

基于机器视觉的物料检测算法设计

戴星宇, 金珣宏, 王轩宇

上海理工大学, 上海

收稿日期: 2022年11月11日; 录用日期: 2022年12月23日; 发布日期: 2022年12月31日

摘要

传统的物料检测方法主要采用人工检测, 然而人工检测劳动强度较大, 工作效率低。本文采用基于机器视觉的图像处理技术, 提出了自动检测通过传送带后下落的物料以及其计数的方法, 大大减少人工测量、计数的工作量。首先对图像进行预处理, 即采用旋转、滤波去噪等方式将原始图像调整为方便后续处理的图像, 接着对感兴趣的区域(ROI)进行图像分割, 通过对下落背景的形状以及在图像中占据的主要位置的把握, 将其截取出来单独处理。得到截取的下落图像后, 分析下落背景以及物料灰度值差异, 根据其特点(圆弧度、面积)将物料作为目标从背景中提取分割出来。将分割后的图像通过腐蚀运算将可能连在一起的两个或两个以上的物料分开, 再膨胀使其接近原有面积并其所在位置中心点坐标进行记录, 最终实现计数算法。

关键词

物料检测, 图像分割, 目标跟踪, 目标计数

Design of Material Detection Algorithm Based on Machine Vision

Xingyu Dai, Xuanhong Jin, Xuanyu Wang

University of Shanghai for Science and Technology, Shanghai

Received: Nov. 11th, 2022; accepted: Dec. 23rd, 2022; published: Dec. 31st, 2022

Abstract

Manual detection is mainly used in traditional material testing. However, the method of manual detection is low in efficiency and high in cost has high labor intensity and low work efficiency. In this paper, a new method of automatically detecting the falling materials passing through the conveyor belt and counting them is proposed. The Image Processing Technology which is based on the machine vision is adopted in this new method. Consequently, the workload of manual measurement

and counting can be greatly reduced. First, the image is preprocessed, that is, the original image is adjusted to the image for subsequent processing by rotation, filtering and de-noising. Secondly, the Region of Interest (ROI) is segmented. By grasping the shape of the falling object and the main position it occupies in the image, we extract it for further processing. Thirdly, by analyzing the falling object and the difference of material gray value, we segment from the background as a target according to its characteristics (roundness and area). Fourthly, two or more materials that may be connected together are separated from the segmented image through corrosion operation, which is expanded to make them close to the original area. Finally, the coordinates of the center point of their location are recorded to facilitate the final counting.

Keywords

Material Inspection, Image Segmentation, Target Tracking, Target Count

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

相比于人的视觉，机器视觉可以对采集的图像进行更快速的处理；同一时间处理的对象可以更多；在不同环境下也有各种不同功能能够得以实现；灰度分辨能力强；空间分辨能力强，可以观测到微米-天体等目标；对环境的适应能力也比人眼要强得多。相比于计算机视觉，机器视觉涉及图像的处理、人工智能以及模式识别。机器视觉集合机械、光学、电子、软件系统、检验自然物体和材料、人工缺陷和生产缺陷的工程。该方法出现提高了操作效率保障产品过程安全。机器视觉是将计算机视觉应用在工业自动化上[1]。目前机器视觉技术应用领域广泛，主要包括半导体、军工、航天、工业制造、医疗等多方面领域[2]。

计数问题在生活中无处不在，如何通过机器视觉来进行计数是当今机器视觉技术的主流方向之一。越来越多的人开始研究计数问题，也对一些特定的问题提出了针对性的解决方案和研究成果。

从国内来看，有一种图像计数方法[3]主要针对工业场景中传送带上运动的物料袋被称之为在线运动物料袋智能图像。通过对工业场景中前景图像应用线性差值的采集，来计算某时刻前景图像的概率分布及其分布密度，最后比较概率分布密度大小确定传送带的在线运动物料袋数目；在半导体工业生产上，田齐和张令提出了基于机器视觉的晶圆裸片计数算法[4]，通过直线算法并利用晶圆的一些规律测出晶圆裸片的数目；针对鱼苗饲养、运输和销售过程中对一定量的鱼苗做精确计数这一问题上，郭俊、郑堤等人提出了基于机器视觉的鱼苗计数算法[5]。该方法通过每个图像的面积特征，建立图像特征值与鱼苗数量之间的模型，再利用跟踪拍摄多组图像数据回归最优模型参数，应用到计数累加和可以实现精确计数。

在国外，Chinmaya Panigrahy 等人提出灰度图像定量纹理测量[6]，他们使用改进的差分盒计数法的分形维数解决了大多数先进方法都存在的问题——框的过度计数和计数不足的；Flavio E. Spetale 等人提出的针对番茄果实中的气孔计数[7]，提出 LocAnalyer 加快处理传统视觉检查水果横截面来确定子房的数量，并减少了主观性。Gulpreet Kaur Chadha 通过对图像进行形态学纹理特征提取，阈值分割以及霍夫变换对血液中的红细胞提出计数算法，且该算法的总体计算准确度达到了 91.667%，极大的方便了医疗时通过显微镜手动计数检查患者的细胞数。减少大量人力，可以快速有效的诊断疾病。

为了解决传统方法测量费时费力。本文设计了基于机器视觉的物料检测算法，对采集到的下坠物料图像进行处理后的做计数算法。大大节约了时间成本和人工劳动力，精度上也有一定保证，本文提出的计数算法精度在 88.89%。

2. 图像预处理

2.1. 图像的平滑去噪

本实验主要是针对相机连续采取的通过传送带后下落的物料进行提取，并对其数目进行统计。连续采集的图像可能会因为光线条件等外在环境变化而受到影响，被采集物料的外在显性特征也可能在每一帧上表现得不一致。因此，在对感兴趣的物料提取追踪前，需要对相机采集到的图像进行预处理。

