



信息论课程实践

——基于最大熵原理的图像二值分割算法实现



2015-12-2

姓名：莫帮杰（2013301200227）

一、 实践背景

在当今社会，电子化、信息化渗透在人类生产生活的各个方面。随着人类文明的快速发展，人们对信息的传输、储存、和交流的有效性和可靠性的要求也越来越高。在特定环境下，如何改进系统设计，提高这些指标，已经成为了人们不得不去思考的问题。而信息论则作为一门新兴的理论，则开宗明义的解答了这些问题，为人们提供了一套科学有效的理论依据。

信息论的其中一个应用领域是数字图像处理。数字图像处理中的一个很重要的研究方向是图像的二值分割。图像的二值分割在计算机图像处理技术中的应用非常广泛，可以应用于目标识别、图像分析、文本增强、字符识别、牌照识别等各种应用领域，因此对其进行深入研究具有很重要的实际意义。 图像的二值分割就是把一幅图像分成目标和背景两大类，传统的二值分割方法主要是根据目标和背景的灰度分布差异选取适当的阈值进行分割，因此如何来选取阈值是图像分割的关键问题。本文结合信息论中的基本知识，通过运用信息论中的方法，就图像领域的二值分割问题，来一瞥信息论的惊鸿之美。

二、 理论回顾

1. 信息量的定义

在信息论中，信源中某一符号 s 所包含的信息量 I 是由香农信息定义式给出的：

$$I = \log \frac{1}{P_s}$$

其中 P_s 为该符号出现的概率， \log 的底数通常取 2，e，或 10，对应的信息量的单位分别为比特（bit）、奈特（nat）和哈特（hart）。

2. 信息熵的定义

在信息论中，信源空间中平均每一符号所包含的信息量即为信源的熵（香农熵）。注意这里的平均是指依概率平均。由此定义可得熵的定义式：

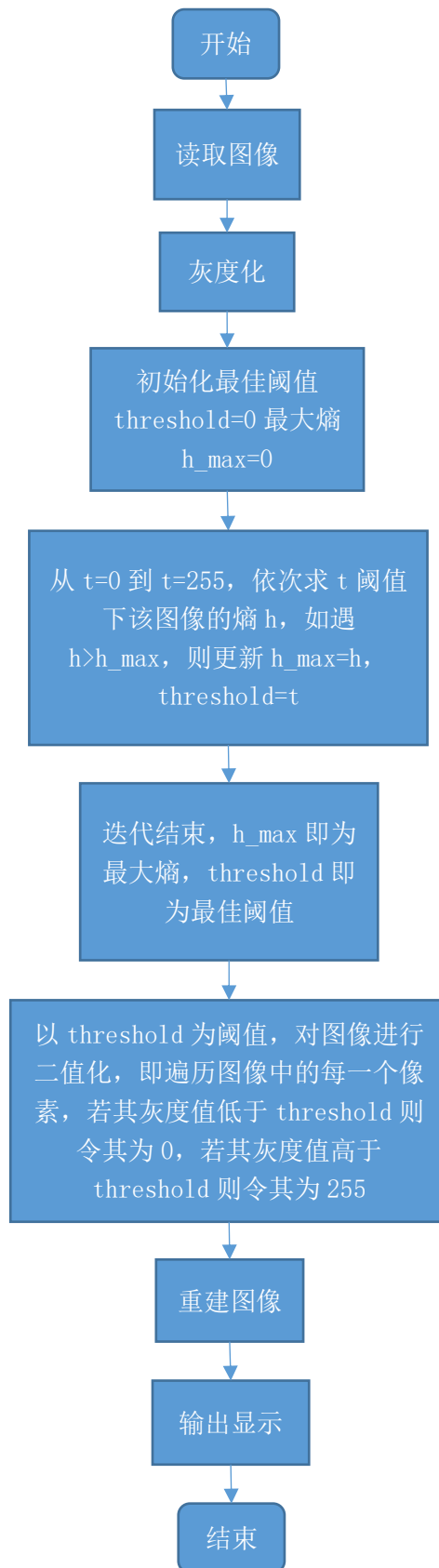
$$H(X) = \sum_{i=1}^N P_i * \log \frac{1}{P_i}$$

其中 P_i 为第 i 个符号出现的概率，同样 \log 的底数可取 2，e，或 10，对应的熵的单位分别为比特/符号（bit/symbol）、奈特/符号（nat/symbol）和哈特/符号（hart/symbol）。

