

南京理工大学泰州科技学院

毕业设计说明书(论文)

作者: 诸葛胜结 学号: 1901050252

学院(系): 电子电气工程学院

专业: 电气工程及其自动化

题目: 基于机器视觉的肉类新鲜度数据采集装

置设计

指导者: 刘超 副教授

评阅者: 刘磊 讲师

2023 年 5 月

毕业设计说明书（论文）中文摘要

当使用人工检测肉类品质，不仅效率低，且由于客观原因，人工检测存在很大的缺陷，难以保证获得准确有效的识别效果。总所周知，机器学习具有快速，无损，便捷，高度智能化的特点，在检测前期的准备工作时，采用单片机控制图像采集装置获取大量试验样本图像，采用模块化代码实现，操作简单，易上手，行业契合度高，可以进行广泛运用。

在本文中，说明了如何将机器视觉应用于基于图像检测实际应用。设计了一种采用机器视觉的新型的肉类的鲜度评估系统。主要设计思路为，先用单片机控制图像采集装置，用摄像头采集到图片后，送入到计算机中，然后进行图像的预处理后，用神经网络框架进行识别，并输出可视化结果。本次研究结果得到了实证分析的验证，具有很高的可操作性和结构简单，非专业人员易上手，检测效率高等特点。

关键词 机器视觉 stm32 图像检测

毕业设计说明书（论文）外文摘要

Title Machine Vision-based Meat Freshness Data Acquisition Device

Abstract

When using manual detection of meat quality, not only is the efficiency low, but also due to objective reasons, manual detection has great defects, and it is difficult to ensure accurate and effective identification results. As we all know, machine learning has the characteristics of fast, lossless, convenient and highly intelligent, in the preparation of the early stage of testing, the use of single-chip microcomputer control image acquisition device to obtain a large number of test sample images, using modular code implementation, simple operation, easy to use, high industry fit, can be widely used.

In this article, it is explained how machine vision can be applied to practical applications based on image inspection. A new meat freshness evaluation system using machine vision was designed. The main design idea is to first use a single-chip microcomputer to control the image acquisition device, collect the picture with the camera, send it to the computer, and then preprocess the image, use the neural network framework to recognize, and output the visualization results. The results of this study have been verified by empirical analysis, which has the characteristics of high operability, simple structure, easy to use by non-professionals, and high detection efficiency.

Keywords Machine vision Image detection

目 录

1 绪论	5
1.1 课题国内外研究现状	6
2 总体方案论证	7
2.1 系统方案	7
2.2 设计方案论证	7
3 硬件电路设计	9
3.1 装置结构	9
3.2 单片机系统设计	9
4 检测装置软件设计	13
4.1 图形可视化界面	14
5 基于机器视觉的肉类品质检测模型	17
5.1 数据处理	17
5.2 卷积神经网络框架	20
6 实物识别	27
6.1 整体工作流程	27
结束语	31
致 谢	32
参 考 文 献	33
附录 A 代码文件.....	35
附录 B 单片机系统原理图.....	37

1 绪论

随着人们生活水平的提升，猪肉在国内的肉类制品消费结构中占据很高的比例，近几年中在牛，羊，猪三大肉类的消耗量尤以猪肉为最甚（图 1.1），可以看出在餐桌上猪肉对我国人民的重要程度。

