

基于机器视觉的羽毛分拣系统的设计与实现

金雨璇, 许宗华, 马兴录, 马圣洁

(青岛科技大学 信息科学技术学院, 山东 青岛 266061)

摘要: 在羽毛生产过程中, 洗净烘干后的羽毛中常掺杂杂色羽毛, 现有分拣方法效率低且人工筛选成本高, 制约了羽毛品质的提升; 针对上述问题, 对 ResNet-18、YOLOv5 和 OpenCV 等技术方案进行了实验对比及分析, 设计并实现了一个基于机器视觉的羽毛分拣系统; 详细介绍了单片机和继电器、电磁阀、气泵等硬件系统设计, 以及 HSV 提取、运动检测、颜色识别、数据通信、气泵控制和远程监控等软件模块的具体实现; 结合图像处理与机器视觉相关算法, 设计并实现了基于气泵控制的羽毛自动化分拣系统; 对该系统在正常室内环境下进行测试, 手持羽毛识别测试的分拣正确率在白天和傍晚分别达到 96% 和 90%; 自由下落羽毛识别测试的分拣正确率在白天和傍晚分别为 80% 和 74%。

关键词: 羽毛分拣; 图像处理; 机器视觉; ResNet; YOLO; OpenCV; 气泵控制; 自动化

Design and Implementation of a Feather Sorting System Based on Machine Vision

JIN Yuxuan, XU Zonghua, MA Xinglu, MA Shengjie

(Institute of Information Technology, Qingdao University of Science and Technology,
Qingdao 266061, China)

Abstract: In the process of feather production, after washing and drying, feathers are often mixed with colorful feathers. The existing manual sorting methods are the characteristics of low efficient and high labor costs, which restricts the quality of feathers. To address this issue, an experimental comparison and analysis of technologies such as ResNet-18, YOLOv5, and OpenCV were made. Based on the results, a machine vision-based feather sorting system was designed and implemented. The hardware modules, including microcontrollers, relays, solenoid valves, and air pumps, was detailed, along with the software modules such as HSV extraction, motion detection, color recognition, data communication, air pump control, and remote monitoring. By integrating image processing and machine vision algorithms, an automated feather sorting system based on the air pump control was developed. The system was tested under normal indoor lighting conditions, achieving sorting accuracy rates of 96% during the daytime and 90% in the evening for handheld feather recognition tests, and 80% during the daytime and 74% in the evening for free-fall feather recognition tests.

Keywords: feather sorting; image processing; machine vision; ResNet; YOLO; OpenCV; air pump control; automation

0 引言

目前, 羽毛在服装、家居用品、工艺品等行业中有着广泛的应用, 随着这些行业的不断发展, 羽毛的需求量也在不断增加。传统的羽毛加工方式通常依靠人工进行分类和分拣, 优点在于灵活性强, 能够处理复杂多变的羽毛形状, 且初期投入成本较低; 然而, 其缺点也十

分明显, 包括分拣效率低、劳动强度大、易受人为因素影响导致错误率较高。如果检验人员长时间进行操作, 很容易引起视觉疲劳^[1], 可能会导致同一样品的检测结果不尽相同。且难以应对大规模、高频次的分拣需求。在一些工艺品中, 对于羽毛的颜色有一定要求, 纯白色羽毛的需求量很高, 但羽毛经过洗净、烘干后, 里面会掺杂黑色羽毛 (一般 100 斤羽毛中大约有 5 斤左右的杂

收稿日期: 2025-03-04; 修回日期: 2025-04-22。

基金项目: 国家自然科学基金面上项目(22374086); 山东省 2023 年本科教学改革研究重点项目(Z2023152)。

作者简介: 金雨璇(2004-), 女, 大学本科。

通讯作者: 马圣洁(1994-), 女, 博士, 讲师。

引用格式: 金雨璇, 许宗华, 马兴录, 等. 基于机器视觉的羽毛分拣系统的设计与实现[J]. 计算机测量与控制, 2025, 33(9): 245-253.

色或黑色羽毛)。

在羽毛分拣技术的研究领域,国内外学者已取得了一系列进展。国内方面,早在 2004 年,郭晓峰和刘文田提出了一种基于图像处理的羽毛自动分拣系统,该系统通过图像采集和处理,实现了对羽毛的实时分拣^[2]。2018 年,袁海萍开发的旋转气流分拣仪器实现了羽毛与羽绒的高效分离^[3]。对于机器视觉,我国的发展相对起步较晚。2014 年,翟敬梅^[4]团队对视觉识别和检测系统进行了模块化处理,实现了机器人对目标的成功抓取。2023 年,刘红^[5]团队提出了一种基于机器视觉的集成电路板焊点缺陷识别方法,通过多种技术手段实现了焊点缺陷的准确识别,启发了羽毛微损伤检测的技术迁移。国外研究则相对领先,2012 年普利和巴克舍耶夫利^[6]用 OpenCV 实现了实时图像处理。2018 年格雷厄姆·D·芬利森^[7]在计算机视觉中的颜色和照明一文中证明最好的线性偏差校正涉及解决单应性问题。2020 年,Bochkovskiy 等人提出了 YOLOv4 算法,在保持高精度的同时显著提升了目标检测的速度,适用于实时应用场景^[8]。总体来看,机器视觉技术在羽毛分拣系统中的应用正逐步成为提升分拣效率和准确性的关键,国内外的研究成果为该领域的发展提供了重要的理论基础和技术支持。

随着科技的发展,自动化系统的设计及应用越来越广泛,使得实现羽毛的自动分拣变得可行^[9]。同时,人工智能及机器视觉技术正在不断发展进步。机器视觉通过对数字图像进行处理和分析,实现对目标的识别和解析,使得机器能够理解外界环境,从而更好地完成人机交互,使机器发挥人类眼睛的功能^[10]。从而促使基于机器视觉的实时图像处理能力、性能、模块化处理及准确性方面都取得了显著的进展。相比于传统人工分拣方法,自动化分拣系统有高效率、稳定性强和可持续性好的优点,能够在短时间内处理大量物品,减少人为错误,提高分拣精度,但在灵活性和成本控制方面仍需进一步优化。

