

## 大津二值化算法OTSU的理解



西河沿的風

只要我代碼敲得夠快，bug就追不上我！

57 人

### otsu 大津算法介紹：

OTSU算法是由日本學者OTSU於1979年提出的一種對圖像進行二值化的高效算法。

利用閾值將原圖像分成前景，背景兩個圖像。

前景：用 $n1, csum, m1$ 來表示在當前閾值下的前景的點數，質量矩，平均灰度

背景：用 $n2, sum - csum, m2$ 來表示在當前閾值下的背景的點數，質量矩，平均灰度

當取最佳閾值時，背景應該與前景差別最大，關鍵在於如何選擇衡量差別的標準，而在otsu算法中這個衡量差別的標準就是最大類間方差，在本程序中類間方差用 $sb$ 表示，最大類間方差用 $fmax$

這段引用自百度百科，不是很好懂。

### otsu 大津算法原理

otsu 大津算法是一種圖像二值化算法，作用是確定將圖像分成黑白兩個部分的閾值。

將圖像背景和前景分成黑白兩類很好理解，但是如何確定背景和前景的二值化界限（閾值）呢？

▲ 贊同57



● 10 條評論

➦ 分享

♥ 喜歡

★ 收藏

📄 申請



對於不同的圖像，這個閾值可能不同，這就需要有一種算法來根據圖像的信息自適應地確定這個閾值。

首先，需要將圖像轉換成灰度圖像，255個灰度等級。

可以將圖像理解成255個圖層，每一層分佈了不同的像素，這些像素垂直疊加合成了一張完整的灰度圖。

我們的目的就是找到一個合適的灰度值，大於這個值的我們將它稱之為背景（灰度值越大像素越黑），小於這個值的我們將它稱之為前景（灰度值越小像素越白）。

怎麼確定這個值就是我們想要的值呢？

這裡引入方差的概念，方差越大，相關性越低，黑白越分明。

我們將每一個灰度值之上下之間的像素的方差求出來不就行了嗎？找到方差最大的那個灰度值，那個就是我們想要的二值化分隔閾值。

先定義幾個符號代表的意義：

$h$ ：圖像的寬度

$w$ ：圖像的高度（ $h*w$  得到圖像的像素數量）

$t$ ：灰度閾值（我們要求的值，大於這個值的像素我們將它的灰度設置為255，小於的設置為0）

$n_0$ ：小於閾值的像素，前景

▲ 贊同57



● 10 條評論

➦ 分享

♥ 喜歡

★ 收藏

📄 申請



$n1$ : 大於等於閾值的像素, 背景

$$n0 + n1 == h * w$$

$w0$ : 前景像素數量佔總像素數量的比例

$$w0 = n0 / (h * w)$$

$w1$ : 背景像素數量佔總像素數量的比例

$$w1 = n1 / (h * w)$$

$$w0 + w1 == 1$$

$u0$ : 前景平均灰度

$$u0 = n0 \text{灰度累加和} / n0$$

$u1$ : 背景平均灰度

$$u1 = n1 \text{灰度累加和} / n1$$

$u$ : 平均灰度

$$u = (n0 \text{灰度累加和} + n1 \text{灰度累加和}) / (h * w) \text{ 根據上面的關係}$$

$$u = w0 * u0 + w1 * u1$$

▲ 贊同57



● 10 條評論

🔗 分享

♥ 喜歡

★ 收藏

📄 申請



g: 類間方差 (那個灰度的g最大, 哪個灰度就是需要的閾值t)

$$g = w_0 * (u_0 - u)^2 + w_1 * (u_1 - u)^2$$

根據上面的關係, 可以推出: (這個一步一步推導就可以得到)

$$g = w_0 * w_1 * (u_0 - u_1)^2$$

然後, 遍歷每一個灰度值, 找到這個灰度值對應的 g

找到最大的g 對應的 t

代碼實現:

```
import cv2
import numpy as np

# 这里直接将数据转换成float32了, 方便后续计算
img = cv2.imread(r'001.jpg').astype(np.float32)

# 灰度化
def bgr2gray(img):
    b = img[:, :, 0].copy()
    g = img[:, :, 1].copy()
    r = img[:, :, 2].copy()

    gray_img = 0.2126 * r + 0.7152 * g + 0.0722 * b
    gray_img = gray_img.astype(np.uint8)

    return gray_img
```