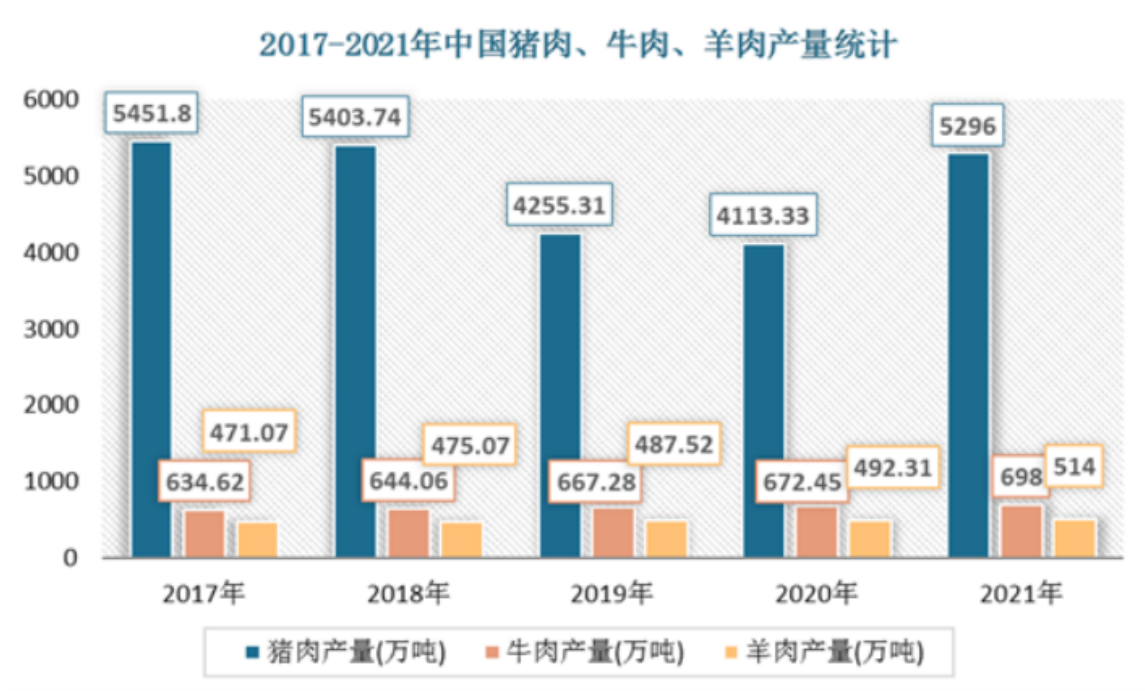


图 1.1 2017-2021 我国主要肉类和猪肉产量

随着猪肉的产量升高的同时，肉的品质越来越受到人们的广泛重视，因为猪经过高度选择性培育后却造成其适应外界能力较差，容易导致猪肉的品质下降，从而影响营养价值和实用价值，甚至出现一些劣质肉，使用后对人体健康产生危害。在购买猪肉时，通常人们通过判断肉的颜色来决定购买，当肉的颜色呈现暗紫色、灰白色时就会引起人们感官上的不适，从而拒绝购买。然而，猪肉在屠宰后变色是不可避免的，随着时间的推移，肉的颜色会暗红→鲜红→紫褐变化，这是由于猪肉的肌红蛋白被氧化成氧化肌红蛋白，同时当氧化时间过久时，颜色呈现紫褐色。产量过高就很有可能带来产量过剩的问题，一些不良商家就会以次充好，将变质的肉出售给人们，这就很大程度上导致了各类健康问题的出现，然而通过感官来判断猪肉品质的高低，显然无法达到人们的要求，以致于猪肉品质确定性研究越来越重要^[17]。通过查阅相关资料，了解到，当下的猪肉品质检测方法，个别虽然能准确识别猪肉品质却操作过于复杂，不能即时使用，另外的就是检测的速度满足要求，但是准确率差强人意。因而，研究出一种兼具稳定性，快速性

及高准确率的检测方法，是一个非常贴切实际生活的实用型的研究课题。

1.1 课题国内外研究现状

1.1.1 猪肉新鲜度无损检测研究现状

考虑到无损检测^[18]方法的特点是检测过程中不必使用的化学试剂，故不会对样品的完整性造成损坏，且没有繁杂的操作步骤，还具备检测客观高效，准确率较高。在无损检测中，采用图像处理，分析图像特征评估猪肉新鲜度，通常把样本图像划分区块，针对不同区块分别进行评估，从而使评估的准确度更高^[10]。赵红霞等以通过颜色理论构建了一个机器视觉系统用于猪肉颜色量化，保证了对肉的颜色评定时的客观化和定量化，具有快速无损、客观精准、经济成本低等特点^[11]。王爱国等猪肉的颜色进行测量，同时对像中 R 分量进行了具体说明，将肉的颜色五分类，然后五个类别的猪肉颜色对应的 RGB 数值范围，然后通过查表的方法对其进行评定^[12]。Gerrar 等人用计算机视觉从肉样品原始图像获取颜色特征参数值，定量的测定肉样本肌红蛋白的含量，同时对数据集做了线性回归分析^[13]。2003 年 D.V Byrne^[15]等人用图像处理和色度计的方式证明了，图像处理猪肉等级分类的优越性要优于色度计法，且检测的准确率也要高于色度计法。

2008 年 Patrick 等人从原始数据集中肌肉与脂肪肌肉中提取了区域 R、G、B、H、S、I 的标准差和平均值，得到两个数学模型的相关系数^[14]。2017 年 Sutskever^[16]等人用机器视觉对牛肉大理石纹进行分级研究，用 C 设计出牛肉品质评估检测系统，系统操作快速性较高，准确率达到了 87.5%。可以看出，国内外对猪肉品质研究领域依旧存在不少空白区域未涉及到，例如在制作数据集时，图像特征数量少，预测方法多为传统的图像处理方法或者浅层的人工神经网络模型，且一次只能预测一个表征猪肉品质的组成指标^[4]。深度学习的发展历程可以追溯到 1940 年代，当时人工神经网络的概念已经提出，然而由于计算技术的限制，深度学习技术并没有得到广泛应用。该技术无需损坏被测样品，并且无需使用任何化学试剂，就可以对被测样品做出快速且可靠的检测。同时图像处理技术、人工智能、传感器等科学技术对被测活体猪身体内部的物理性质进行深度分析。本文将机器视觉技术应用到猪肉图像采集之中，其明显的优势有：1)在采集图像时，可以避免对样品造成破坏，确保样品信息的完整性。2)可以同时快速的提取猪肉的多种特征。3)非专业人员使用易上手操作，工作效率高。

2 总体方案论证

设计一个黑色的检测盒，主要系统由 Stm32 单片机提供控制，然后在电脑上进行数据，其余装置包括：Stm32 单片机，LED 补光灯，摄像头，传感器。

2.1 系统方案

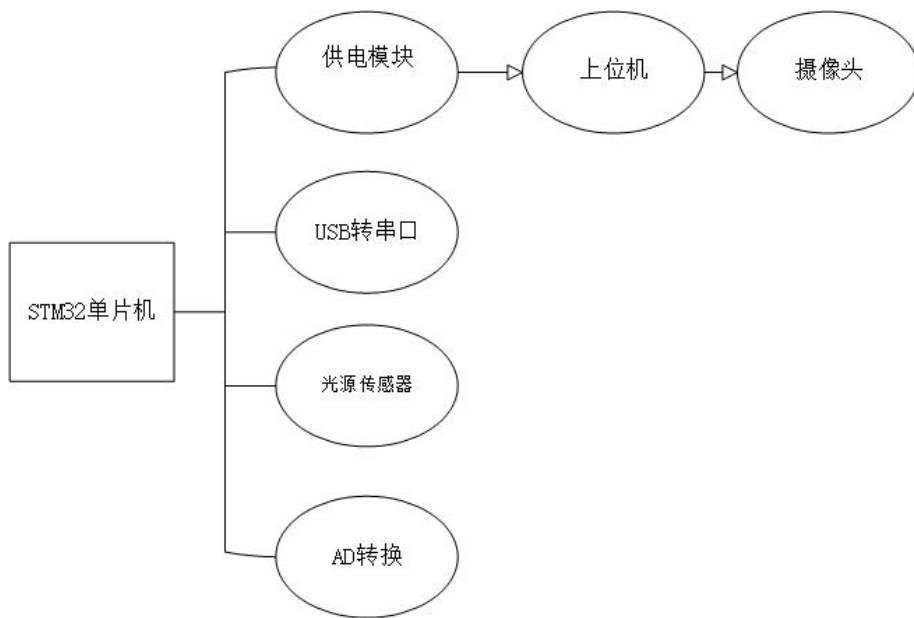


图 2.1 工作流程

检测流程:肉样本放在检测箱底上,摄像机对样本进行图像采集,获取图片;然后将图像传送给 PC 机,PC 机对图像进行分析处理,运用神经网络框架获取需要的参数^[3]。作流程如图 2.1 所示