针对上述问题,本文通过对于相关技术方案的分析比较^[11-12],设计并实现了一个基于机器视觉的羽毛分拣系统,能够自动化分拣出杂色羽毛。

自动化是羽毛分拣系统设计须具备的一个要素,其实现主要在于使用工控机通过机器视觉检测后,串口通信控制单片机所连接设计的气动分拣模块剔出杂色羽毛,达成对于羽毛的准确分拣。

为了提升开发效率、系统的可维护性和可扩展性,本系统采用了模块化设计理念。羽毛分拣系统的软件部分主要包括以下几个模块:HSV 提取模块、运动检测模块、颜色识别模块、数据通信模块、气泵控制模块、远程监控模块。

系统能够通过颜色识别算法和运动检测算法以及图像处理算法进行杂色羽毛识别,配合单片机、继电器、电磁阀和气泵的设计连接的自动化分拣出杂色羽毛。该系统不仅可以提高白色羽毛分拣的效率及准确率,也能够为不同应用场景提供可行的技术路线及系统的软硬件系统的设计及实现思路。

1 羽毛分拣系统关键技术分析

1.1 基于 ResNet-18 模型

ResNet 是卷积神经网络中最为经典的模型之一^[13],通过在网络中引入了残差模块,添加一个跳跃连接在输入和输出之间,使得网络可以更加轻松地学习残差,解决了梯度消失和模型退化的问题^[14]。ResNet-18 具有 18 层的网络深度,能够有效提取图像特征。

本次训练使用数据集由包含羽毛的正样本和不包含羽毛的负样本组成。每个正样本包含一张图像及其对应的 XML 标注文件,负样本只有图像。数据集按照 8:1:1 的比例划分为训练集、验证集和测试集。程序根据名称读取对应文件进行加载,然后对数据集进行预处理,将图像大小调整为 224×224 ,转换为 Tensor 格式并进行归一化。然后为训练集、验证集和测试集分别创建数据加载器,方便在训练和评估过程中批量加载数据。

完成数据预处理后,加载预训练好的 ResNet-18 模型,并替换其最后的全连接层以适应当前分类任务(两类:包含羽毛和不包含羽毛)。使用交叉熵损失函数和 Adam 优化器进行训练。模型训练时每个 epoch 包含训练和验证两个阶段。最后保存训练好的模型。并且绘制损失函数变化图和准确率变化图,训练结束后使用测试集进行测试。

本次训练共训练 40 轮,损失函数变化图和准确率变化图如图 1、图 2 所示。最后运行测试集的结果中,测试损失为 0.879 9,准确率为 0.917 3。这些结果表明 ResNet-18 的模型训练结果表现良好,具有较高的准确率。

上述结果表明 ResNet-18 在分类任务中表现良好,但在羽毛分拣系统的实现中,还需要精准定位羽毛的位置并识别羽毛的颜色。而且在实际应用中,实时性是一个关键因素。目标检测任务需要在实时视频流中快速识别和定位目标。针对实时目标检测任务来说,ResNet-18 就不是最优的选择了,它是针对图像分类而不是目标检测来进行优化的。因此,在进一步的研究中,研究学习了 YOLOv5 模型,该模型专门用于目标检测,并且在实时性方面拥有较好的效果。

1.2 基于 YOLO 模型

YOLO 是一种目标检测模型,具有实时检测的能

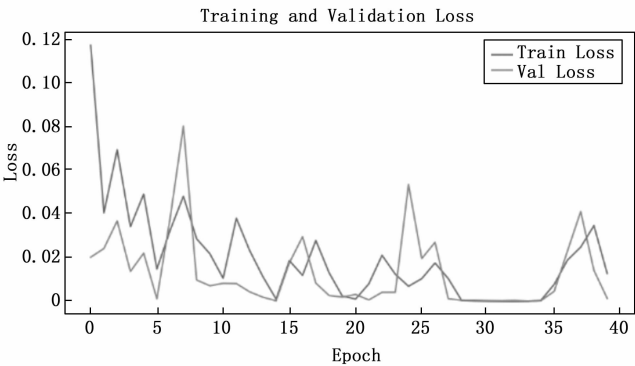


图 1 损失函数变化图

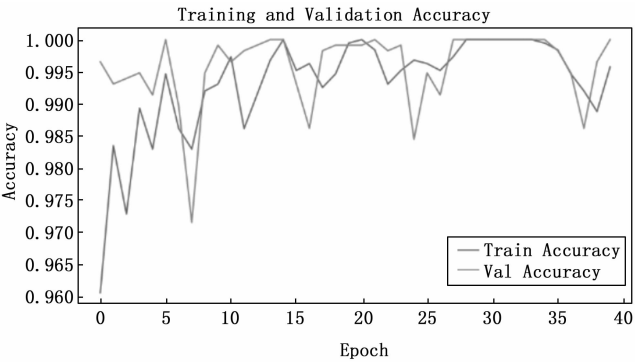


图 2 准确率变化图

力，能够在单张图片上检测多个目标，并显示其边界框和类别。

因为直接训练识别羽毛的颜色需要大量的数据集，所以采取了通过 YOLOv5 检测出羽毛，然后再调用 HSV 识别其颜色的方案进行分析验证。

HSV 是一种常见的颜色空间。颜色空间作为颜色的数学表示模型，是颜色特征提取的基础。想要更好的表达颜色信息，需要选择合适的颜色空间。常见的颜色

空间还有 RGB 颜色空间等。RGB 颜色空间通过红 (R)、蓝 (B)、绿 (G) 3 个通道改变和叠加来得到各种各样的颜色，是完全依据人眼识别的颜色而定义出来的颜色空间，可以表示出大部分颜色。而 HSV 颜色空间则是根据人的视觉感知即颜色的直观特性提出的一个六角锥体模型。这个模型中的参数分别是色调 (H)、饱和度 (S) 和亮度 (V)。