物料检测算法流程图如图 1 所示。

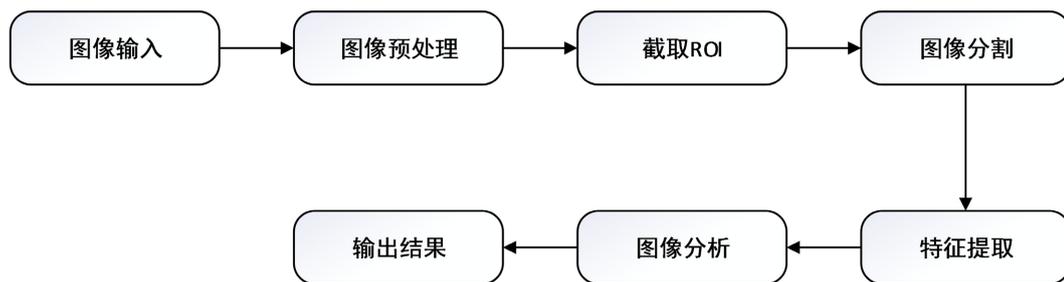


Figure 1. Flow chart of material detection algorithm

图 1. 物料检测算法流程图

由于相机的位置及角度影响，采集图像不处于水平位置，将不利于后续将物料与背景分割开来；或者因为外界因素引起的一些来源不明的噪声和失真等。如果直接对图像进行分割提取，则物料的部分特征会与背景相似，会提升截取物料的难度。部分噪声的影响可能会导致最终计数上出现大幅度的误差，影响最终结果的计算。因此，针对相机所采取到的实时图片进行预处理以便于后续操作和减小误差。这一系列内部或者外部产生的噪声，可以采取平滑滤波的方法进行去除。具体的算法分为两种，一种基于像素点的滤波，即空域滤波；另一种是对图像进行傅里叶变换进行修改，即频域滤波[8]。

2.1.1. 空间滤波实现原理

空间滤波增强采用模板处理方法对图像进行滤波，以去除图像噪声或增强图像的细节。模板本身被称为空间滤波器。空域滤波是将领域内的图像像素值同对应的与领域有相同维数的子图像值相作用。

在 $M \times N$ 的图像 f 上，使用 $m \times n$ 的滤波器：

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t) \quad \text{式 2.1}$$

其中， $m = 2a + 1$ ， $n = 2b + 1$ ， $w(s, t)$ 为滤波器系数， $f(x, y)$ 为图像值，可简化为：

$$R = w_1 z_1 + w_2 z_2 + \dots + w_m z_m = \sum_{i=1}^{nm} w_i z_i \quad \text{式 2.2}$$

其中， w 是滤波器系数， z 是与该系数对应的图像灰度值， nm 为滤波器中包含的像素点总数。

2.1.2. 线性平滑滤波图像增强：领域平均法

领域平均法又称均值滤波，它的核心思想是将周围领域的灰度值均值用来代替该点的灰度值，通过

不同模板使其周围不同位置的领域像素有不同的比重，这样可以避免均值滤波之后将目标也滤除或者除噪效果不明显等情况。典型的均值滤波除噪声模板如下：

$$\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{式 2.2}$$

2.1.3. 非线性平滑滤波图像增强：中值滤波法

中值滤波是一种非线性的滤波器，可以克服先行平滑滤波器引起的图像模糊问题，在滤除噪声的同时保证图像的边缘信息不受破坏。它的基本原理是使用某种中值滤波窗口，并对该窗口所覆盖的所有像素灰度值进行排序，用中间值代替所有像素的中点像素灰度值，弥补使用均值滤波造成的不足。

均值滤波主要用来处理图像中含有随机出现的黑白亮度值的椒盐噪声、只含有随机白亮度值或随机黑亮度值的脉冲噪声等。中值滤波主要用来处理椒盐噪声。均值滤波的缺点会将图像模糊化；中值滤波则会造成图像的不连续性。在实验所采集的图片噪声不是很多，所以采用中值滤波做降噪处理。

2.2. 图像的几何运算

本论文采用 Halcon 软件进行实验。Halcon 中实现图像旋转的方式有两种。一种是按照图像中心旋转的 rotate_image 算子，该方式实现简单。二是可以按照任意位置旋转的 affine_trans_image，该方式实现较为复杂。本文将采用 rotate_image 算子进行旋转，如图 2。

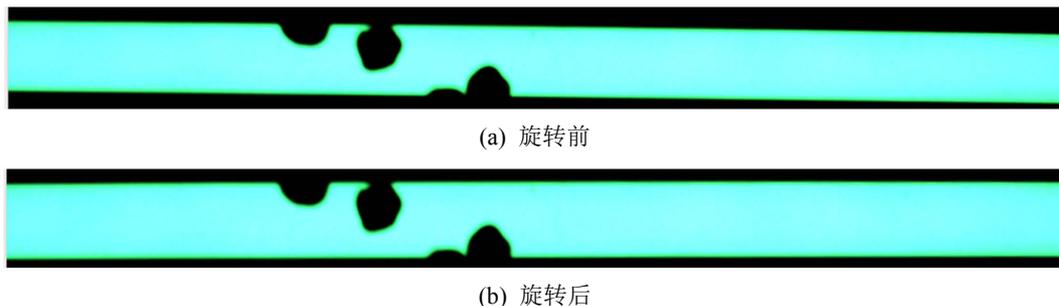


Figure 2. Rotate_image operator processing effect
图 2. Rotate_image 算子处理效果

2.3. 感兴趣区域获取

目前 ROI 提取技术广泛应用于机器视觉中目标区域的选定与识别，通过 ROI 可大大减少计算机将要计算的像素数目，大大减少图像处理的工作量。既节约了时间还保证了处理的精度。本文截取 ROI 的流程图如图 3 所示。



Figure 3. Flow chart of falling background interception
图 3. 下落背景截取流程图

通过对图像进行旋转使得图像中的物料下落背景在水平位置，然后采取根据下落背景的相对位置信息以及尺寸大小进行分割提取，从而得到下落区域。第一步需要获取整个原始图像的尺寸信息。由于是

