

说明文档：

1. Seam Carving

1. 使用说明：

编程语言：python

需要安装的库：numpy、scikit-image、matplotlib

将代码放至一个具有上述包和python编译器的环境中即可运行

2. 寻找能量最小路径的算法主要代码：

#动态规划实现，方法一：循环嵌套

```
for row in range(1, H):
    for j in range(0, W):
        if j == 0:
            cost[row][j] = min(cost[row-1][j], cost[row-1][j+1]) + energy[row][j]
            paths[row][j] = np.argmin([cost[row-1][j], cost[row-1][j+1]])
        elif j == W-1:
            cost[row][j] = min(cost[row-1][j-1], cost[row-1][j]) + energy[row][j]
            paths[row][j] = np.argmin([cost[row-1][j-1], cost[row-1][j]]) - 1
        else:
            cost[row][j] = min(cost[row-1][j-1], cost[row-1][j], cost[row-1][j+1]) + energy[row][j]
            paths[row][j] = np.argmin([cost[row-1][j-1], cost[row-1][j], cost[row-1][j+1]]) - 1
```

3. 算法时间复杂度和空间复杂度

时间复杂度： $O(HW^2)$ ，要将图片横向压缩一半，则需要删去 $W/2$ 个seam，每删一个都要执行上面的循环计算查找最小能量路径，上面的循环时间复杂度是 $H*W$ ，因而总时间复杂度是 $H*W*W$

空间复杂度： $O(HW)$ ，用到两个 $H*W$ 大小的数组分别存放每个节点累计cost值，和每个节点最小能量路径索引。

4. 缩放前和缩放后的图像