通过数据集的转换以及模型的训练，经过 100 轮训练后，结果如图 3 所示。

测试结果中，绝大多数 feather 置信度在 0.8 以上，识别效果良好。但是在使用实时摄像头识别测试时，视频有卡顿，效果欠佳。虽然 YOLOv5 在静态图像上的目标检测性能较好，但在实时检测过程中由于硬件资源及处理速度原因，视频不流畅，而且由于数据集较少，实时视频识别准确度也不如验证集识别准确度高，无法达到系统的准确性和实时性要求。可见，采用 YOLOv5 的技术方案对于硬件资源的匹配以及数据集资源都有较高要求。

1.3 基于 OpenCV 的传统机器视觉检测方法

第三种技术方案是基于 OpenCV 并采用运动检测与颜色识别结合的方法，采用运动检测算法实现对于下落羽毛的检测，绘制轮廓并且标定为感兴趣区域，然后对运动区域采用 HSV 颜色空间模型^[15-18]进行颜色识别，增加颜色识别的准确度和鲁棒性。运动检测和颜色识别的优缺点如表 1 所示。

验证分析过程中，通过背景差分的运动检测算法并绘制运动物体轮廓，以及编写 HSV 颜色识别代码，对于羽毛进行颜色识别，将运动识别检测到的羽毛区域转换为 HSV 颜色空间，设定不同颜色的阈值范围进行颜色识别，记录识别的准确率情况，实验检测结果如表 2 所示。

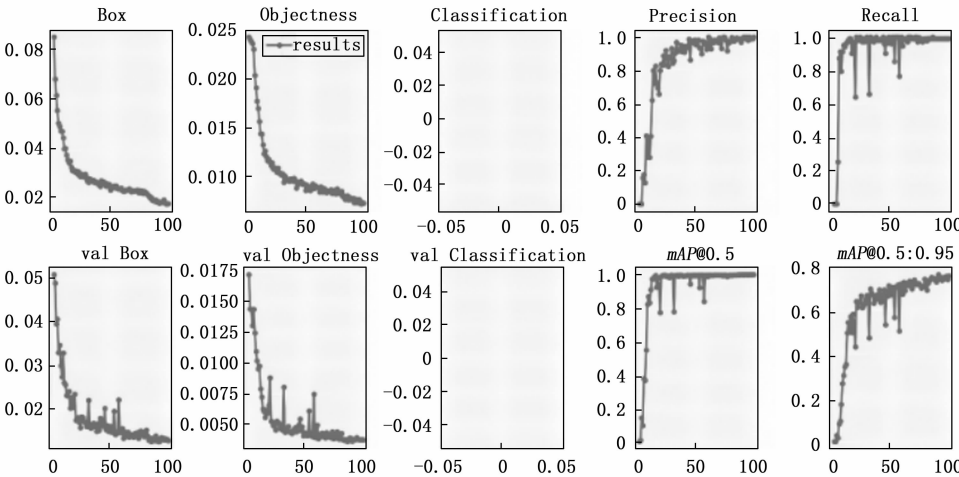


图 3 训练结果图

表 1 运动检测与颜色识别优缺点

算法名称	优点	缺点
运动检测	1) 实时性强, 适合实时处理。 2) 对于静态背景中的运动目标检测良好。	1) 对于背景复杂或变化较大的场景效果不佳。 2) 需要进行背景建模。
HSV 颜色识别	1) HSV 颜色空间更符合人类对颜色的感知, 且对光照不敏感。 2) 颜色阈值分割简单计算复杂度低。	1) HSV 转换过程中可能会导致颜色信息损失。 2) 对于颜色相近或者分布复杂的目标检测效果不理想。

表 2 实验检测结果

算法名称	识别准确率
运动检测	静态背景: 95% 以上 变化背景: 检测率下降
HSV 颜色识别	白天: 85% 以上 傍晚: 70% 左右

实验结果表明, 基于运动检测和 HSV 颜色识别的方法在静态背景和光照稳定的条件下具有较高的检测和识别准确率。但在背景复杂和光照变化的场景中, 检测和识别效果有所下降。

针对以上结果的准确率问题, 对运动检测识别算法进行了优化改进。原先实验使用的背景差分法检测效果受光线影响较大, 选择使用高斯混合模型的背景差分法和帧差法^[19]相结合的方式。使用 MOG2 背景减除器检测前景物体, 并使用中值滤波去除噪点, 调用帧差法对连续帧进行灰度处理和高斯模糊处理, 然后计算帧差并进行二值处理, 将结果与背景减除的结果取或相结合, 获得更加可靠的前景掩膜, 最后使用闭运算填补空洞并去除噪点, 让前景区域连通。

对于颜色识别, 在原先直接识别基础上添加了形态学闭运算对掩膜进行消除小孔洞和噪点, 然后再对运动检测绘制的轮廓内的每一个像素点进行 HSV 颜色识别, 识别出下落羽毛的颜色。

1.4 方案对比与选择

通过实验对比结果如表 3 所示。经分析本文选用基于传统方法的羽毛检测方案进行设计与实现。

2 系统设计与实现

2.1 系统总体架构

本羽毛分拣系统通过工业控制计算机连接 USB 摄像头和机器视觉进行颜色识别, 分拣采用瞬发气泵吹出杂色羽毛实现。羽毛分拣系统的系统架构如图 4 所示, 系统主要由识别部分、分拣部分和远程监控部分组成。各部分完成羽毛检测及分拣过程如图 5 所示。