2.2 设计方案论证

检测系统由两部分构成,一个就是根据食物气味,通过检测酸盐基氧化氮。选取电极传感器装置 MQ137（检测氨气），MQ136（检测硫化氢）。二是根据采集到的图像，通过搭建深度学习框架来进行检测。在猪肉样品放到箱体里后，用摄像图拍摄图像，保存在指定路径下，然后用图像采集卡经过数字化后传输到神经网络中进行检测。考虑到肉品腐败的过程是其内部成分发生生物化学变化产生新物质的过程，这直接会使其吸收系数和散射系数都发生较为明显的变化^[1]。为了更好的采集图像，增设一个补光灯,可以有效率地减小肉体漫反射,这样可以避免猪肉在表面大量水分流失的情况下对于镜面的光反射,进而提高获得的图片的清晰度。猪肉腐败的实质是猪肉在微生物的作用下发生变质以及在自身酶的分解作用下引

起 酸臭性发酵的过程，这一过程会产生氨气，硫化氢等挥发性物质，这些物质的含量会随着肉品腐败的加剧而增加^[19]。

采集到的图像在数据库里，在处理装置将图像信息中的红绿蓝三色的灰度数值、色调，饱和度，亮度转化为数据库可识别的信息后，数据库会用神经网络进行处理，处理好的信息保存在指定路径，最后用机器学习进行检测。

3 硬件电路设计

3.1 装置结构

选取四块黑色板组成四面不透光的箱子，在箱子中间加入 LED 光源，将 STM32 单片机放在箱体上方的检测板上，使用 MQ137 和 MQ136 传感器测量空气中气体数据，在箱子旁边有双目摄像头用于采集肉类的图像数据。流程图如图 3.1 所示。

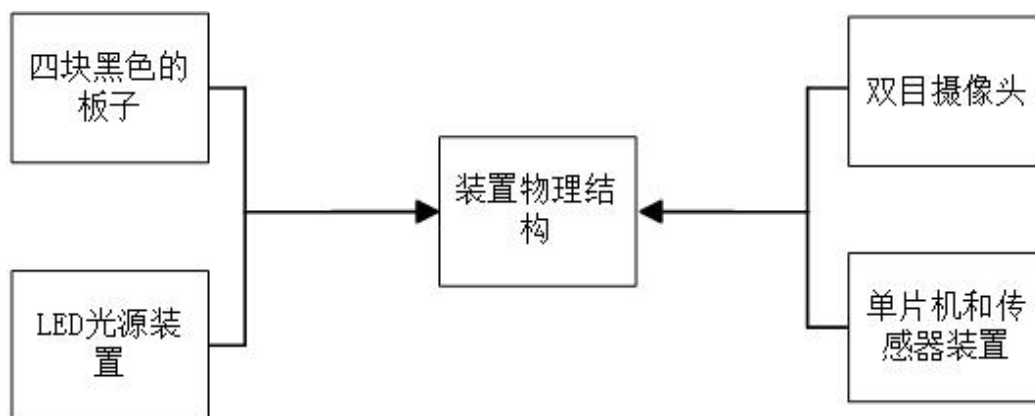


图 3.1 机械结构图

其具体步骤如下：

- (1)把处理过的猪肉放置于箱体中间。
- (2 把 LED 置于箱内适当高度，使得猪肉光照均匀，保证图像清晰完整。

3.2 单片机系统设计

单片机由电源模块，时钟，调试，复位，以及控制芯片组成。电路图放在附录 B。

3.2.1 单片机控制模块

如图 3.2，用 CH340 将 USB 连接笔记本的端口转换成 TTL 电平（如图 3.2）后，与电脑进行连接，VCC 接 USB 的 5V，V3 引脚需外接 $0.1\mu\text{F}$ 去耦电容，以使用 CH340G 自身的降压对芯片的工作供电。USB-5V 供电，需要外接 12MHz 晶振^[5]。另外加装双电源模式下防电流内灌电路。采用 NPN 串联一个二极管，防双向倒灌。

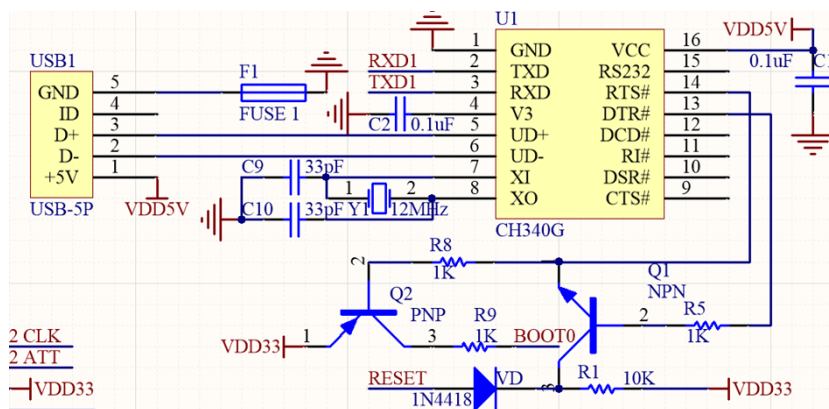
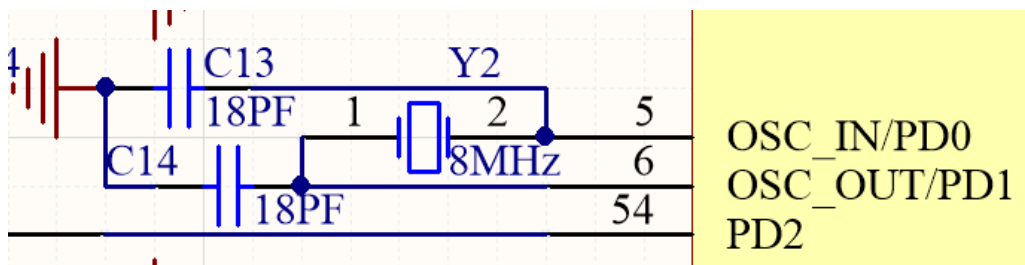


