联系 订回 首页 新随笔 管理 随笔 - 81 文章 - 0 评论 - 21 阅读 - 10万

opencv实战——机器视觉检测和计数

引言

在机器视觉中,有时需要对产品进行检测和计数。其难点无非是对于产品的图像分割。

由于之前同院的修生集片,有时候忘了今天有沒有吃过。就想时瓶子里的所片计数。在学习opency以后,希望实现时干你生集片分割计数算法,本文实现在<mark>是于多二学</mark>的基础上又衍生比<mark>是于庙或文司的分址等算是</mark>,使其实现的效果更具管遍性。

基于形态学的能生素片检测和计数

整体思路:

- 读取图片
 形态学处理(在二值化前进行适度形态学处理,效果俱佳)

int main(int argc, char** argv)

获取轮廓索引,并筛选所需要的轮廓
 面出轮廓,显示计数

opencv实现:

Nature of the second of the s

Point2i PL; for (size_t i = 0; i < contours.size(); i++)

Area - contourAnas(contours[i]);
if (gres (50))emtimus;
FL - contourS[i],frent();
Scalar color - Scalar(reg.uniform(0, 255), reg.uniform(0, 255));
Scalar color - Scalar (reg.uniform(0, 2

效果展示:



由上限可以看的。原图在经过彩态学处理后,可以去除很多细节,简化后线的药片分割操作。

但是在计数结果图上发现。索引17号药片并没有完全分割(实际上修改形态学的结构元素尺寸(改为20°20)也可以完全分离这两个药片)。

对于复杂的产品图片,我们可以使用基于距离变换的分水岭等该对其分割。

基于距离变换的分水岭算法检测和计数

OpenCV 果用了基于标记点的分水岭算法,在这种算法中我们更设置哪些出谷点会汇急,哪些不会,这是一种交互式的同像分别,我们要做的就是给我们已知的可象打上不同的标签(周溪加注水血),然后实施分水岭算法,每一次灌水,我们的标签就会被更新,当两个不同顺色的标签相遇可效构建规,直到将所有山楂淹没,最后我们得到的边界对象(提明)的位为一1。

对于如何打上标签(即指加注水点)有两种办法:

opencv中,对于一张二值化的图像。后途处理方式有两种。第一种方式就是利用findContours,drawContours等函数进行轮廓分析(opencv以对轮廓的处理为主)。第二种方式就是计算连通域进行区域分析。

第一件(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,先对寻找函像中的前景图(即注水点),再寻找到青景图(进行膨胀),最后找到未知区域(青景减去前景,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类型阳像。

第二个给入/给出参数markers必须是32位总通道图像。和image尼寸一样。包含不同区域的轮廓、每个轮廓有一个自己唯一的给是

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,并对与个像素点的区域扫属进行划定,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"一",以散区分

距离变换函数distanceTransform函数原型

距离变换运算用于计算二位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_32r);

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:若抽出类型为CV_32F,想要基示原高变换后的骨架图像,需要对其归一化。(normalize)

先來看看第一种标记mark(基于轮廓)的方法:

(一)读入图像、形态学、二位化(消除噪声)



(二)距离变换(归一化显示), 再二值化

distanceTransform(src_binary, src_distance, DIST_12, 3, 5);
normalize(src_distance, src_distance, 0, 1, NOMM_NIMBOX);
inshow("語言改作", src_distance);
threshold(src_distance, src_distance, 0.4,1, THRESH_BINARY)
inshow("第二化化", src_distance);



我的标答

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

随笔分类

C++面向对象总结(10)

随笔档案

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)

间证挂行槽

1. opencv——膨兴趣区域(ROI)的分析和选取[详细总结](6063)

3. QT从入门到入土(一)——Qt5.14.2安装教程和VS2019环境配置(5434)

5. halcon---缺陷检测常用方法总结(领域空间域结合)(4963)

1. halcon--缺陷检测常用方法总结(领域空间域结合)(6)

QT从入门到入土(八)——项目打包和发布(5)
 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)