相机连续采集通过传送带后下落的物料，经观察发现每幅图片下落位置在整组图片中位置大致相同，处于正中央且宽度与原图宽度相近，根据这一特性，将整组的下落背景区域都提取出来，生成相对应的区域。

2.4. 对比度拉伸

一幅图像中会有最亮的区域和最暗的区域，而其亮度层级的测量就是图像的对比度。差距越大代表图像对比度越大；差距越小代表图像对比度越小。在对采集的图像进行处理之前，某些图像可能会因为相机采集时对比度的缺陷导致其失去重要轮廓特征，因此在处理图像之前要对其进行图像对比度增强。

由于光照不足、传感器动态范围受限或透镜光圈设置等一系列因素会导致采集的图像对比度不理想。所以需要通过对比度拉伸，即增强图中各部分的反差来增强图像。实际过程中是增加某些灰度值动态范围来增强对比度。根据不同的目的可以设计出不同的增强对比度的函数。对比度拉伸的基本思想是针对图像的灰度直方图进行调整，从而突出目标与背景之间的差异。具体的对比度拉伸方法有线性拉伸、非线性拉伸和一些分段拉伸，如指数拉伸、对数拉伸等

实验采用 `emphasize` 算子对采集图片进行对比度拉伸处理，使得图像中下落物料轮廓更加清晰，目标更凸显。对比度拉伸结果如图 4。

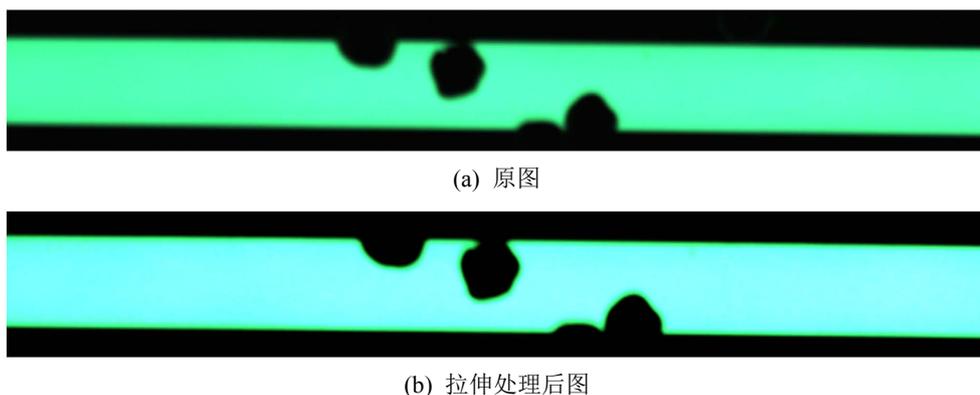


Figure 4. Contrast stretching results
图 4. 对比度拉伸结果

3. 分割算法

3.1. 边缘检测的图像分割

在做完图像预处理之后，采集的图像可以进行后续的分割处理。图像分割的主要目的有两个：将下落的物料检测出来，获取其轮廓特征并加以总结；确定所有下落的物料平均面积的大小，为计数做准备。

图像的边缘的意义是由某一区间内改变跨度到另外一个更大或者更小的灰度区间，它在图像的中起到了桥梁的作用，既是背景的一部分，又是目标的一部分。边缘检测，就是检测目标区域与背景区域中出现的灰度值大幅度跳跃的像素点的集合。这种大幅的跳跃形成了背景与目标之间的差异，这种差异表现在灰度值上。检测目标区域的边缘即是检测这些差异所在像素的具体位置[9]。根据边缘区域灰度值的特殊跳跃性，由此联想到了导数。图 5 是边缘与导数之间的关系。

目前的边缘检测方法中主要有三种方法：微分算子法、曲线拟合法以及小波变换法。其中微分算子法的主要原理就是将边缘和导数间的关系用于查找目标与背景间的跳跃像素[10][11]。接下来介绍 Halcon 里的几种常见边缘检测算子，并将其运用于实验进行对比。

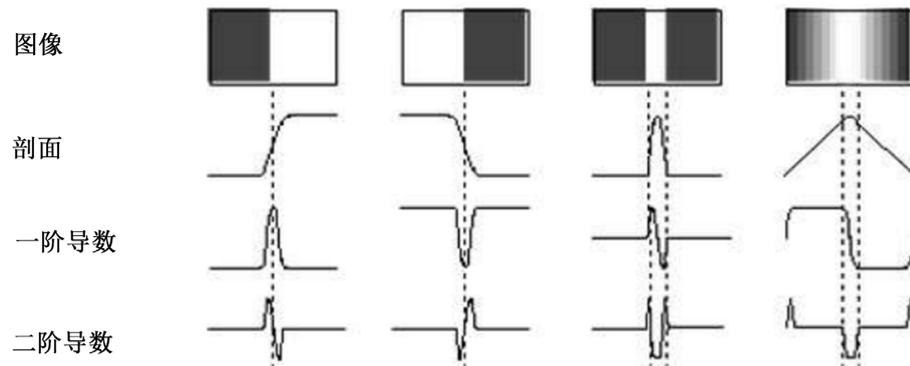


Figure 5. Relationship between image edge and derivative
图 5. 图像边缘和导数间关系

Sobel 算子通过对相邻区域梯度值计算像素梯度, 根据定值取舍, 它是一个一阶微分算子。它的公式如下:

$$S = \sqrt{dx^2 + dy^2} \quad \text{式 3.1}$$

Sobel 算子是 3×3 算子模板。其计算模板如下:

$$\begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \quad \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix} \quad \text{式 3.2}$$

左边的是 x 方向的模板, 右边的是 y 方向的模板。选择特定阈值, 若计算出的梯度值大于该阈值, 则认为该像素落在边界区域里面, 反之, 则落在边界之外。根据该检测原理, Sobel 算子对于纹理清晰简单的图像有很大的优势, 算法简单运算量小, 运算速度快。如果遇到复杂的图像, Sobel 算子会因为只检测了 X 与 Y 两个方向的边缘, 所以最终成像效果不会很好, 容易受某些噪声干扰。