图 3.2 TTL 电平转 USB 电平电路原理图

3.2.2 时钟电路

如图 3.3，时钟电路提供一个系统内的时钟，电路板通电后开始工作。



3.3 时钟电路图

3.2.3 复位按钮

如图 3.4 所示，NEST 处低电平或计数终止复位。

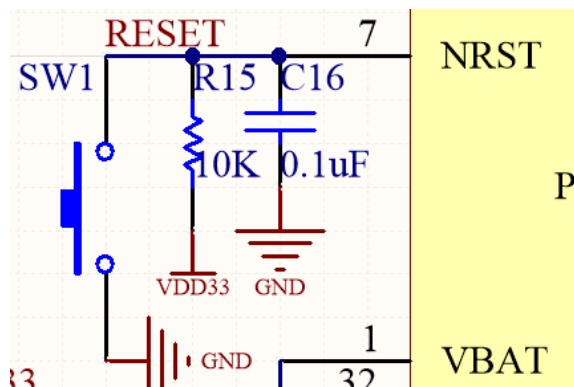


图 3.4 复位电路图

3.2.4 程序下载电路

如图 3.5 所示，考虑到 SWD 节省接口资源，下载调试速度也快，同时比 JTAG 在高速模式下面更加可靠，在数据量较大的情况下，SWD 下载成功率也较高。

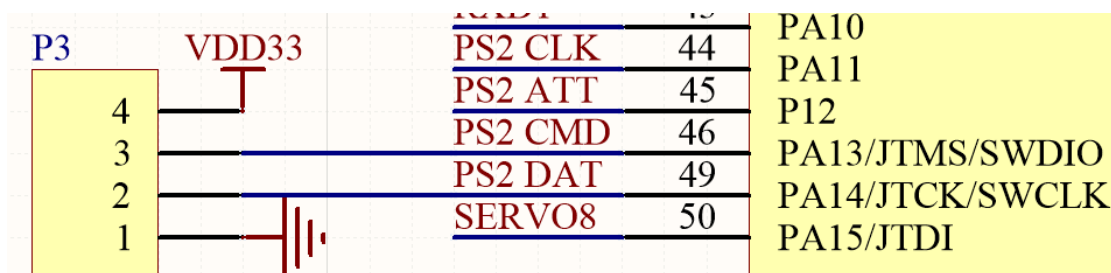


图 3.5 程序 SWD 下载电路

3.2.5 FLASH 存储器

选取 25Q64(64M-bit)。如图 3,6 所示，/CS 端为片选端，高电平时，不选择芯片，芯片处于待机状态，低电平时，芯片正常工作，CLK（空闲时为高电平）上升沿时 DI 写入命令，CLK 下降沿时 DO 输出命令，/WP 引脚在低电平时和状态寄存器对存储器进行保护，/HOLD 低电平时，芯片暂停工作。配置函数如下：

```
void SPI_W25Q_Init(void);
```

```
Void SPI_W25Q_Read(u8*pBuffer,u32 ReadAddr,u16 NumByteToRead);
```

```
void SPI_W25Q_Write(u8*pBuffer,u32 WriteAddr,u16 NumByteToWrite);
```

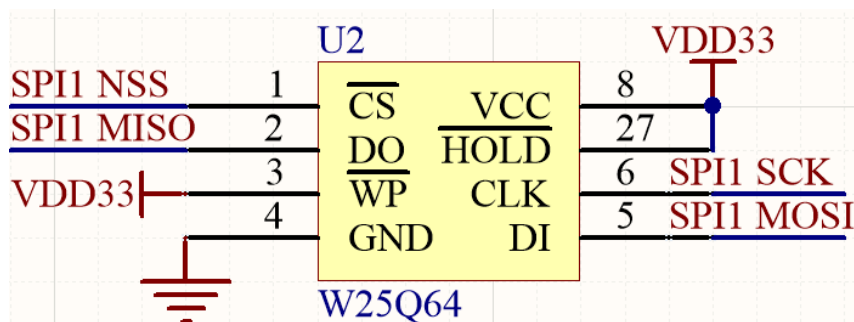


图 3.6 FLASH 芯片电路图

3.2.6 图像采集单元

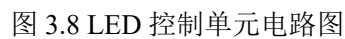
图像的采集选用双目 CCD 摄像头，如图 3.7。



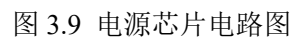
图 3.7 CCD 摄像头

3.2.7 光源单元

如图 3.8，PWM 占空比大小电机运行速度成正比，由于单片机的工作电流较小，达不到 LED 的工作电流，所以通过 TB6612FNG 驱动来实现。



如图 3.9，根据各个组成部分的功能，电源模块应提供电机驱动所需的 12V、STM32 主控核心所需的 3.3V、其他芯片工作所需的 5V 三幅值的电压。因此，采用 12V 的航模电池作为供电电源^[5]。



4 检测装置软件设计

用 keil5 编写好 Stm32 代码，通过气体传感器的数据动作程序，存入存储器当中，然后读写的数据。主程序流程图 4.1 所示，初始化运行后，定时器设定初值，打开中断允许，中断打开后，定时器开始工作，工作时，如有中断触发，主程序中断优先级优先于定时器中断优先级。