表 3 实验对比结果

	ResNet-18 模型	YOLOv5 模型	传统机器视觉方法
优点	1. 分类任务准确率高 2. 网络结构经典	1. 实时目标检测能力强 2. 可多目标识别 3. 边界框定位准确	1. 实时性强 2. 稳定性高 3. 成本低 4. 无需复杂模型训练和大量的数据集 5. 易于在嵌入式系统中实现
缺点	1. 无法进行目标检测与定位 2. 不满足实时性要求	1. 实时视频检测流畅度低 2. 准确率受数据集规模限制 3. 计算资源需求高	1. 受背景复杂度和光照变化影响大 2. 颜色识别对阈值依赖性强
适用性	仅适用于图像分类任务	适用于硬件资源充足且数据充足的场景	适用于静态背景、光照稳定的场景
最终选择结果	不选择, 功能不匹配	不选择, 实时性与资源消耗不达标	选择, 成本低、实时性强、满足基础准确率需求

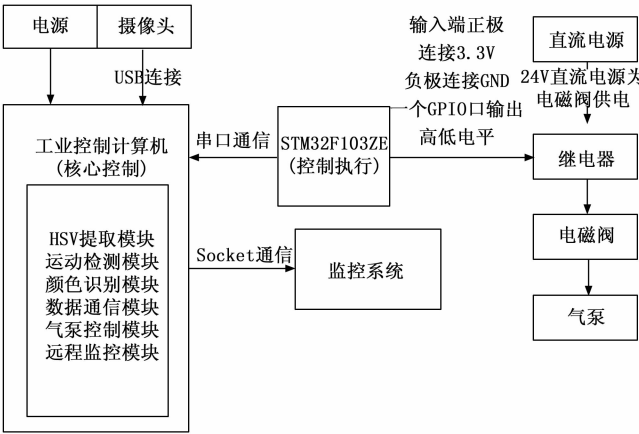


图 4 系统架构图

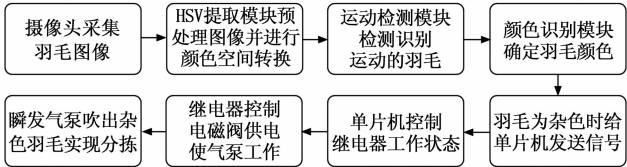


图 5 羽毛检测及分拣过程

1) 识别部分包括工控机连接 USB 摄像头用于采集图像和实时检测识别, 实时显示当前检测结果, 并且在羽毛为杂色羽毛时, 给单片机发送信号。

2) 分拣部分主要包括 STM32F103ZE 单片机^[20], 继电器, 电磁阀和其所需的直流电源, 气泵。分拣部分接收到计算机发送的信号后, 通过 GPIO 控制继电器工

作状态, 继电器控制电磁阀供电, 使气泵工作, 改变羽毛下落方向, 实现羽毛分拣。

3) 远程监控部分包括在工控机端识别部分进行服务器端创建, 在局域网内的计算机比如管理室的监控系统进行客户端创建, 通过传输处理后的视频流, 实现在客户端远程监控识别处理效果。

2.2 硬件架构设计

在本系统中硬件主要使用了工控机、USB 摄像头、单片机、继电器、电磁阀、气泵等。硬件架构如图 6 所示。

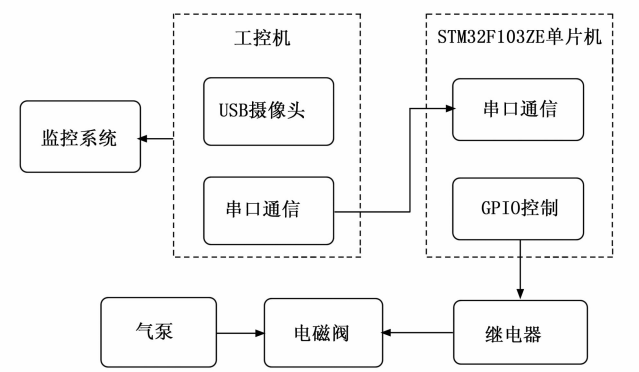


图 6 硬件架构图

单片机型号为正点原子战舰 V4 开发板, 采用 STM32F103ZET6 主控芯片, 该芯片的主频可达 72 MHz, 具备高运算速度, 适合实时控制羽毛分拣过程中的各项任务。此外, 它拥有丰富的外设接口, 能够方便地与各种传感器和执行器连接, 满足系统对数据采集和控制的需求。低功耗特性使其适合长时间运行, 确保系统的稳定性。开发环境友好, 便于快速调试和迭代, 适应项目开发的需求。继电器用于控制电磁阀的通电与否, 达成开关效果。本系统选用的继电器为 3.3 V 高低一路光耦隔离继电器, 通过跳线设置高低电平触发, 能够灵活控制电磁阀的开关状态, 确保羽毛分拣过程中的精确控制。电磁阀选用一个型号为 4V210-08 的二位五通电磁阀, 使用电磁阀作为气泵开关, 堵住 A 口, 使用 B 口作为吹出气体口, 达到通电控制吹出羽毛的效果。其设计使得在通电和断电状态下能够快速切换工作口, 适应羽毛分拣过程中对气流的快速响应需求。气泵使用 600 W, 容积为 9L 的气泵, 通电打开开关后压缩气体存入底缸, 开启气口阀门输出气体。其强大的气体压缩能力能够提供稳定的气流, 满足羽毛分拣过程中对气体流量的要求。

各硬件器件的连接如图 7 所示。其中, 电脑端通过 USB 转串口连接 STM32 单片机进行通信, 分拣控制部分由单片机的 3.3V 的 VCC 端连接继电器的 DC+ 端口, GND 连接继电器的 DC-端口实现继电器供电,