Prewitt 算子检测原理与 Sobel 相类似, 但在 Sobel 上做了提升。它不光考虑了像素点本身灰度, 还考虑了邻域像素灰度。在做微分计算之前对图像先进行平滑处理。Prewitt 算子的计算公式与 Sobel 算子的公式相同, 模板同样为 3×3 , 左侧为 x 方向, 右侧为 y 方向:

$$\begin{vmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix} \quad \begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix} \quad \text{式 3.3}$$

Sobel 与 Prewitt 都是检测梯度的算子, 它们根据一阶导数, 利用图像边缘时一阶导数出现在极值点这个性质进行检测。除此之外, 根据边缘与导数间的关系图还可以利用二阶导数的零点做文章。Laplacian 算子便是利用零点交叉这个性质根据完成的水平和垂直方向上的二阶偏导数计算检测边缘。Laplacian 算子对模板的要求较高, 所有系数和为零, 中心系数要求是正数邻域系数为负数或 0。虽然 Laplacian 算子的检测在理论上完全成立, 但其无法运用在实际检测过程。相比于梯度算子, 二阶零点交叉的处理方法对噪声更为敏感, 所以单独使用二阶算子很难对边缘准确检测, 会造成很多边缘误判结果。因此 Laplacian 算子多用于理论而不用于实践。

由于 Laplacian 算子无法应用于实际所以提出了 Log 算子。Log 算子全称 Laplacian-Gauss 算子。针对 Laplacian 算子对噪声太过敏感这一问题, 它在进行边缘检测前先对图像做了平滑去噪处理。通过对图像使用高斯滤波平滑处理之后, 再通过其检测运算后的零点计算得出边缘像素点。

其像素模板为:

$$\begin{bmatrix} -2 & -4 & -4 & -4 & -2 \\ -4 & 0 & 8 & 0 & -4 \\ -4 & 8 & 24 & 9 & -4 \\ -4 & 0 & 8 & 0 & -4 \\ -2 & -4 & -4 & -4 & -2 \end{bmatrix} \quad \text{式 3.5}$$

二维高斯函数 $G(x, y)$ 定义如下:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad \text{式 3.6}$$

将 $G(x, y)$ 与原图像做卷积运算得到平滑图像 $I(x, y)$:

$$I(x, y) = G(x, y) * f(x, y) \quad \text{式 3.7}$$

接着运用 Laplacian 算子对平滑图像运算得到其二阶导数 $M(x, y)$:

$$M(x, y) = \nabla^2 \{I(x, y)\} = \nabla^2 \{G(x, y) * f(x, y)\} \quad \text{式 3.8}$$

由此可推出 Log 算子的公式为:

$$\nabla^2 G(x, y) = \frac{1}{2\pi\sigma^2} \left(\frac{x^2 - y^2}{\sigma^2} - 2 \right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad \text{式 3.9}$$

Roberts 算子是一种利用局部差分算子寻找边缘的算子, 其检测原理是将算子近似看作由一个方向垂直的差分逼近, 采用相邻对角像素差代替梯度值进而与模板做卷积运算。它是 2×2 算子的模板。

其公式为:

$$g(x, y) = \left[\sqrt{f(x, y)} - \sqrt{f(x+1, y+1)} \right]^2 + \left[\sqrt{f(x, y+1)} - \sqrt{f(x, y)} \right]^2 \quad \text{式 3.10}$$

不同算子效果图 6 如下。

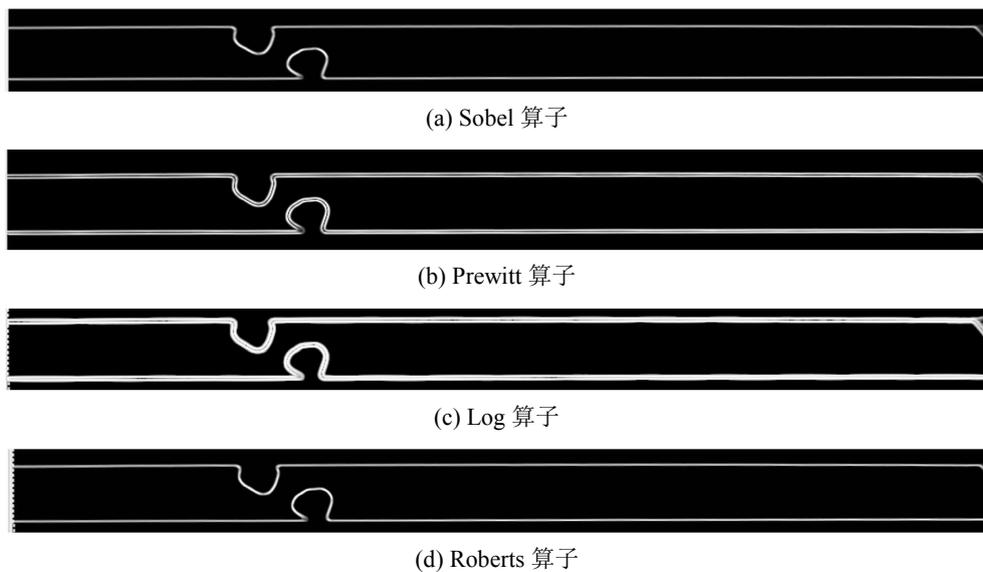


Figure 6. Rendering of different operators

图 6. 不同算子效果图

通过分析以上四种算子效果图可以看出, 简单快捷的 Sobel 算子表现良好, 呈现出的图像轮廓较为清晰, 在物体与边界连接处略有一点模糊不清。相比之下 Prewitt 算子的效果在清晰度上增强了很多, 不过由于其选中的轮廓为双线条, 会对物料的表面特征采取造成一定影响, 所以本文中并未采用 Prewitt 算子。Log 算子的表现直观上并不如 Prewitt 算子好, 采取的边缘特征轮廓不够清晰。最后一个 Roberts 算子所采集到的图像边缘特征并不如 Prewitt 呈现的那么好, 但是相较于 Sobel 算子略有一定的提升, 在某些边缘上没有出现断层等现象, 因此最终选择呈现效果较好的 Roberts 算子作为本实验边缘检测的手段。