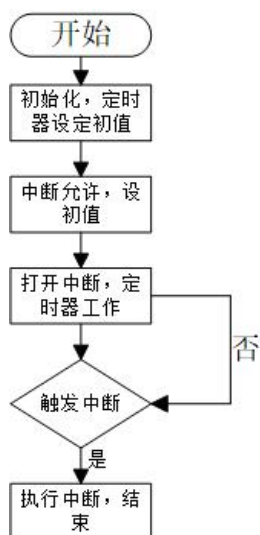


图 4.1 主程序流程图

单片机串口工作流程图如图 4.2。

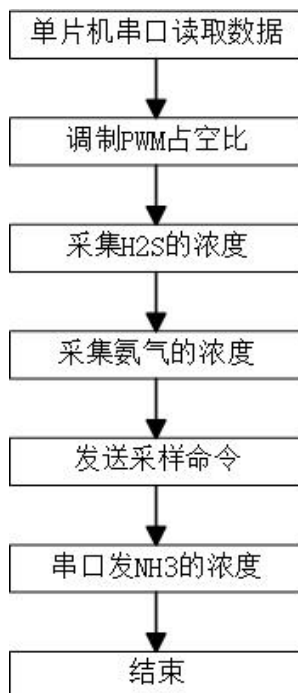


图 4.2 串口数据传输中断流程图

4.1 图形可视化界面

使用 tkinter 函数配，通过 tk.Button 函数来配置按钮，函数是来配置按钮,使用 command 实现要求实现的各种功能,如图 4.3 所示。

```
self.ab_Btn = tk.Button(self.win_Main, text='系统说明', font=('微软雅黑', 12), width=10, height=1,
                        command=self.ShowMe).pack(side='left', padx=20)
self.con_Btn= tk.Button(self.win_Main, text='连接系统', font=('微软雅黑', 12), width=10, height=1,
                        command=self.Connect_Box).pack(side='right', padx=20)
```

图 4.3 command 调用函数

使用 tkinter 函数搭建窗口的代码如图 4.4 所示。

```
Msg.showinfo('请检查摄像头是否正确连接!')
self.panel = tk.Tk()
self.panel.title('肉类品质检测控制板')
self.panel.geometry('400x800+900+600')
self.panel.focus_set()
self.io = Box_IO()
self.processor=Img_Processor()
thread.start_new_thread(self.Loop_Video, ())
self.show_BtnL = tk.Button(self.panel, text='显示原始图像', font=('微软雅黑', 12), width=15, height=1,
                        command=self.processor.Show_Prime).pack(side='top', padx=20, pady=20)
self.show_BtnL = tk.Button(self.panel, text='显示HSV图像', font=('微软雅黑', 12), width=15, height=1,
                        command=self.processor.Show_HSV).pack(side='top', padx=20, pady=20)
self.show_BtnL = tk.Button(self.panel, text='存储图像', font=('微软雅黑', 12), width=15, height=1,
                        command=self.processor.Flick).pack(side='top', padx=20, pady=20)
self.show_BtnL = tk.Button(self.panel, text='肉质识别', font=('微软雅黑', 12), width=15, height=1,
                        command=self.processor.Rec).pack(side='top', padx=20, pady=20)
c1 = tk.Checkbutton(self.panel, text='左摄像头', variable=self.processor.left_Open, onvalue=1, offvalue=0, command=self.processor.prime_Dat) #
c1.pack(side='left')
c2 = tk.Checkbutton(self.panel, text='右摄像头', variable=self.processor.right_Open, onvalue=1, offvalue=0, command=self.processor.prime_Dat)
c2.pack(side='right')
```

图 4.4 tkinter 配置操作界面

具体界面如图 4.5 所示。

选用 Opencv 处理图像，函数配置如下，

Cv2.imread('zhurou/1.jpg')

Cv2.imshow('img', img)

Cv2.imwrite(str2, self.processor.cam.frame_L)