•

经过距离变换后的二值化,可以清晰看到,药片以及完全分割开来。

(三)打上标签(添加注水点),基于轮廓

```
/ 「視信心点mass的心原理。包証分分析的水川
are_distance.comvertFolare_distance, CV_SUC1);
vectorevectorがDairthlo.comcurs;
fininContours:rec_distance, cutorecs, RETS_TREE, CMAIN_APPROX_SIDPLE);
//心限元abce
Mat_marker - Matirizeros(strc.size(), CV_125);// //因为分水种后的心体有效是-1. 所以必须使用有符号的/_125
for (size_t t = 0; t < contours.size(); t++)
(finincontours.size(); t++)
                    )
circle-measure, contours, static_castcisto(t), Scalar(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(static_castcisto(
                                    drawContours(markers, contours, static_cast<int>(t), Scalar(static_cast<int>(t) + 1), -1);//轮腺数字编号
```

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

(四)进行分水岭操作,并给分水岭后的区域随机上色,并打印出检测的药片个数。

```
生成種則形

vector/vec2bb colors;

for (size_ti=0; i < contours.size(); i++) (

int = -theRMC().uniform(0, 255);

int g - theRMC().uniform(0, 255);

colors.push_back(Vec2bf(uchar)b, (uchar)g,
// 形色填充与量价品并
Nat det1 = Nat;:recro(markers.mise(), CV_ROC3);
int index = 0;
for (int row = 0; row < markers.rows; rows+) (
for (int row = 0; row < markers.rows; rows+) (
index = markers.at(int)(row, col);
                      if (index > 0 && index <= contours.size()) {
   dst1.at<Vec3b>(row, col) = colors[index - 1];
                      }
else {
  datl.at<Vec3b>(row, col) = Vec3b(0, 0, 0);
```



■ C:\Users\86175\Desktop\维生素片机器; 药片检测个数: 23

再來看看第二种标记mark(基于区域)的方法:

(一)读入图像, 形态学, 二值化(消除噪声)

```
Not foreground, background, subcomes://包括图表 背景,未放区域
Not sern serplinery, data, ser_distances
ser - invended: foreground-suller/ 也是主有相關定位指制证表示的";
inshow (個別年, ser)
Sata barcal - spetimenteringlement(DODEP, MCTT, Size(1, 3), Point(-1, -1));
sumpublosyfictors, dat, NORSE, OFEN, Normal);
inshow (NORSE, dat, COLOR, NORSE, DODE);
vertice(dat, dat, COLOR, NORSE, DODE);
inshow(1-06-6, ser_binary);
```



(二)对二值化图像进行膨胀操作。得到大部分是背景的图片

```
src_binary, background, kernel, Point(-1, -1), 3);
"背景阳片", background);
```



(三)通过对二值图像距离变换得到前景图片(即注水点)

```
// 医乳空池
distance/rossfore(src_Mnary, src_distance, DIST_L2, 3, 5);
inshev(音楽文学、src_distance);
normalize(src_distance);
normalize(src_distance), src_distance), 0, 255, 10000 (11000X);
double wy, fur - 0.0, wy, saw - 0.0;
niconatize(src_plinary, New, Saw, New, Naw);
threscholize(src_distance, foreground, 0.4 * wy_manw, 255, 10002H_BINARY);
foreground.convextTo(foreground, CV_BU);
inshow(*粉変別所*, 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
unkonum - background - foreground;
imshow("未知区域", unkonwn);
```



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

```
//包括和arkers
Mat markers = Nat(arc.size(), CV_322);
int num - connoctedComponents(foreground, markers, 8);
cost < num << medi;
markers = markers = num << medi;
for (int i = 0; i < unknown.rows; i++)
              for (int j = 0; j < unknown.cols; j++)
                   if (((int)unkonwn.atcuchar>(i, j)) == 255)
```

学無理解该步骤:

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

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

該面数会对前景区域连通域分析,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

```
for (int col = 0; col < markers.cols; col++)
 imshow("結果", 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_冰冰bing的博客-CSDN博客

分类: opency实战项目 标签: opency 笔记

```
#大阪東 美世界 (松田被文 6) №
```

« 上一篇: opencv——几何变换原理与实现 » 下一篇: opencv——连通城标记与分析

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

刷新评论 刷新页面 返回頂部 🤜 登录后才能查看或发表评论。立即 <u>登录</u> 或者 <u>逛逛</u> 博客园首页 機構機等:
- Three」。東周3D开放世界小游点:同種的多元干亩
- 一个高新月起: 身步龍件会計量比程吗?
- 元政宗斯原政治(十七九) 一分验
- ASP.NET Core 在 US 下的两种部署模式
- 记表第一次微线下技术分享的那些事