▲ 贊同57



● 10 條評論

🔗 分享

♥ 喜歡

★ 收藏

📄 申請



```
# 大津二值化算法
def otsu(gray_img):
    h = gray_img.shape[0]
    w = gray_img.shape[1]
    threshold_t = 0
    max_g = 0

    # 遍历每一个灰度层
    for t in range(255):
        # 使用numpy直接对数组进行运算
        n0 = gray_img[np.where(gray_img < t)]
        n1 = gray_img[np.where(gray_img >= t)]
        w0 = len(n0) / (h * w)
        w1 = len(n1) / (h * w)
        u0 = np.mean(n0) if len(n0) > 0 else 0.
        u1 = np.mean(n1) if len(n0) > 0 else 0.

        g = w0 * w1 * (u0 - u1) ** 2
        if g > max_g:
            max_g = g
            threshold_t = t
    print('类间方差最大阈值:', threshold_t)
    gray_img[gray_img < threshold_t] = 0
    gray_img[gray_img >= threshold_t] = 255
    return gray_img

gray_img = rgb2gray(img)
otsu_img = otsu(gray_img)
cv2.imshow('otsu_img ', otsu_img )

cv2.waitKey(0)
cv2.destroyAllWindows()
```

▲ 贊同57 ▼

💬 10 條評論

🔗 分享

❤️ 喜歡

★ 收藏

📄 申請



發佈於2019-12-03

二值化算法

圖像處理

圖像

## 文章被以下專欄收錄



一個小白的Python學習路

## 推薦閱讀

### Matlab 小工具 (3) ----圖像二值化的閾值計算

圖像分割是圖像處理這門學科中的基礎操作，基於閾值的分割則又是圖像分割的最基本的難題之一，其難點在於閾值的選取。事實證明，閾值的選擇的恰當與否對分割的效果有著決定性的作用。由於閾...

若谷



### 圖像二值化處理

wqias...

發表於計算機視覺...



### Canny算子進行圖像邊

勤勞的打工人

▲ 贊同57



● 10 條評論

➦ 分享

♥ 喜歡

★ 收藏

📄 申請

## 10 條評論

⇌ 切換為時間排序



寫下你的評論...



知乎用戶

2020-05-09

灰度值越大越白...#FFF是白色...

👍 8



在路上

2020-11-21

除了255應該對應白，剩下的很不錯

👍 1



幻羽朧月

08-09

很不錯

👍 贊



eatfish

07-27

倒數第五條代碼拼錯了gray\_img = bgr2gray(img)

👍 贊



BATB

02-05

對我幫助很大，感謝答主

👍 贊



Angelo

2020-12-11

小白提個問題，程序好像可以優化，先統計了直方圖，然後再實現Otsu是不是會降低很大的時間複雜度？

👍 贊

▲ 贊同57



💬 10 條評論

➦ 分享

❤️ 喜歡

★ 收藏

📄 申請



Angelo

2020-12-11

小白提個問題，程序好像可以優化，先統計了直方圖，然後再實現Otsu是不是會降低很大的時間複雜度？

👍 贊



風一樣的鍵盤

2020-08-14

不錯，看明白了，哈哈

👍 贊



章思止望

2020-04-29

感謝答主

👍 贊



林青春

2020-03-02

不錯，說明白了。

👍 贊

▲ 贊同57



💬 10 條評論

🔗 分享

❤ 喜歡

★ 收藏

📄 申請