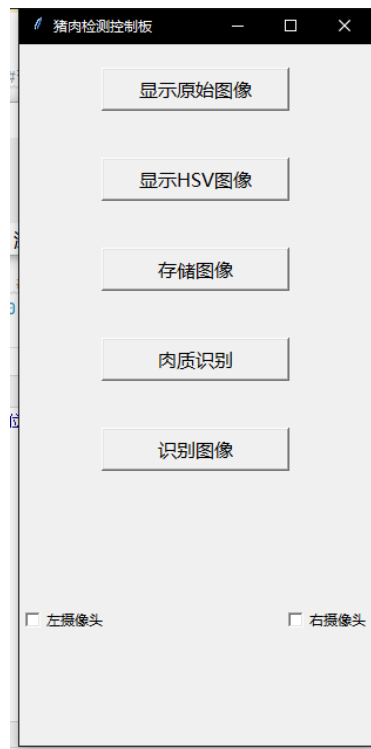


图 4.5 显示界面

初始图像窗口，如图 4.6 所示。

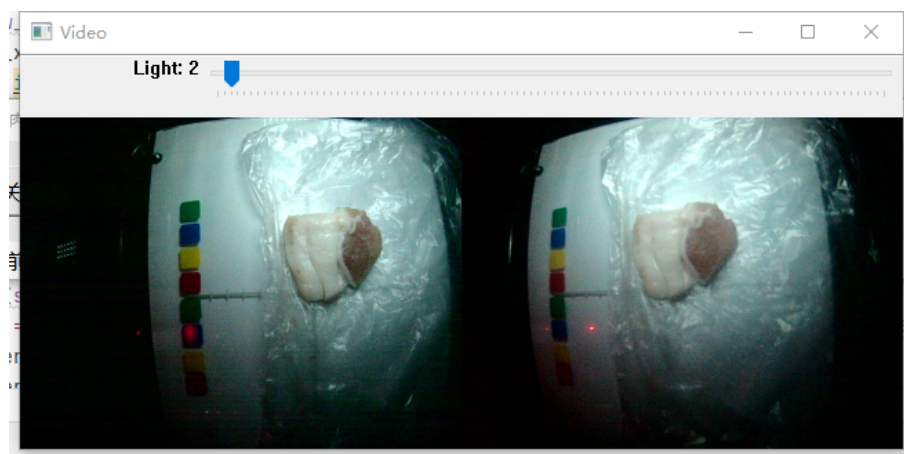


图 4.6 初始图像

显示 HSV 图像（由 RGB 颜色模型计算 H, S, V ^[18] 三个通道像素之和得到，如图 4.7 所示。

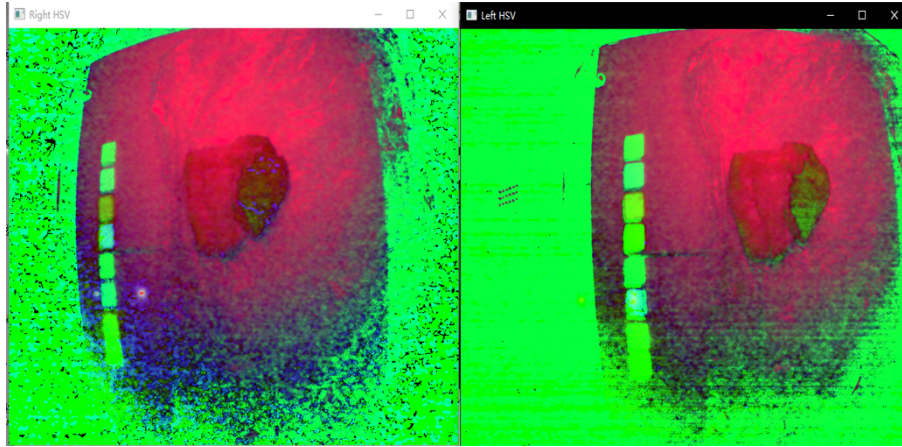


图 4.7 转化为 HSV 图像

将初始图像保存到本地磁盘 D，如 4.8 所示

```
str=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
cv2.imwrite("D:\%s.jpg"%str,self.cam.frame_L)
Msg.showinfo('Done', '图像已写入D:\%s.jpg'%str)
```

图 4.8 将图像保存到 D 盘

检测结如图 4.9 所示。

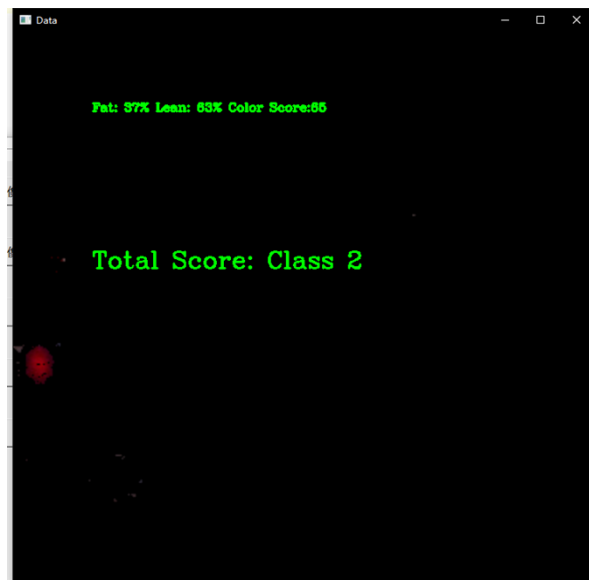


图 4.9 二值化简单识别猪肉

5 基于机器视觉的肉类品质检测模型

5.1 数据处理

采用图像特征的图像处理分析技术能够迅速获取很大的信息量，并且可以自动处理，同时容易与设计方面的信息以及加工控制信息集成数据处理^[8]。通过大量的关于目标的输入特征和对应标签的建立出数据集，然后放入框架进行神经网络训练，新数据通过对比网络就可以得出结果。

5.1.1 数据集的制作

在制作数据集时，为了获得清晰度高，对比度好的原始数字图像，采用 Nikon D7000 数码相机、光照箱、计算机制作。初步分割好的猪肉样品放于光照箱^[18]内进行图像信息采集，并将拍摄好的原始图像通过数码相机接口传输到计算机中。在拍摄完猪肉的图片后我们需要对这张图像数据进行处理,之后成为应用程序所需要的数据，，拍摄的图片以 JPG 格式保存^[7]，然后将其分为训练集和测试集保存在指定文件夹中。然后给所有图片打上标签，将图片名和对应的标签一一对应保存。之后规定肉的新鲜度分类等级，新鲜度是评价生鲜猪肉质量的重要指标之一^[6]，直接影响着消费者的购买意愿，在本试验中将所有猪肉的新鲜率分别确定为新鲜与次级肉一级、次级肉二级、腐肉一级、腐肉二级 5 个品种。标签设置 5 级对应。

使用 `generateds` 函数，将图片所在位置和标签好文件输入所写的函数中，同时使用 `generateds` 函数返回输入特征和标签，因此图片的路径作为第一参数，标签的文件作为第二参数，将图片路径与拍摄图路径重合同时可实现拍摄图片与检测。

5.1.2 数据预处理

把图像分成独特性质的、若干特定的区域，并把感兴趣目标提取出来的过程和技术，称为图像分割利用边缘的最小外接矩^{Error! Reference source not found.}，如图 5.1。裁剪猪肉样本的分辨率为 28*28，调整大小后约 10KB^[7]。



图 5.1 样本裁剪示意

噪声是造成图像干扰的重要原因，减少数字图像中所含有的噪声，使图像质量达到预期效果的过程称为图像去噪。去噪是图像处理的基础，往往在更高级的图像处理之前进行此步骤。经过去噪处理后的图像质量可以得到一定程度的改善，信息量的可靠性变强，提高图像分析和判读效果^[20]。

在去除噪点后，将高斯函数应用于图像的每个像素，将像素点与周围的像素按照高斯函数的权重进行加权平均，以消除图像中的噪点，用高斯模糊的方式进行图片降噪，函数如图 5.2 所示。

```
import cv2
img = cv2.imread("image.jpg") # 加载图像
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # 将图像转换为灰度图像

blur = cv2.GaussianBlur(gray, (5,5), 0) # 应用高斯模糊
thresh = cv2.adaptiveThreshold(blur, 255,
                               cv2.ADAPTIVE_THRESH_MEAN_C,
                               cv2.THRESH_BINARY_INV, 11, 4) # 应用自适应阈值
cv2.imshow("Processed Image", thresh) # 显示结果
cv2.waitKey(0) # 等待键盘输入
```