PA0 连接继电器的 IN 端口实现高低电平控制其工作状态; 继电器的常开端口 NO 连接电磁阀所需直流电源的正极, COM 端口连接电磁阀的正极, 直流电源的负极直接与电磁阀负极相连, 实现继电器控制电磁阀供电; 气泵的气管连接电磁阀的进气口, 使用折管堵住电磁阀的 A 口也就是未通电下的工作位置出气口, 实现供电 B 口出气, 不供电没有气流输出。从而控制气流输出分拣出杂色羽毛。

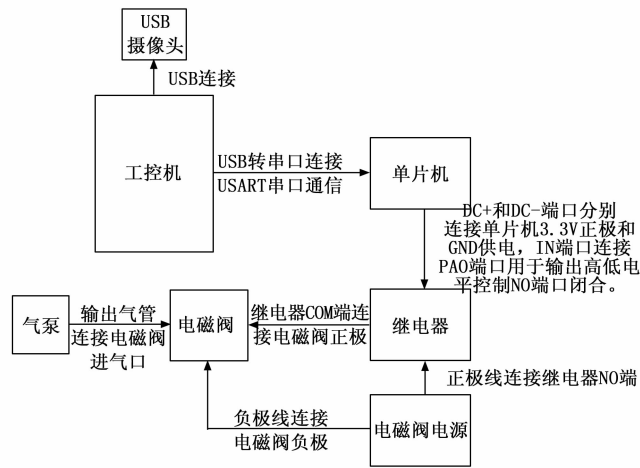


图 7 硬件连接图

2.3 软件架构设计与实现

羽毛分拣系统的软件部分主要包括 HSV 提取模块、运动检测模块、颜色识别模块、数据通信模块和气泵控制模块以及视频监控模块等, 软件设计的架构如图 8 所示。

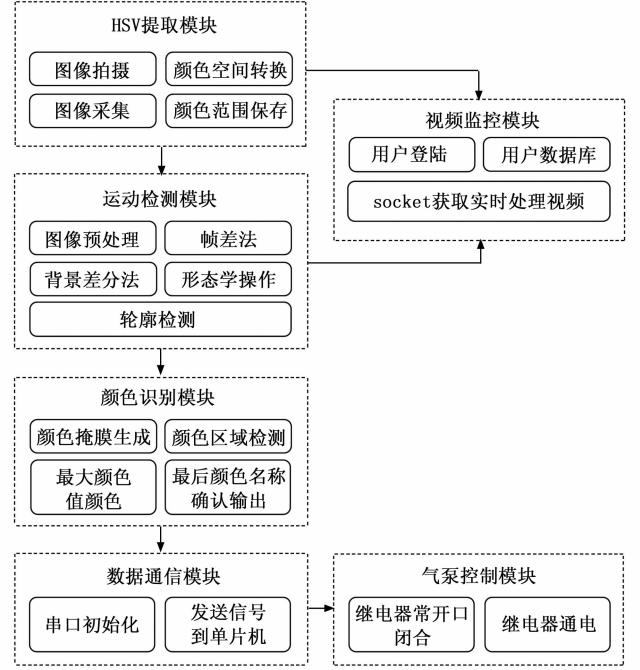


图 8 软件设计架构图

HSV 提取模块首先处理图像并提取 HSV 信息，然后将处理结果传递给运动检测模块。运动检测模块通过背景减除法和帧差法识别运动区域后，将结果传递给颜色识别模块。颜色识别模块根据预定义的颜色范围进行分类，将识别到的颜色信息通过数据通信模块发送给气泵控制模块。气泵控制模块根据接收到的颜色信息，通过 GPIO 接口控制继电器和电磁阀，实现气流的精确控制。视频监控模块同步传输分拣画面至客户端，实现闭环监控。

HSV 提取模块主要负责图像预处理和 HSV 颜色空间转换^[21]。提取模块包括两个部分，首先使用当前的 USB 摄像头拍摄图像照片并保存，然后对于保存的图像进行处理和提取 HSV。下面是处理和提取 HSV 的详细描述。

首先，选取并读取文件夹中摄像头捕获的图像，先使用双边滤波对其进行边缘保留去噪。它能够在有效处理图像中的噪声并且保持图像的边缘信息完整。接下来，应用对比度受限的自适应直方图均衡化（CLAHE）增强图像的对比度，使得图像在不同光照条件下依然能够保持较好的可识别性。然后，将预处理后的图像转换为 HSV 颜色空间，HSV 颜色空间对于颜色的分割和识别更为直观和有效。最后，部署调整阈值，并创建掩码，根据掩码提取图像中的特定颜色区域，实现特定颜色区域的 HSV 阈值提取，并将其保存在文件中。HSV 提取的关键代码如下所示：

```
def preprocess_image(image):
    # 对图像应用边缘保留去噪
    denoised_image = cv2.bilateralFilter(image, 9, 75, 75)
    # 应用 CLAHE 增强对比度
    lab = cv2.cvtColor(denoised_image, cv2.COLOR_BGR2LAB)
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    lab[:, :, 0] = clahe.apply(lab[:, :, 0])
    enhanced_image = cv2.cvtColor(lab, cv2.COLOR_LAB2BGR)
    return enhanced_image

# 将预处理的图像转换为 HSV 空间
hsv_image = cv2.cvtColor(processed_image, cv2.COLOR_BGR2HSV)
# 获取当前 trackbars 的 HSV 阈值
hsv_low, hsv_high = get_hsv_thresholds(window_name)
# 根据 HSV 创建掩码
mask = cv2.inRange(hsv_image, hsv_low, hsv_high)
# 应用掩码提取图像中的特定颜色区域
result = cv2.bitwise_and(processed_image, processed_image, mask=mask)
```

