去获得一个旋转矩阵，旋转中心设在图片中央（高一半，宽一半），题目要求的135°，我们就写135，然后缩放比例为1

1. M=cv2.getRotationMatrix2D((h/2,w/2),135,1)

最后我们只要把图像，旋转矩阵放入warpAffine函数里就行了，图片尺寸我们保持不变

1. ret=cv2.warpAffine(img,M,(h,w))
2. **return** ret

完整的函数为

1. **def** PictureRotation(img):
2. h,w=img.shape[:2]
3. M=cv2.getRotationMatrix2D((h/2,w/2),135,1)
4. ret=cv2.warpAffine(img,M,(h,w))
5. **return** ret

针对题目的最后一个要求，我们针对性地再定义一个函数PicturePan，内部我们先获取一下高和宽

1. **def** PicturePan(img):
2. h,w=img.shape[:2]

我们接着定义一个np.float32 类型的二维数组M，一个平移变化矩阵。题目要求左移四分之三，向上平移五分之四，我们就左移0.75倍的宽，上移0.8倍的高。

1. M=np.float32([[1,0,-0.75\*w],[0,1,-0.8\*h]])

这里我们用来平移的函数依然是warpAffine，我们将一点不变地将这些参数写进去，最后return

1. ret=cv2.warpAffine(img,M,(h,w))
2. **return** ret

完整的函数为：

1. **def** PicturePan(img):
2. h,w=img.shape[:2]
3. M=np.float32([[1,0,-0.75\*w],[0,1,-0.8\*h]])
4. ret=cv2.warpAffine(img,M,(h,w))
5. **return** ret

在最后，我们只需要轻松地将这些函数代入即可

1. CvShow("img1",ImageZoomOut(img))
2. CvShow("img2",PictureRotation(ImageZoomOut(img)))
3. CvShow("img3",PicturePan(img))

完整的代码为：

1. **import** cv2
2. **import** numpy as np
3. img=cv2.imread("D:/1699194637144535.jpg")
4. **def** CvShow(name,img):
5. cv2.imshow(name,img)
6. cv2.waitKey(0)
7. cv2.destroyAllWindows()
8. **def** ImageZoomOut(img):
9. ret=cv2.resize(img,None,fx=0.75,fy=0.75)
10. **return** ret
11. **def** PictureRotation(img):
12. h,w=img.shape[:2]
13. M=cv2.getRotationMatrix2D((h/2,w/2),135,1)
14. ret=cv2.warpAffine(img,M,(h,w))
15. **return** ret
16. **def** PicturePan(img):
17. h,w=img.shape[:2]
18. M=np.float32([[1,0,-0.75\*w],[0,1,-0.8\*h]])
19. ret=cv2.warpAffine(img,M,(h,w))
20. **return** ret
21. CvShow("img1",ImageZoomOut(img))
22. CvShow("img2",PictureRotation(ImageZoomOut(img)))
23. CvShow("img3",PicturePan(img))