图 5.2 图片降噪函数

在进行噪点消除后，进行数据集增强操作如图 5.3。

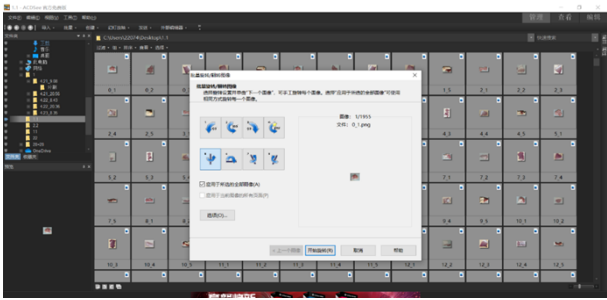


图 5.3 数据增强操作

5 个等级如上图 5.4~5.8 所示。



图 5.4 新鲜

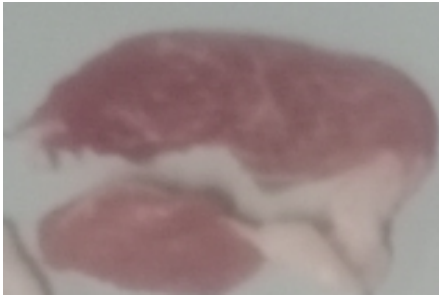


图 5.5 次级肉一级



图 5.6 次级肉二级



图 5.7 腐败肉一级

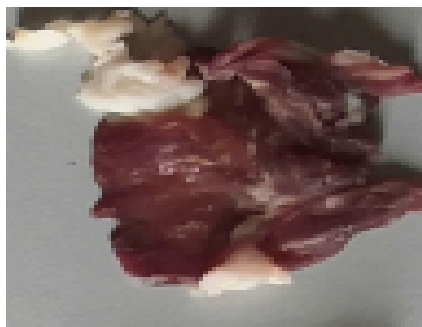


图 5.8 腐败肉二级

5.2 卷积神经网络框架

卷积：指输入数据与某一组卷积核进行卷积运算后，从而提取出数据中的特征，用于后续对数据的提取和分类操作。对于提取数字特征的运算过程如图 5.9 所示。

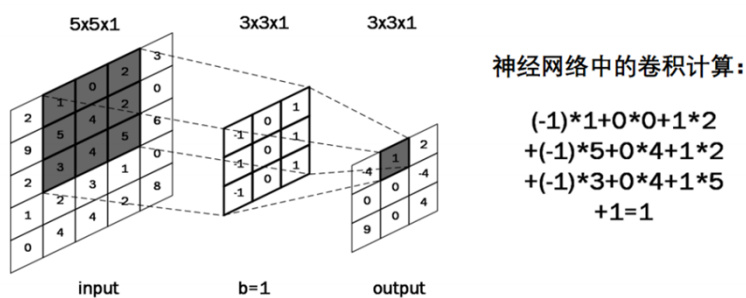


图 5.9 卷积计算

感受野就是输入的数据对某一层的输出神经元所造成的影响程度。影响大，但是不一定好。其原理如图 5.10 所示。当有 3 个特征图输出时，用 2 个 3*3 的卷积层可以代替 1 个 5*5 的卷积层。第一层中所有部分视作一个元素，3*3 的绿色方格为 3*3 的卷积核。第二层由第一层中 3*3 的卷积核由卷积运算输出，输出的尺寸是 3*3。明显可以看出，层次 2 中的绿色方格是通过第一层中 3*3 的卷积核所得出来的，此时，第二层的感受野就是 layer1 中的绿色方格区域。同样的，第三层中的黄色方块由第二层经过 3*3 的卷积层输出。

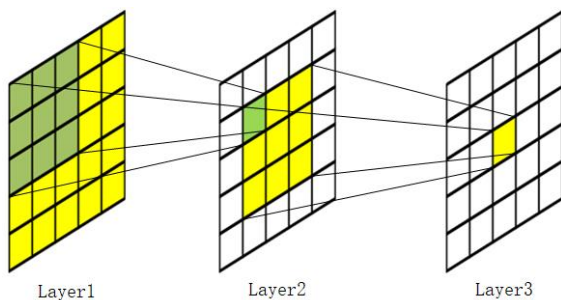


图 5.10 感受野原理

全零填充：如图 5.11 所示，在输入的数据周围填充一层或者 2 层 0，则输出数据尺寸不发生变，这里选择 5*5 举例说明。

计算方式为： $(-1) * 0 + 0 * 2 + 1 * 1 + (-1) * 0 + 0 * 9 + 1 * 5 + (-1) * 0 + 0 * 2 + 1 * 3 + 1 = 10$ 。

输出边长=输入边长/步长,此图：5/1=5。

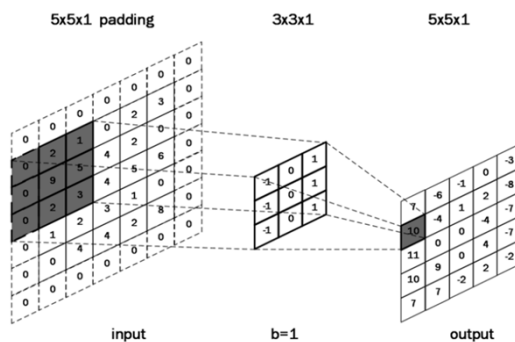
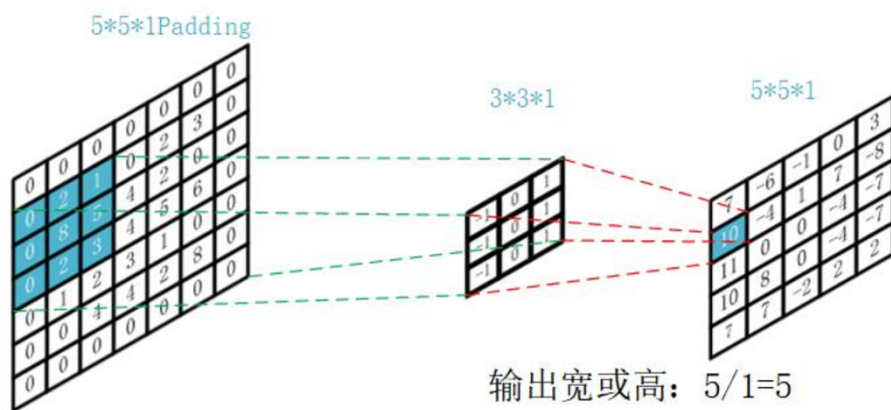