3.2. 阈值分割

在对图像进行分割处理的过程中, 对想要提取的目标也不是唯一的, 这些目标各自具有自己的特征, 这些特征可能相同也可能不相同。图像分割的算法也有很大的差异, 有的根据区域间联系或者区域间的不同进行分割处理; 有的根据背景与目标的灰度值间差异进行分割。阈值分割便是前者的一种, 当一幅图片由较亮的背景和较暗的物体组成时, 便可以显著地将物体分割出来。其判断条件, 也就是所谓的阈值选取则是根据实验要分割的目标对象是否与其背景具有明显不同的灰度值。如若要分割的目标区域灰度与背景灰度极为相近, 则用阈值的设定就会比较棘手, 效果也会不尽如人意。因此, 阈值分割的关键所在便是确定选取阈值[12] [13]。

全阈值分割优势是算法简略速度快、计算量少, 但其对图片的灰度直方图要求也是很高的。如若在图像里有较多的干扰相如未处理掉的噪声过多等情况出现时, 全局阈值的确定就会变得比较困难。由于整张图像都根据这一个阈值来分割, 一旦出现偏差, 则会导致感兴趣的部分损失其部分特征或者又添入了不感兴趣的区域, 这将对实验后续的计数处理造成比较大的麻烦, 会导致实验结果出现偏差。

Auto_threshold 算子是自动全局阈值分割算子(效果图见图 7)。其运行原理也需要通过灰度直方图, 所以第一步便是计算出图像灰度直方图。第二步是对图像做高斯平滑处理, 对图像噪声等干扰因素做一个简单的处理后找出灰度直方图里的最小值。最后便是用这个最小值进行阈值分割。



Figure 7. auto_threshold operator segmentation result
图 7. auto_threshold 算子分割结果

全阈值分割针对整张图片进行分割, 该算法采用一个固定的阈值, 用该阈值对图像进行分割。该算法的优点是处理图片是较为直观, 可以清晰的看到处理后的效果, 其内部计算也相对简单, 因此它的计算速度很快可以节约很多处理时间。与此同时, 全阈值分割也有相应的缺点, 当一幅图像较为复杂并且图中的灰度分布既散乱又相差不大时, 其分割效果会大打折扣。

图像灰度值特征提取、计算分析在做图像分割处理时是必不可少的环节。阈值的确定选择与灰度值紧密相关。对双峰形的灰度直方图而言, 最恰当、合适的阈值边在其谷底获得。

Threshold 算子应用于全局阈值分割, 意味着其分割对象是整张图片。它对环境要求较高, 图片里的目标与背景的灰度差应比较明显。该算子通过灰度直方图进行阈值的选定。图 8 为经过预处理后的图像, 其灰度直方图如图 9。在图 9 中可以看到, 所感兴趣的物料的灰度值为 0~10, 而下落背景的灰度值大约在 80~150 之间。所以对于阈值的选取在 20~60 之间的任意一点就可以将物料从背景中上分离出来, 如图 10。

由图 7 可以看出自动阈值分割后会出许多干扰, 并不能将物料完全分割出来, 再看图 10 通过使用合适的全局阈值对图像进行确定阈值的全局分割, 则可以很好的将物料分离出来, 达到实验目的。



Figure 8. Preprocessed image

图 8. 预处理后的图像

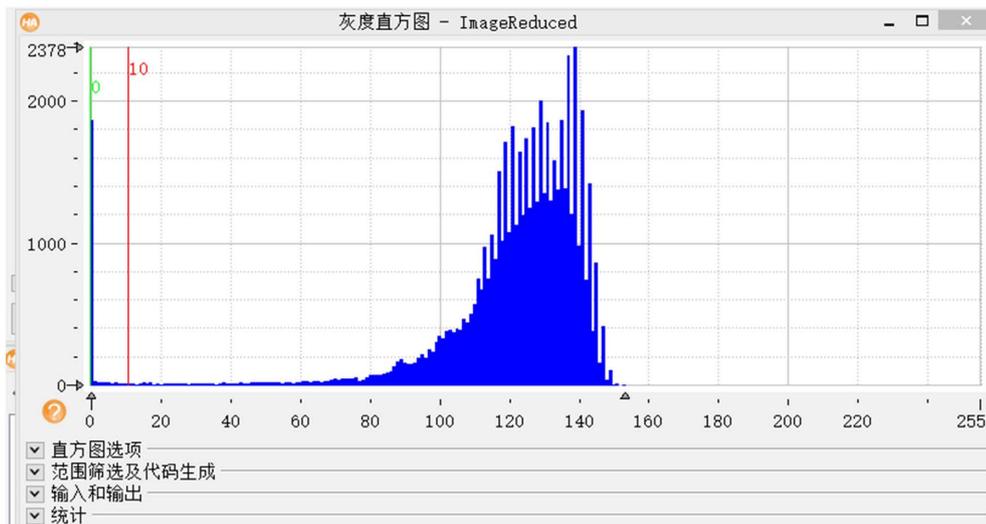


Figure 9. Gray histogram of falling background

图 9. 下落背景灰度直方图



Figure 10. The segmentation result of threshold operator ($T = 50$)

图 10. Threshold 算子的分割结果($T = 50$)

3.3. 图像形态学分析

图像在经过分割处理之后截取的部分在还需进行进一步计数。若是出现某两个或两个以上的物料在下落时紧挨在一起，在之后的计数过程里会被计算成一个。这将对后续的最终结果造成影响，若是紧挨在一起的物料较多的话，会导致产生较大的误差。而图像的形态学分析可以帮助解决这一棘手问题。