显示结果图像

```
cv2.imshow(window_name, result)
```

运动检测模块主要负责检测识别图像中的运动的物体，系统采用背景减除法和帧差法相结合的方法，背景减除法使用高斯混合模型进行背景建模，从而提取出前景中的运动区域。再使用帧差法通过计算当前帧与前一帧的差异来检测运动^[22]。将这两种方法结合起来，能够在不同的场景和光照条件下有效地检测运动。检测到的前景区域再经过形态学处理，去除噪声并填充空洞，最后使用轮廓检测算法识别出运动区域的轮廓。相关实现代码如下所示：

初始化背景减除器

```
back_sub = cv2.createBackgroundSubtractorMOG2(detectShadows=False)
```

上一帧灰度图,用于帧差法

```
last_frame = None
```

背景减除获取前景掩模

```
fg_mask = back_sub.apply(frame)
```

```
fg_mask = cv2.medianBlur(fg_mask, 5) # 中值滤波去噪点
```

使用帧差法

```
frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
frame_gray = cv2.GaussianBlur(frame_gray, (21, 21), 0)
```

if last_frame is not None:

```
frame_delta = cv2.absdiff(last_frame, frame_gray)
```

```
thresh = cv2.threshold(frame_delta, 25, 255, cv2.THRESH_BINARY)[1]
```

```
fg_mask = cv2.bitwise_or(fg_mask, thresh)
```

```
last_frame = frame_gray
```

颜色识别模块的功能是根据检测到的运动区域，识别特定的颜色并进行分类。这一模块利用 HSV 颜色空间的优点，通过用户预定义的颜色范围进行颜色识别。在识别过程中，该模块使用了一系列图像处理技术，包括颜色掩码生成和形态学处理等，以提高颜色识别的准确率，并且能够在每一个羽毛下落离开后两帧输出所有检测值中最大的颜色，便于区分含有多种颜色的羽毛。具体代码如下：

颜色识别

```
hsv_roi = cv2.cvtColor(masked_roi, cv2.COLOR_BGR2HSV)
```

```
for color_info in custom_color_ranges:
```

```
color_name, lower, upper = color_info['name'], np.array(color_info['lower']), np.array(color_info['upper'])
```

```
color_mask = cv2.inRange(hsv_roi, lower, upper)
```

```
color_mask = cv2.morphologyEx(color_mask, cv2.MORPH_CLOSE, kernel) # 图像闭运算
```

```
hsv_sum = np.sum(color_mask)
```

数据通信模块采用的是 USART 串口通信，用于将识别到的颜色信息传递给单片机，方便单片机进行气泵控制操作。

在工控机端，初始化串口，并在颜色识别后通过串

口发送信号。在单片机端，通过 USART_Config 函数配置 USART 的 GPIO 引脚模式、波特率、数据长度、停止位、校验位和工作模式，并使能 USART。最后在主程序中初始化 USART 和 GPIO。在无限循环中等待接收串口数据，当接收到的数据大于 0 时，闭合继电器（设置 GPIO 引脚为高电平），延时 2 秒后断开继电器（设置 GPIO 引脚为低电平）。数据通信的两端部分代码如下所示：

```
# 工控机端
ser = initialize_serial()

if ser and best_color_name != "0": # 如果不是白色且串口可用
    ser.write(b'1')

# 单片机端
USART_Config(); # 初始化串口
int received_char = fgetc(stdin); # 等待接收数据
```

气泵控制模块主要负责在电路正确连接后，通过 GPIO 接口发送高低电平信号进而控制继电器的 NO 口的闭路与开路，继电器是作为电磁阀的供电开关的，进而控制电磁阀通电进入另一种工作位置，连接进气口的气泵在这时通过电磁阀 B 口喷出气流，实现气泵控制。具体实现如下：

```
# 设置 GPIO 引脚为高电平(继电器闭合)
GPIO_SetBits(RELAY_PORT, RELAY_PIN);
# 设置 GPIO 引脚为低电平(继电器断开)
GPIO_ResetBits(RELAY_PORT, RELAY_PIN);
```

视频监控模块主要分为服务器端视频流传输、客户端接受视频流和客户端登录 3 个部分。主要实现如下：

首先，在识别部分的程序中编写部署一个服务器端，通过 TCP Socket 通信，将实时处理后的视频流帧数据使用 pickle 模块进行序列化，然后使用 struct 模块打包成一个数据包发送到客户端。

其次，客户端在接受视频流数据后对其解码并在用户登录后显示，实现客户端实时监控的功能。

最后，为保证人机交互和监控查看的安全性，设置客户端登陆功能和一个保存用户信息的 sqlite 数据库用于登录操作。用户可以在登录窗口输入已存储在数据库中的管理员或者普通用户的用户名和密码进行登录，成功后可以对服务器端进行连接，失败后提示输入错误；设定第一个创建的用户为管理员，管理员可以进行注册和删除用户，注册和删除时需要输入管理员用户名和密码进行确认，而且还会将注册和删除的用户名与数据库内进行比对，保证用户名的唯一性和只删除存在的用户。

关键实现代码如下所示：

```
# 服务器端发送数据
data = pickle.dumps(frame)
packet = struct.pack("L", len(data)) + data
```

```
for sock in client_sockets:
    try:
        sock.sendall(packet)

# 客户端接收数据
# 反序列化帧数据并显示
frame = pickle.loads(frame_data)
cv2.imshow("Received Frame", frame)
```

3 测试及验证

3.1 硬件测试

对于选择的硬件进行通电测试，观察功能是否可以正常使用。具体测试如表 4 所示。

表 4 硬件测试

序号	测试名称	测试步骤与方法	预测结果	测试结果
1	单片机测试	1. 使用 ST-Link 烧录 USART1 通信程序（波特率 115 200） 2. 通过 CH340G 模块连接 PC 端 3. 发送 "AT + SN = STM32F103ZET6" 指令验证回传	电脑端通过串口调试程序与单片机通信发送接收数据正常	符合预期
2	继电器测试	1. DC+ 接 3.3 V, DC- 接地 2. PA0 输出 PWM 信号 3. 万用表测量触点导通电阻	继电器供电正常, PA0 输出电平转换后 NO 端口闭合	符合预期
3	电磁阀测试	1. 将所需直流电源直接连接到正负极 2. 进气口连接气泵检查供电后工作位置是否变化	工作位置变换, 通电后 B 口出气	符合预期
4	USB 摄像头测试	1. 连接电脑端打开相机 2. 检查成像是否正常	摄像头成像正常	符合预期
5	气泵测试	1. 通电开启气泵 2. 检查压缩储气是否正常	打开放气口有气流喷出	符合预期

