**大作业：用列表实现二维卷积**

**一、二维卷积**

用一个*h×w*大小的卷积核矩阵***K***和一个*H×W*大小的图像矩阵***im***进行二维卷积操作包括如下步骤：

1. **翻转核矩阵**

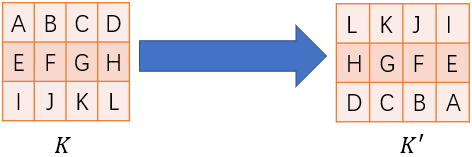
输入: 原始矩阵K

输出：翻转后的矩阵K′

方法：横向翻转 + 纵向翻转

K′[i][j]=K[h−i−1][w−j−1]

其中，h为K矩阵的行数（列表长度）；w为K矩阵的列数(列表中每个元素的长度)



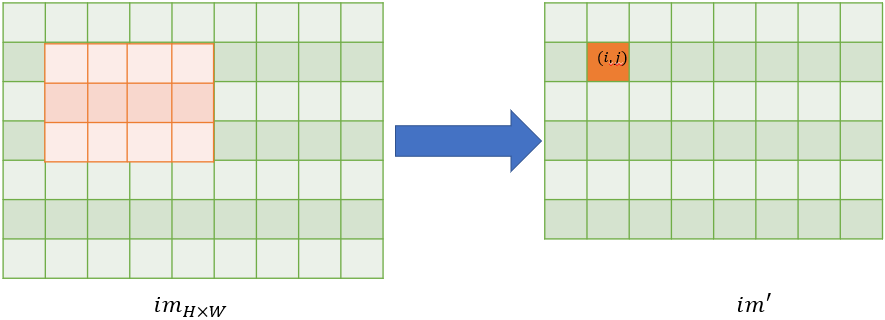
2. **按指定步长在不同位置计算响应值**

输入：图像矩阵𝑖𝑚，翻转后的核矩阵K′，步长stride

输出：卷积后的图像矩阵𝑖𝑚’

方法：

用翻转后的卷积核K′，从左上角开始，在图像上从左到右、从上到下滑动：每次向右滑动移动的步长由stride指定；当滑动后的卷积核超出图像区域结束该行滑动，向下移动步长stride，再次从左向右滑动；直到滑动到右下角。在每个位置（卷积核左上角对齐到图像（i，j）位置处），将K′和其覆盖的im的对应位置的元素相乘并求和，作为这个位置的响应值。所有位置的响应值按位置顺序组成的矩阵𝑖𝑚’（列表的列表）。



**二、作业**

实现图片二维卷积函数conv2d(im, K, stride=1)

输入：图像转换成的由list表示的二维数组im；另一个由list表示的二维数组K；步长stride；

输出：卷积后的列表表示的二维数组

说明：

1. 下载Jupyter Notebook代码 “第3周大作业1\_题目.ipynb”和图片“ruc.jpg”

2. 阅读已有的代码（已有代码负责读入图片、转换为二维数值列表以及将二维数值列表转换为图片进行输出）

3. 编写代码，实现conv2d(im, K, stride)函数，其中stride步长，默认为1

* 完成K矩阵的翻转操作
* 完成对im矩阵的卷积操作，返回卷积后的二维数值列表

4. 观察在不同卷积核下输出图片的差异

可以用下述测试用例检验conv2d函数是否正确：

例1：输入im = [[1, 1, 0, 0],

[1, 1, 0, 0],

[0, 0, 1, 1],

[0, 0, 1, 1]]

K = [[1, 1],

[1, 1]]

stride = 2

输出：

[[4, 0],

[0, 4]]

例2：输入

im = [[ 8, -1, -8, 2, 3, -9, -2, 5, 5, 5, -8],

[ 8, 2, -8, -4, 1, 7, 6, -7, 8, -4, -6],

[ 5, 7, -1, 2, 1, -7, -4, 1, -1, 1, 0],

[-3, 0, -2, -2, -5, -4, -7, 5, 0, -3, -1],

[-5, 0, 3, 1, -1, 4, 8, -2, -3, -8, 5],

[-2, -7, -6, -3, -3, -3, 2, 5, -7, 7, 3],

[ 8, -9, -3, 3, 0, -4, -6, -8, 6, -8, -7],

[-6, 0, 0, 8, 3, -6, 1, 8, -2, 2, 7],

[-4, -8, 6, -3, -9, 2, -5, -4, -9, 0, -5],

[-6, 4, -1, 0, -4, 7, 4, 5, 0, -4, -6],

[ 6, 4, 2, -4, 7, 4, 8, -5, -1, -7, -5]]

K = [[ 5, -3, 3, 5, 7],

[ 0, -9, 8, -8, -8],

[ 6, 6, 1, -5, 3],

[-3, -9, 5, 0, 1],

[ 1, -9, -6, 0, 8]]

stride =1

输出：

[[87, -26, -86, 134, 99, -71, 118],

[57, -80, -238, -198, 24, 11, -46],

[81, 97, 201, 162, -73, -147, -249],

[-102, 130, 77, 164, 35, -69, 356],

[-90, 27, -99, -430, -197, 251, -221],

[184, 137, -151, 270, 68, 64, 165],

[53, -125, 34, 45, 2, -54, -113]]

Stride=2

[[87, -86, 99, 118], [81, 201, -73, -249], [-90, -99, -197, -221], [53, 34, 2, -113]]