下面将利用以上两个基本理论来对图像进行二值分割。

三、 算法分析

算法的目的是选取一个阈值，使得该阈值下的二值化图像的熵最大。因此设计程序框图如下：



四、 JAVA 关键代码

```
public int threshold() {  
    //initial threshold  
    int threshold = 0;  
    //initial max entropy  
    double h_max = 0;  
    //normalized histogram  
    double[] nh = nhistogram();  
    //cumulative distribution  
    double[] cd = cumulative(nh);  
    for (int i = 0; i < 256; i++) {  
        //low range cumulative distribution  
        double cl = cd[i];  
        //low range entropy  
        double hl = 0;  
        if (cl>0) {  
            for (int j = 0; j < i+1; j++) {  
                if (nh[j]>0) {  
                    hl = hl - (nh[j]/cl)*Math.log(nh[j]/cl);  
                }  
            }  
        }  
        //high range cumulative distribution  
        double ch = 1-cl;  
        //high range entropy  
        double hh = 0;  
        if (ch>0) {  
            for (int j = i+1; j < 256; j++) {  
                if (nh[j]>0) {
```

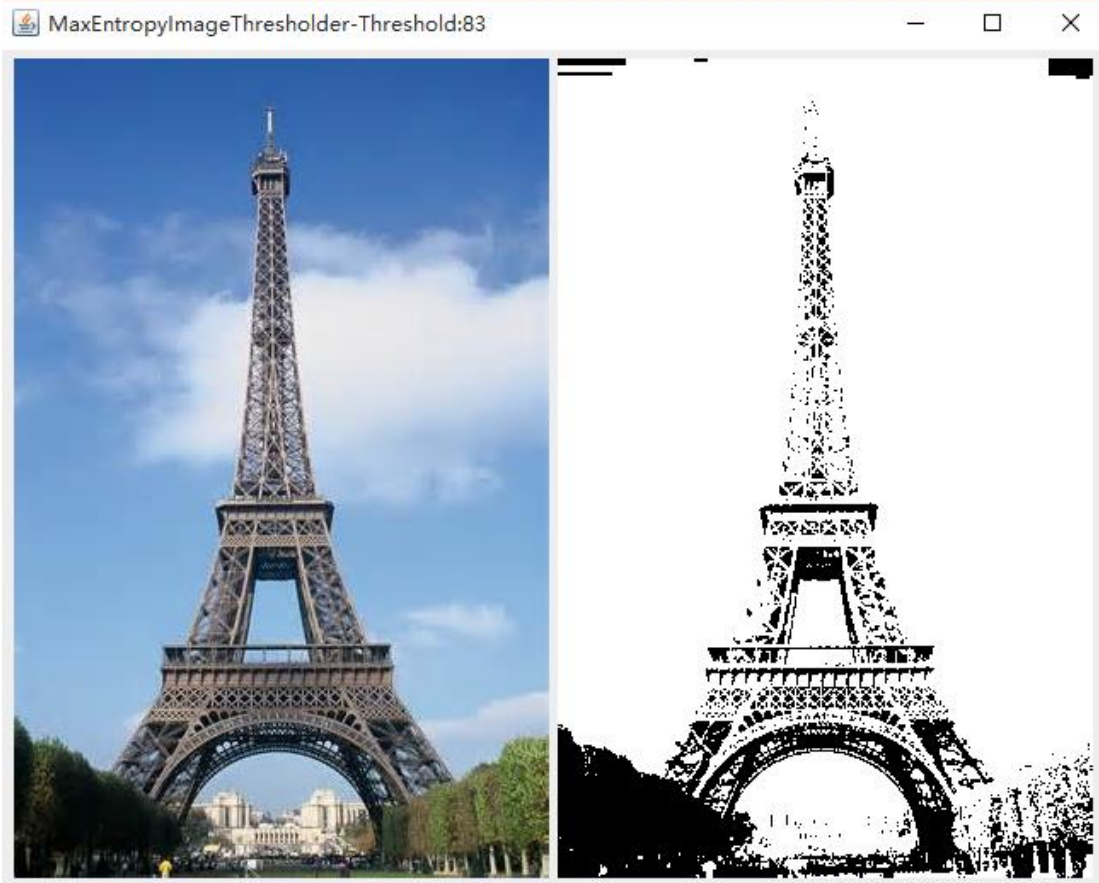
```

        hh = hh - (nh[j]/ch)*Math.log(nh[j]/ch);
    }
}
}
double h = hl+hh;
if (h>h_max) {
    //update max entropy
    h_max = h;
    //update best threshold
    threshold = i;
}
}
return threshold;
}

```

说明：以上只是项目算法 JAVA 代码实现代码中比较核心的部分，完整的可运行程序请查看附件中的 JAVA 代码包。

五、 处理效果示例





六、 进阶讨论

其他二值分割算法：

1. 直方图谷点分割法。

如果图像的目标和背景区域的灰度差异较大，则该图像的灰度直方图包络线就呈现双峰一谷的曲线，那么选取两峰之间的谷值就可以作为阈值来分割图像的目标和背景。这种方法在图像的目标和背景之间的灰度差异较为明显时，可以取得良好的分

割效果，通常可以满足我们的分割要求。虽然由于该方法对图像直方图的特殊要求和依赖，使其在图像分割中具有一定的局限性，但其操作简单运算量低，因此也被经常使用。处理效果示例如下，代码请参考附件。