3.2 功能测试

运动检测模块测试方面，使用不同背景和光照条件下的视频流测试运动检测模块。在白天和傍晚分别测试，测试摄像头能否识别不同光照条件下的运动物体并绘制出轮廓。其中运动识别效果相对最差的为傍晚的白色羽毛运动检测，因为与墙壁颜色几乎一致，但也能识别出来。傍晚的白色羽毛运动检测和颜色识别效果如图 9 所示。

颜色识别模块测试方面，使用各种颜色的羽毛进行测试，确保系统能准确识别预设的颜色范围。

白色名称设置为 0，杂色羽毛多为黄棕色名称设置

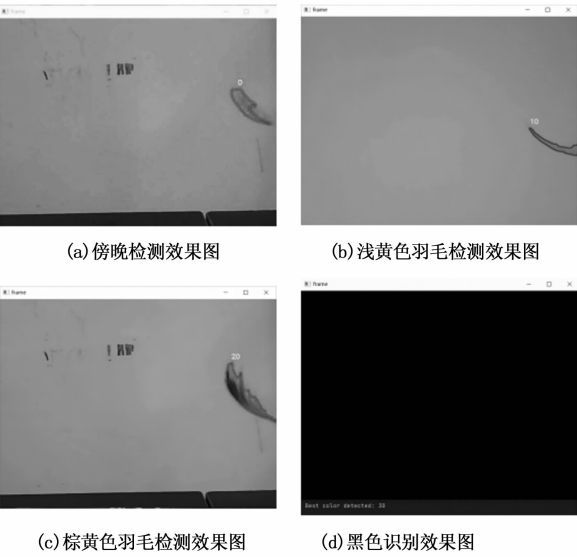


图 9 实验检测效果图

为 20，稍微泛黄色名称设置为 10，黑色名称设置为 30，识别效果良好。图 9 (b)、9 (c)、9 (d) 分别为浅黄色和棕黄色羽毛以及黑色的识别效果。

为了验证提出的方法的羽毛检测识别准确率，按照不同的光照条件以及样本运动方式提出两组对比实验。

1) 光照条件分为白天正常室内及傍晚室内两种情况，对于每种光照条件分别进行两组测量，每一组测量次数为 300 次，统计羽毛识别准确率。具体测量结果如表 5 所示。

表 5 分组对照测试—光照条件对羽毛识别准确率影响的测试结果

实验组别	光照条件	运动方式	测试次数	识别准确率	改善后准确率
1	白天正常室内	手持羽毛	300	96 %	—
		自由下落	300	75 %	80 %
2	傍晚室内	手持羽毛	300	90 %	—
		自由下落	300	67 %	74 %

具体对照测试结果分析如下：
从测试结果可以看出，在其他条件同等的情况下，白天正常室内条件下的识别准确率都高于傍晚室内的识别准确率，这主要是在傍晚的环境下，光照条件的变化对于泛黄色影响较大，光线较暗时会识别为白色。

2) 不同运动方式分为白天正常室内及傍晚室内两种情况，对于每种运动方式分别进行两组测量，每一组测量次数为 300 次，统计羽毛识别准确率。具体测量结果如表 6 所示。

具体对照测试结果分析如下：
从测试结果可以看出，在同等光照条件下，自由下落情况下的识别准确率相对于手持情况大大下降，根本原因是自由下落的速度原因导致运动模糊，浅黄几乎全

表 6 分组对照测试—运动方式对羽毛识别准确率影响的测试结果

实验组别	光照条件	运动方式	测试次数	识别准确率/%	改善后准确率/%
1	手持羽毛	白天正常室内	300	96	—
		傍晚室内	300	90	—
2	自由下落	白天正常室内	300	75	80
		傍晚室内	300	67	74

部识别为白色，白天正常室内通过调高帧率和曝光时间可使结果从 75 %改善至 80 %。傍晚光线较暗时会让运动模糊更加凸显，通过调节摄像头亮度可将结果从 67 %改善至 74 %。

实验结果表明，在复杂背景下也能检测出运动物体并绘制轮廓，运动检测模块功能完备，效果良好。

3.3 性能测试

性能测试的目的是评估系统的响应速度和处理能力。下面是本测试的具体模块测试描述。

1) 处理速度测试：测试系统处理每帧图像的时间，确保系统能实时处理视频流。在代码中加入计时功能，使用 `time.perf_counter()` 函数对处理每帧图像的时间进行测试输出。

2) 资源占用测试：测试系统运行时的 CPU 和内存占用情况，确保系统在资源受限的环境下能够稳定运行。本测试使用 `psutil` 库结合 Python 脚本来监控系统运行时的 CPU 和内存占用情况。

从测试的结果来看，系统在处理每帧图像时的性能表现非常好。大多数每帧的处理时间在 0.013 5 秒到 0.024 5 秒之间。这表明系统的处理速度相对稳定且快速，可以在 40 到 70 帧/秒的速度下处理图像，处理效率非常不错。CPU 使用率显示为 0.00 %，CPU 负载较低，系统资源利用情况良好。内存使用量稳定在 131.74 MB 到 134.48 MB 之间，变化幅度不大，说明内存使用比较稳定，没有明显的内存泄漏现象。

