

引言

由于之前网购的维生素片，有时候忘了今天有没有吃过，就想对瓶子里的药片计数...在学习opencv以后，希望实现对于维生素片分割计数算法。本次实战在[基于形态学](#)的基础上又衍生出[基于距离变换的分水岭算法](#)，使其实现的效果更具普适性。

整体思路：

- ### opencv实现:

效果展示：



对于如何打上标签(即添加注水点)有两种办法:

第一种(基于轮廓):在二值化后,对图像寻找轮廓findContours,筛选出注水区域轮廓,然后通过drawContours对轮廓标记。

第二种(基于区域):在二值化后,先对寻找图像中的前景图(即注水点),再找到到背景图(进行膨胀),最后找到未知区域(背景减去前景,得到边缘图),通过connectedComponents()获取标记点。

相关API:

- 分水岭函数watershed函数原型

```
void watershed( InputArray image, InputOutputArray markers );
```

第一个输入参数 `image`，必须是CV_8UC3类型图像。

第二个输入/输出参数markers必须是32位单通道图像,和image尺寸一样。包含不同区域的轮廓,每个轮廓有一个自己唯一的编号。

在执行watershed函数后,算法会根据markers传入的轮廓作为种子,对图像上其他的像素点根据分水岭算法规则进行判断,并对每个像素点的区域归属进行划定,直到处理完图像上所有像素点。而区域与区域之间的分界处的值被置为“-1”,以做区分。

- 距离变换函数distanceTransform函数原型

距离变换运算用于计算二值化图像中的每一个非零点距自己最近的零点的距离。距离变换图像上越亮的点,代表了这一点距离零点的距离越远。

距离变换通常用于求解图像的骨骼和查找物体的质心(即获取距离变换的极大值)和计算非零像素到最近零像素点的最短距离。

```
distanceTransform( InputArray src, OutputArray dst, int distanceType, int maskSize, int dstType = CV_32F);
```

第一个输入参数src, 必须是CV_8UC1类型的二值图像(只有0或1)

第二个输出参数dst，表示的是计算距离的输出图像，输出类型是CV_32F/CV_8U的单通道图像，大小与输入图片相同。

第三个参数distanceType, 表示的是选取距离的类型, 可以设置为DIST_L1, DIST_L2, DIST_C

第四个参数maskSize, 表示的是距离变换的掩膜模板, 可以设置为3, 5(常用3)

第四个参数dstType, 表示输出类型, 可选择CV_32F/CV_8U

注:若输出类型为CV_32F, 想要显示原高交换后的骨骼图像, 需要对其归一化。(normalize)

先来看看第一种标记mark(基于轮廓)的方法:

(一)读入图像,形态学,二值化(消除噪声)

(二)距离变换(归一化显示),再二值化

公告



我的标签

- opencv 笔记(23)
- c++笔记(21)
- halcon笔记(17)
- Qt笔记(15)
- C#(2)

隨筆分類

- C#(2)
- C++(9)
- C++面向对象总结(10)
- Halcon(10)
- Halcon缺陷检测实例转OpenCV实现(5)
- Love(1)
- opencv(12)
- opencv实战项目(6)
- QT常用控件(4)
- QT从入门到入土(10)
- QT经验(1)
- vs+halcon(C#)(4)
- VS+Qt+Halcon混合编程(1)
- 机器视觉(6)

隨筆檔案

2022年3月(2)
2022年1月(1)
2021年8月(16)
2021年7月(20)
2021年6月(24)
2021年5月(18)

阅读排行榜

1. opencv——感兴趣区域(ROI)的分析和选取[详细总结](6063)
2. halcon——缺陷检测常用方法总结(模板匹配(定位)+差分)(5509)
3. QTM从入门到入土(一)——Qt5.14.2安装教程和VS2019环境配置(5434)
4. QTM从入门到入土(五(1))——多线程(QThread)(5392)
5. halcon——缺陷检测常用方法总结(颜色+区域结合)(4963)

推荐排行榜

1. halcon——缺陷检测常用方法总结 (颜色空间域结合)(6)
2. QT从入门到入土(八)——项目打包和发布(5)
3. QT从入门到入土(三)——信号和槽机制(4)
4. halcon——缺陷检测常用方法总结 (测量拟合)(4)
5. QT从入门到入土(四)——文件的读写操作(3)



部分标签markers轮廓数据截图。可以看到0代表背景，轮廓线用正数索引标识。



C:\Users\86175\Desktop\维生素片机器

药片检测个数: 23

再来看看第二种标记mark(基于区域)的方法:

(一)读入图像,形态学,二值化(消除噪声)