腐蚀和膨胀运算是形态学最简单、基础的算法，它们的应用范围也最为广泛。其中，腐蚀算法的主要功能是将通过分割截取出来的 ROI 的边界做一个收缩。腐蚀运算使其边界收缩会将两个或两个以上的物料在下落时紧挨在一起的物料分离。分离之后在进行计数运算就可减少最终的结果误差。

腐蚀运算的数学定义是使用结构元素与截取图像做与操作。得到它们差值的最小值。膨胀运算则与腐蚀相反。通过膨胀运算可以使图像变得更连续、平滑，膨胀也可使其边界包容背景的一部分，即将一部分背景内容划入自己内部。

在对下落的物料进行腐蚀运算后会发现：其原有的面积特征被改变了，腐蚀虽然将两个或两个以上的紧挨在一起的物料分开，但是它们都由边界向内部进行收缩，这就导致其在原图上的面积变小，进而影响到了后续的中心点坐标以及面积大小的确定。单独使用一个膨胀运算或者腐蚀运算会导致其自身出现特征被破坏，因此会在实际应用中用膨胀运算与腐蚀运算的组合来处理图像。基于上述考虑内容，提出了开运算与闭运算的算法。开运算是先将目标做腐蚀运算之后再膨胀运算，而闭运算则与之相反。

实验中对下落的物料采取开运算，先将其未处理的某部分干扰和多个紧挨在一起的物料使用腐蚀运

算处理好，在通过膨胀运算使其边界变得更平滑，且不明显改变物料的面积大小。图 11 是对下落物料进行处理后的结果。

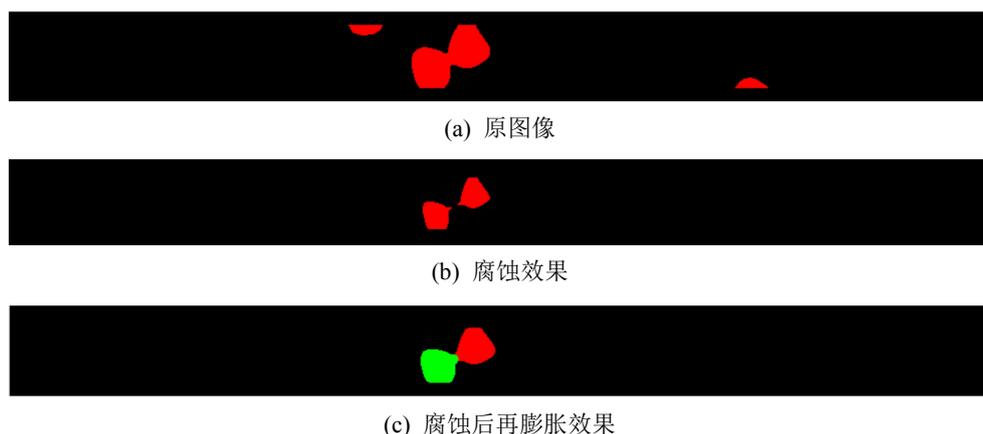


Figure 11. Morphological analysis results

图 11. 形态学分析结果

由图 11 可以发现，经由形态学分析后，有部分面积特征较小的物料会在腐蚀运算过程中被吞掉，因此在对图像腐蚀的时候之后在做计数会对最终统计物料数目造成一定的误差，因此需要调节 Halcon 腐蚀算子的腐蚀程度参数，经过多次调整之后选取一个最恰当的值。

4. 跟踪与计数算法

提取 ROI 之后就需要设计合适的算法对连续下落的物料进行计数。图像特征选择就是挑选对能够追踪到 ROI 的最为合适的追踪标准。针对下落的物料而言，在不同帧下看到的同一物料会具备不同的图像特征。在做图像特征的选择时，需要满足计算机计算量小、特征受其他因素干扰小的标准。图像提取的过程也被称作降维的过程，它能是本来处于高维特征的原始图像用低维特征空间来表示，从而实现降维的目的。要对下落的物料进行计数，对每一个物料的追踪是决定整个结果误差的关键所在[14]。

实验提出了基于物料中心点的追踪算法。由于摄像头是固定的，采集的照片为同一角度下拍摄的。因此可以联想到使用图像的横坐标值对图像进行追踪。多次实验后得出，当提取的某一物料中心点坐标值与其上一帧或下一帧图像里相比差值在一定区间内，则定义为该图像是已出现过的图像。由此设计出了以下跟踪算法：1) 通过循环逐个读取图像；2) 对读取的图像做预处理；3) 将预处理后的图像里的物料分割出来，并用 `area_center` 算子对所有物料的中心值横坐标予以记录，并存入数组 `c` 中；4) 将数组 `c` 中所存的中心点横坐标值与前一帧图像里所存的中心点横坐标一一比较，若有一个与前一帧图中所存数值的差值的绝对值在 7 以内，则判定为同一物体。如图 12 中(a)中红色的物料与图(b)中蓝色的物料，经上述的中心点坐标值追踪后判定为同一物体。

`Count_obj` 算子是用来计算输入区域中的连通域的个数。通过分割提取和形态学分析后，使用 `connection` 算子将仍然相切相邻的区域连接在一起后并使其变为连通区域，计算整个 ROI 中有几个单独的区域。使用 `count_obj` 算子时可能会碰到计算出来的区域数目比实际的数目要多 1。其原因有两点：`connection` 算子使用次数过多，导致其连通区域混乱，因此出现了计算偏差；黑色的背景(黑色底面)也算作了一个区域，导致最终结果多 1。实验中使用 `select_gary` 算子对其黑色底面区域进行去除，最终通过 `count_obj` 算子计算得出最终的结果。