3.4 可靠性测试

可靠性测试的目的是验证系统的容错能力以及在长时间运行和恶劣环境下的运行稳定性。下面是本测试具体描述。

1) 容错能力测试：不插入 USB 转串口连接，测试程序运行后是否直接崩溃。

程序加入了检测串口函数，如果没有检测到则提示用户“没有检测到串口连接”，可以继续选择运行而不调用串口通信。具有一定容错能力。

2) 长时间运行测试：让系统连续运行一段时间，观察系统是否会出现崩溃或性能下降的问题。

运行测试成功，系统运行 12 小时后仍然正常执行功能。

3) 环境适应性测试:在不同光照条件下测试系统,确保系统在各种环境下都能稳定运行。

在白天和傍晚测试,功能均成功运行使用,工作较稳定。

综上所述,系统在处理每帧图像时的性能非常高,CPU和内存的使用都在可接受的范围内,可以满足实时检测和分拣的需求。

4 结束语

围绕羽毛分拣的需求及应用场景,本文系统分析及比较了 ResNet、YOLO 以及 OpenCV 等不同实现方案,并设计实现了一个基于 OpenCV 结合运动检测及颜色识别的羽毛分拣系统,包含 HSV 提取,运动检测、颜色识别,数据通信,气泵控制和视频监控等模块。其中,HSV 提取模块主要通过拍摄当前 USB 摄像头下的大量羽毛的图片,进行颜色 HSV 的提取,反复实验测试以增加羽毛颜色识别的准确性。运动检测模块则通过背景减法法和帧差法相结合,实时检测视频帧中的运动物体,准确识别羽毛的运动轨迹。颜色识别模块通过使用 HSV 颜色空间的阈值处理技术,准确识别羽毛的颜色,实现了对不同颜色羽毛的分拣。数据通信模块通过串口通信与单片机进行数据传输,控制气泵的工作,实现了自动化的羽毛分拣。气泵控制模块通过改变连接继电器 IN 端 GPIO 的高低电平,控制电磁阀通电,进而实现控制连接电磁阀的气泵,实现羽毛的分拣。视频监控模块通过 Socket 通信传输,实现远程监控的功能。

系统经过硬件测试、功能测试、性能测试和可靠性测试,验证了其在不同环境下的稳定性和准确性。通过算法优化和系统集成优化,提升了系统的处理效率和响应速度。

在本系统设计及实现的基础上,后续可通过硬件设备的升级与扩展、羽毛多维度特征识别算法的实现以及深度学习模型的结合,进一步对系统进行扩展和优化。

参考文献:

- [1] 张永革,陈春侠,陈洁,等.含绒量快速测试装置及其方法[J].毛纺科技,2016,44(4):57-59.
- [2] 郭晓锋,刘文田.基于图像处理的羽毛自动分拣系统[J].现代电子技术,2004(3):30-31.
- [3] 袁海萍.羽毛羽绒快速分拣仪器的开发及其使用[J].中国纤检,2018(2):83-85.
- [4] 翟敬梅,董鹏飞,张铁.基于视觉引导的工业机器人定位抓取系统设计[J].机械设计与研究,2014,30(5):45-49.
- [5] 刘红,杨奇,陈莉.基于机器视觉的集成电路板焊点缺陷识别方法[J].焊接技术,2023,52(6):93-97.
- [6] KARI PULLI, ANATOLY BAKSHEEV, KIRILL KORNYSKOV, et al. Real-time computer vision with OpenCV[J]. Communications of the ACM, 2012, 55(6): 61-69.
- [7] D G F. Colour and illumination in computer vision. [J]. Interface Focus, 2018, 8(4): 20180008.
- [8] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: Optimal speed and accuracy of object detection [Z]. 2020.
- [9] 秦珂.基于机器视觉的羽毛片分割和质量检测技术研究[D].南京:南京大学,2018.
- [10] DAVIES E R. Computer and machine vision theory, algorithms, practicalities (fourth edition) [M]. USA: Academic Press, 2012.
- [11] 卢嫚,朱世博.基于 OpenCV 和 YOLOv5 的车道线检测与识别[J].国外电子测量技术,2024(6):134-142.
- [12] 尹俊超,刘直芳.基于 OpenCV 的运动目标检测与跟踪[J].计算机工程与设计,2011(8):2817-2820.
- [13] YANG Y, ZHU Y. Based on serial port class implements of SCM and PC serial communication program design [J]. Advanced Materials Research, 2012, 629(629-629): 667-670.
- [14] 金张根,曹杨,于红绯,等.基于 ResNet-18 网络的城市生活垃圾识别方法研究[J].现代计算机,2023,29(2):73-77.
- [15] 陈梅芬,吴佳骏.基于 STM32F103 的颜色及形状识别系统设计[J].山西电子技术,2023(5):26-29.
- [16] QUAN Z C, WEI L, NAN Z R, et al. Color recognition of video object based on HSV model [J]. Applied Mechanics and Materials, 2011, 143-144(143-144): 721-725.
- [17] 韦琪.基于颜色与形状特征融合的物体识别方法研究[D].长春:东北师范大学,2019.
- [18] SUSANTO A, A S, H Z D, et al. Shallot quality classification using HSV color models and size identification based on naive bayes classifier [J]. Journal of Physics: Conference Series, 2020, 1577(1): 012020.
- [19] 白向峰,李艾华,蔡艳平,等.复杂场景实时目标检测方法[J].计算机辅助设计与图形学学报,2012(1):104-111.
- [20] 张海超,张北伟.基于 STM32 的多串口通信系统设计[J].国外电子测量技术,2019,38(2):99-102.
- [21] 侯宾,张文志,戴源成,等.基于 OpenCV 的目标物体颜色及轮廓的识别方法[J].现代电子技术,2014,37(24):76-79.
- [22] QIAOZHI T, ZHONGQIANG D, SHENG C. Moving target detection based on background modeling and frame difference [J]. Procedia Computer Science, 2023, 221585-592.