

机器视觉技术 第七次实验

实验题目——OTSU 阈值检测

实验目的：

找到区分前景和背景的最佳阈值。

实验原理：

针对不同阈值，获得该阈值下像素点是前景或背景的概率，并由此计算背景和前景的类间方差，该方差最大时，获得的阈值就是目标阈值。

实验步骤：

1. 获得图像灰度图的归一化直方图 H ，并得到 t 遍历的上下限 \max_index 和 \min_index 。
2. 按阈值 t 计算图像是前景 $[t+1, \dots, G-1]F$ 、背景 $[0, 1, \dots, t]B$ 的概率 w_F 和 w_B ，即对应灰度值概率求和。
3. $u(i)$ 为 \min_index 到 t 的 i 与 $H(i)$ 的积的和，得到 $u_F = u(t) / w_F$ 和 $u_B = (u(\max_index) - u(t)) / w_B$ 。
4. 计算类间方差 $v = w_B * (u_B - u(\max_index))^2 + w_F * (u_F - u(\max_index))^2$ 。
5. t 遍历 \min_index 到 \max_index ，获得 v 最大时的 t_{best} 。
6. 将 t_{best} 划分的前景和背景赋值。

程序代码：

main.m

```
clc;clear all;close all;
```

```
% 输入图像并转换为灰度图
```

```
img=imread('testimg.jpg');
```

```
img=rgb2gray(img);
```

```
% OTSU 阈值检测
```

```
OTSU_result = MyOTSU(img);
```

```
figure;
```

```
imshow(OTSU_result);
```

```
title('OTSU_result');
```

MyOTSU.m

```
% 定义函数 MyOTSU，参数为二值图像 img
```

```
function OTSU_result=MyOTSU(img)
```

```
% 归一化直方图
```

```
counts = imhist(img);
```

```
num = sum(counts);
```

```

probabilities = counts / num;

index = find(probabilities ~= 0);
min_index = min(index);
max_index = max(index);

% 遍历
maxv = -inf;
tbest = 0;
u = zeros(1,max_index - min_index);
u(min_index) = min_index * probabilities(min_index);
for i = min_index + 1 : max_index
    u(i) = u(i-1) + i * probabilities(i);
end
for t = min_index : max_index
    wB = sum(probabilities(min_index : t));
    wF = 1 - wB;
    uB = u(t) / wB;
    uF = (u(max_index) - u(t)) / wF;
    v = wB * (uB - u(max_index)).^2 + wF * (uF - u(max_index)).^2;
    if v > maxv
        maxv = v;
        tbest = t;
    end
end

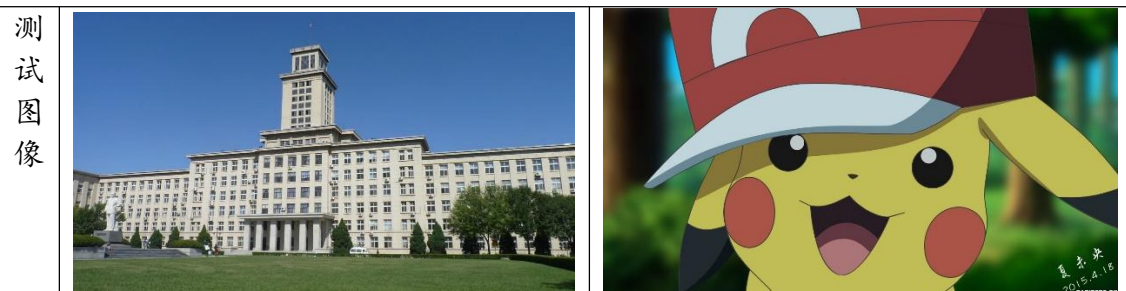
img(img > tbest) = 255;
img(img < tbest+1) = 0;

OTSU_result=img;

end

```

实验结果显示：



成果图像		
阈值t	149	123

实验分析总结：

本算法能自主找到最优的阈值，相对方便，效果也不错。针对不同的测试图像，算法都能找到不同的最合适的阈值。但这在处理前景中的阴影时，会将其判断为背景，如测试案例中教学楼的背阳面和皮卡丘帽子阴影下。

暂未解决的疑问：

在编写代码时，对平方运算的符号的使用产生了疑问。本题中“ $v = wB * (uB - u(\max_index)).^2 + wF * (uF - u(\max_index)).^2;$ ”一句，对两个分别进行了平方，开始时使用的是没有“.”的“^”，会报错，在加上“.”后不再报错。使用文心一言得到如下解释：

.^（点乘方）：元素对元素的乘方操作，对数组或矩阵的每个元素独立地进行乘方操作。如果A和B是两个相同大小的数组或矩阵，那么A.^B的结果是一个与A和B大小相同的数组，其中每个元素是A中对应元素与B中对应元素的乘方结果。

^（乘方）：矩阵乘方操作。用于两个标量（即单个数值）时，它执行普通的乘方操作。用于矩阵时，它执行矩阵的乘方，这通常涉及到特征值和特征向量的计算，不是所有的矩阵都可以被乘方。

然而此时平方的应是两个标量，为什么必须使用“.”^？