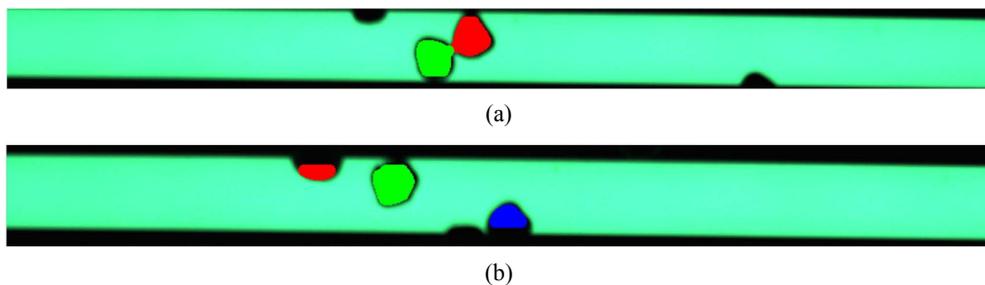


Figure 12. Tracking algorithm demonstration diagram
图 12. 追踪算法演示图

5. 实验结果

5.1. 图像分析

本实验中物料检测的图像分析是基于采集连续下落时的物料的认识而进行的，在经过一系列图像处理操作后，对提取的待分析结构进行下一步处理的过程，其主要步骤如下图 13。



Figure 13. Main steps of image analysis
图 13. 图像分析主要步骤

对于物料的分析是从图像分割开始。紧接着根据各物料关键的特征，将待分割的区域从图像背景(即下落背景)中选择出来。分割出来的图像不能直接用于统计分析，因为无法确定前后会不会出现相同的物料，所以需要对分割出来的区域进行区域追踪比较，处理之后在通过最终的统计分析，存储在最终结果。图 14 和图 15 分别展示了读取的原图像以及经过预处理后的图像。



Figure 14. Read image
图 14. 读取图像



Figure 15. Rotated and enhanced image
图 15. 旋转、增强后的图像

5.2. 物料区域检测算法

物料的区域检测算法流程图如图 16

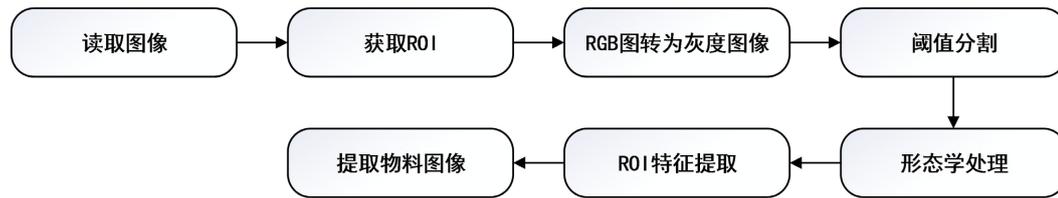


Figure 16. Flow chart of material area detection

图 16. 物料区域检测流程图

物料区域检测流程如下：首先对相机采集的原图像进行预处理，通过感兴趣区域提取、平滑滤波、对比度拉伸等操作之后将图像转换成灰度图像(如图 17)。在运用第三章所说的阈值分割算法，通过全局阈值分割将物料从下落背景中分离出来形成二值图像(如图 18)。在运用形态学分析里的先腐蚀后膨胀的开运算将其两个会两个以上紧挨在一起的物料分离开(如图 19)，并去除途中遗留的噪声干扰，最后通过 connection 算子将各区域形成连通区域。图 20 为所截取的物料图像在原图像上的标记，经过处理后所选中的物料图像与原图中的黑色物料图像基本重合，其面积轮廓等特征基本与原图一致。



Figure 17. Convert to grayscale image

图 17. 转换为灰度图像



Figure 18. Image after threshold segmentation

图 18. 阈值分割后的图像



Figure 19. The final material image after morphological processing

图 19. 形态学处理后最终得到的物料图像



Figure 20. Materials are marked in the original drawing

图 20. 物料在原图中标记

5.3. 数算法

将分析处理好的图像通过 connection 算子形成连通区域之后，便可使用 count_obj 算子对其数目进行计算。但单独使用 count_obj 算子时会将底色背景也算作一个 region 会导致最终结果出现偏差，因此在使

用 count_obj 算子之前先用 select_gray 算子将其底色背景过滤掉。由于整个图像的读取以及操作在一个大的迭代循环之下完成, 而使用 count_obj 算子计算出来的物料数是每帧图片单独的物料数目, 因此在统计最终的物料结果时, 需要通过一个判断条件来判别第 $n+1$ 帧的物料是否会出现第 n 帧或第 $n+2$ 帧里。具体流程如下图 21。图 22 为最终输出结果。