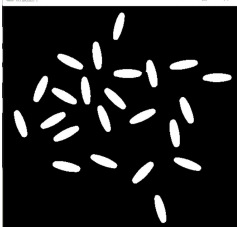
(二)对二值化图像进行膨胀操作,得到大部分是背景的图片

(三)通过对二值图像距离变换得到前景图片(即注水点)

```

# 距离变换
distance = transform(src_binary, src_distance, DIST_L2, 3, 5);
imshow("距离变换", src_distance);
normalize(src_distance, src_distance, 0, 255, NORM_MINMAX);
double my_minv = 0.0, my_maxv = 0.0;
minMaxIdx(src_distance, &my_minv, &my_maxv);
threshold(src_binary, foreground, 0.4 * my_maxv, 255, THRESH_BINARY);
foreground.convertTo(foreground, CV_8U);
imshow("前景图片", foreground);

```



(四)通过背景与前景的差值,得到未知区域(即边缘所在区域)

```
//得到未知区域
unknown = background - Foreground;
imshow("未知区域", unknown);
```



(五)得到这些区域以后,我们可以获取该水点的标签,通过connectedComponents实现(即获取markers标签)

```
//创建标记图markers
Mat markers = Mat_<u>int</u>(src.size(), CV_32S);
int num = connectedComponents(Foreground, markers, 8);
cout << num << endl;
markers = markers + 1;
for (int i = 0; i < unknown.rows; i++)
{
    for (int j = 0; j < unknown.cols; j++)
    {
        if (((int)unknown.at<u>uchar</u>(i, j)) == 255)
        {
            markers.at<u>int</u>(i, j) = 0;
        }
    }
}
```

添加暗角效果

现在我们已经知道哪些是背景,哪些是药片(前景区域)。

因此我们可以创建一个标签(和原图大小,类型为CV_32S),通过connectedComponents函数对前景区域进行标记

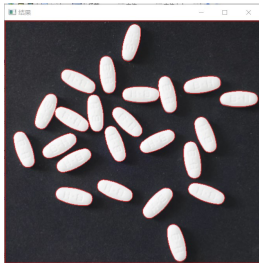
该函数相关博文:opencv——连通域标记与分析 - 唯有自己强大 - 博客园 (cnblogs.com)

该函数会对前景区域进行连通域分析,并将背景设定为0,其他区域从1开始正整数标记(这就是我们的种子,水漫时会从这里漫出),结果返回给markers。

但是对于分水岭算法,会将为0的区域认为是未知区域,因此要markers整体加一。

(六)进行分水岭操作,并显示边缘

```
watershed(src, markers);
for (int row = 0; row < markers.rows; row++)
{
    for (int col = 0; col < markers.cols; col++)
    {
        if (markers.at<u>int</u>(row, col) == -1)
        {
            src.at<u>Vec3b</u>(row, col) = Vec3b(0, 0, 255);
        }
    }
}
imshow("结果", src);
```



由于分水岭算法会将找到的边缘在markers置为-1,因此我们对原图操作,将索引为-1的位置的像素值改为红色(即显示边缘)。

参考链接:OpenCV——分水岭算法 - 山上有风景 - 博客园 (cnblogs.com)

(8条消息) c++和opencv小知识:基于距离变换的分水岭算法(固定流程)_梦游城市的博客-CSDN博客

(8条消息) OpenCV分水岭算法图像分割_冰沐bing的博客-CSDN博客

分类: opencv实践项目

标签: opencv 笔记

标签: opencv 笔记

标签: opencv 笔记

唯有自己强大

关注 - 10

粉丝 - 114

» 上一篇: opencv——几何变换原理与实践

» 下一篇: opencv——连通域标记与分析

posted @ 2021-05-17 16:12 唯有自己强大 阅读(3001) 评论(2) 编辑 收藏 举报

登录后才能查看或发表评论,立即 [登录](#) 或者 [注册](#) 博客园首页

最新评论 刷新页面 返回顶部

推荐阅读:

- Three.js 实现 3D 开放世界小游戏: 阿狸的多元宇宙
- 一个蛮烦问题: 非步操作会创建线程吗?
- 戏院领域驱动设计(十九)——外租
- ASP.NET Core 在 IIS 下的两种部署模式
- 记我第一次做线下技术分享的那些事

最新新闻:

- 苹果给 Mac OS 系统做微端网页版 可免费体验
- 哪吒有戏: 不靠剧小爆
- 华为聘任公用打工: 前副人生副峰 后副裁总走人
- 亏损、裁员、溃败, 非还款加盟共享充电宝?
- 宇宙有个「孪生兄弟」? 一个时光倒流的镜像「反宇宙」

» 更多新闻...