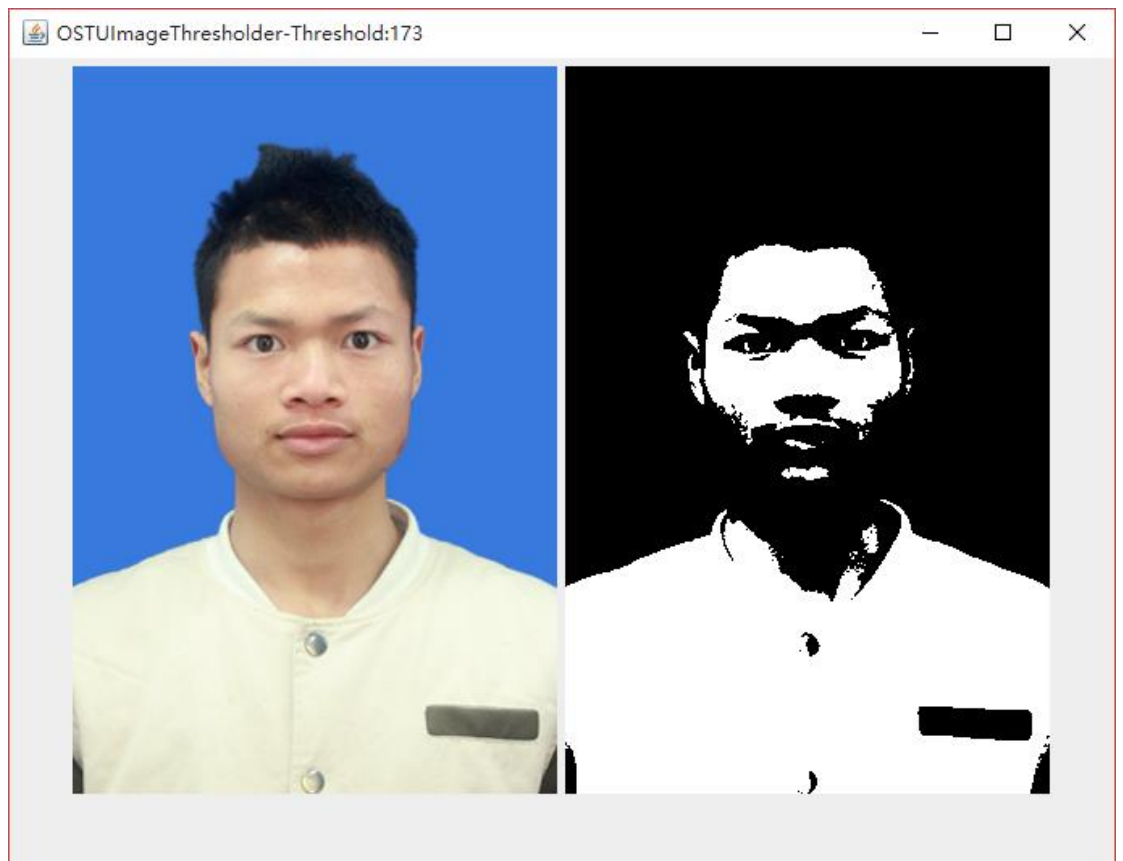
2. 基于灰度期望值的图像分割法。

在对随机变量的统计过程中，期望值是一个十分重要的统计特征,它反映了随机变量的平均取值，类似于物体的质量中心，因此从灰度“中心”进行分割应当是最佳的分割平衡点。该方法基于全局，分割效果要优于灰度差直方图法、微分直方图法以及非等同熵法，并且该算法的复杂度较低且处理速度较快。处理效果示例如下，代码请参考附件。



3. 最大类间方差法（OSTU）。

最大类间方差法也称大律法，其基本原理是：对一幅图像，记 T 为目标与背景的分割阈值， T 的取值范围是从最小灰度值到最大灰度值；目标点数占图像像素点数的比例为 w_1 ，平均灰度为 a ；背景点数占图像像素点数的比例为 w_2 ，平均灰度为 b ，图像的总平均灰度为 u ，则两类间方差的计算公式为： $v^2 = w_1 \cdot (a - u)^2 + w_2 \cdot (b - u)^2$ 。即阈值 T 将图像分成目标和背景两部分，使得两类总方差取最大值的 T 即为最佳分割阈值。因为方差是图像灰度分布均匀性的一种度量，方差越大说明构成图像的两部分差别越大，而部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小，因此使类间方差最大的分割意味着错分概率最小。大量实验结果表明 OSTU 算法对图像的分割质量通常都有一定的保障，对各种情况的表现都较为良好，可以说是比较好的一种分割算法。处理效果示例如下，代码请参考附件。



七、 结语

本文通过应用基于最大熵的自动图像阈值选取分割算法来对图像进行分割,并进一步讨论和实现了直方图谷点分割法、基于灰度期望值的图像分割法、最大类间方差法(OSTU)等其他非最大熵分割算法,可谓是信息论课程基本理论在数字图像处理领域上的一次不错的实践与拓展。