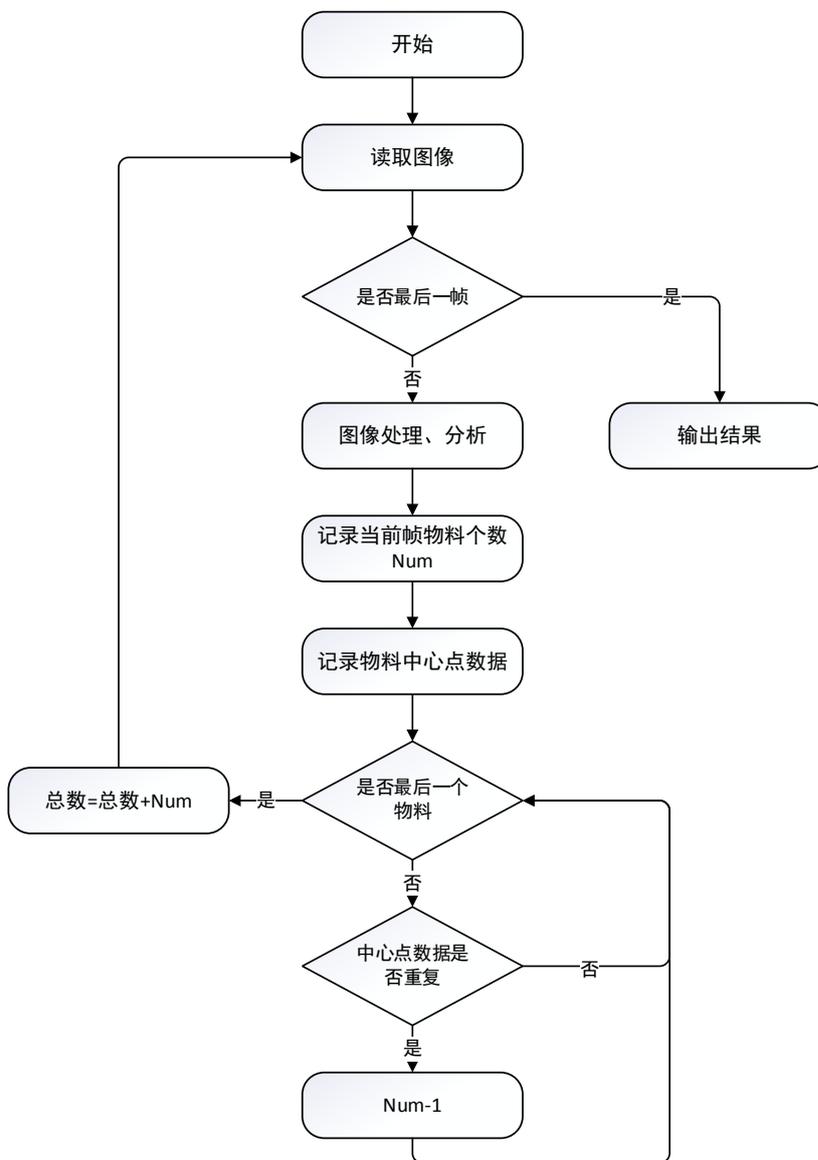


Figure 21. Flow chart of judgment counting
图 21. 判断计数流程图



Figure 22. Output results
图 22. 输出结果

Table 1. Data sheet of experimental results**表 1.** 实验结果数据表

参数	值
图像分辨率	1000 × 128
实际物料个数(个)	60
算法统计物料个数(个)	56
物料统计精度	88.89%
平均每帧消耗时间(s)	0.025

表 1 为实验的结果数据, 本次采集到的 178 张照片中一共有 60 个物料, 本算法统计出的物料为 56 个, 准确率为 93.33%, 平均每帧消耗时间为 0.025 秒, 算法总体耗时为 4.45 秒效率较高, 实验结果误差较小。由于采集照片存在一定的特殊性, 在不同的采集环境下需要对算法中的算子参数进行适当调节。因此本算法存在较大的局限性。

6. 总结

针对物料的检测、追踪和计数问题, 本实验提出了基于机器视觉的物料检测算法。目前, 在农业、工业、医药等众多生产业都会面临物料检测与计数这一操作流程。对人工而言, 其工作量是十分巨大的, 通过机器视觉对其计数, 则会节约人力资源, 也会降低人为因素导致的计数结果误差。本实验提出的算法可以准确处理特定角度下相机采取的下落物料图像, 包括对物料的检测显示以及之后的追踪和最终的计数。但由于实验的数据集有限, 对于不同相机采取的图像处理能力也有限。包括一些周边环境等因素的影响(如打光不均匀、曝光、灰度直方图分布复杂等)无法对所有的采集图像都进行很好的物料检测及追踪计数功能。本算法中依然存在一定的不足去应对新的、未知的采集图像。在遇到多角度不同环境不同质量的新数据集时需要调整算法中的参数值进行一定的调整, 这也是本实验提出算法的局限性所在。

参考文献

- [1] 叶青松. 基于机器视觉的工业检测研究——实时药丸检测系统[D]: [硕士学位论文]. 无锡: 江南大学, 2008.
- [2] 张红霞, 刘义才. 机器视觉技术的应用研究[J]. 电子世界, 2013(17): 106-108.
- [3] 贾伟, 刘炜, 刘彦宾, 王亚, 樊里略. 一种在线运动物料袋智能图像计数方法[J]. 包装工程, 2014, 35(21): 6-9+14.
- [4] 田齐, 张令. 基于机器视觉的晶圆裸片计数方法[J]. 传感器与微系统, 2018, 37(10): 41-43.
- [5] 郭俊, 郑堤, 陈俊华, 刘年福. 基于机器视觉的跟踪计数算法[J]. 传感器与微系统, 2018, 37(2): 154-157+160.
- [6] Panigrahy, C., Seal, A. and Mahato, N.K. (2020) Image Texture Surface Analysis Using an Improved Differential Box Counting Based Fractal Dimension. *Powder Technology*, **364**, 276-299. <https://doi.org/10.1016/j.powtec.2020.01.053>
- [7] Spetale, F.E., Murillo, J., Vazquez, D.V., Cacchiarelli, P., Rodríguez, G.R. and Tapia E. (2020) LocAnalyzer: A Computer Vision Method to Count Locules in Tomato Fruits. *Computers and Electronics in Agriculture*, **173**, 105382.
- [8] Fofanov, V.B. and Zhiznevskii, A.N. (2012) Segmentation of Regions of Interest on Locally Homogeneous Scenes. *Pattern Recognition and Image Analysis*, **22**, 257-264. <https://doi.org/10.1134/S1054661812020022>
- [9] 杨方方. 彩色图像分割技术的研究——图像边缘检测技术的研究应用[D]: [硕士学位论文]. 无锡: 江南大学, 2009.
- [10] 欧温暖. 几种常见边缘检测算子的分析比较[J]. 现代计算机(专业版), 2010(5): 75-77.
- [11] 吴一全, 孟天亮, 吴诗嫻. 图像阈值分割方法研究进展 20 年(1994-2014) [J]. 数据采集与处理, 2015, 30(1): 1-23.
- [12] 谭优, 王泽勇. 图像阈值分割算法实用技术与比较[J]. 微计算机信息, 2007, 23(24): 298-299+233.
- [13] 杨国修. 浅谈图像阈值分割技术[J]. 电子设计工程, 2012, 20(23): 36-37+40.
- [14] 潘锋. 特征提取与特征选择技术研究[D]: [博士学位论文]. 南京: 南京航空航天大学, 2011.