图 5.11 全零填充

在 TF 框架中，用参数 `Padding="SAME"` 或 `Padding='VALID'` 表示是否进行全零填充，如图 5.12 所示。



Padding="SAVE" 时，输入图像5*5*1，输出图像5*5*1

图 5.12 全零填充示意图

计算方法为： $(-1) * 0 + 0 * 2 + 1 * 1 + (-1) * 0 + 0 * 9 + 1 * 5 + (-1 * 0 + 0 * 2 + 1 * 3 + 1 = 10$;

当没有全零填充时，输出边长=（输入边长-卷积核边长+1）/步长，可能产生信息丢失，为避免信息丢失，本次试验采用全零填充，因为卷积的时候，希望图像大小不发生变化。

批标准化：深度神经网络训练的一种方法，其可以在训练层数加深时，根据需求定义将数据约束到固定的一定范围,使得转化后的数值满足一定的特性分布，

使得训练深层网络模型更加简易和稳定,从而避免因训练层数加深导致训练失败。在训练时,当数值较大时,其参数会对训练结果的影响较大,那么解决这个问题,就要设置多种学习率,不够便捷,那么使用批量标准化提前对数据进行预处理。使数据具有统一格式,这样可以实现每次训练时仅设置一个学习率就行。

池化:通过对像素值的最大值或平均选取,可以减小特征图的大小,降低计算量。如图 5.13 所示。

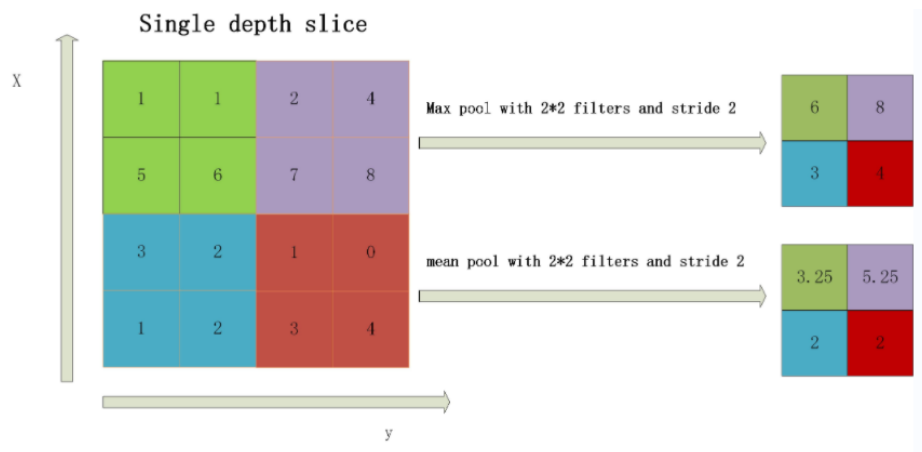


图 5.13 池化流程

舍弃: 在训练过程中,把一部分神经元根据一定概率从神经网络中暂时舍弃,要使用时其又会恢复链接。

全连接: 神经元互相之间有联系,在针对一张分辨率仅为 $28*28$ 的黑白图像全连接时的网络的参数个数为: $28*28*1=784$ 。

丢弃: 当训练集数据量不足、模型复杂度过高、训练过程中的噪声干扰产生过拟合时,会导致模型训练效果达不到预期。采用 dropout 可以解决这个问题。

主要模块如图 5.14 所示。

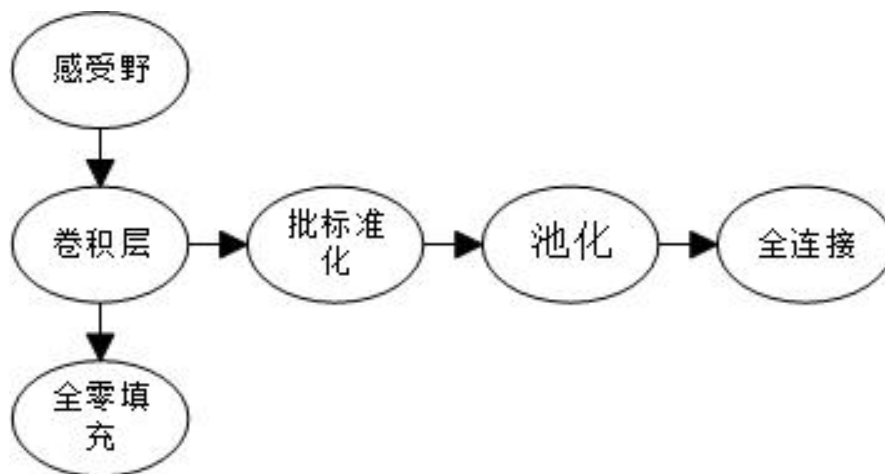


图 5.14 主要模块图

5.2.1 卷积网络框架

使用此网络框架（如图 5.15）运行我们自制的数据集，进行多次训练。



图 5.15 EfficientNet 框架示意图

选择训练次数 20 次时的准确率和损失率的曲线图如图 5.15 所示，随着训练次数的增加，准确率不断的提高，损失率不断下降，达到理想效果。

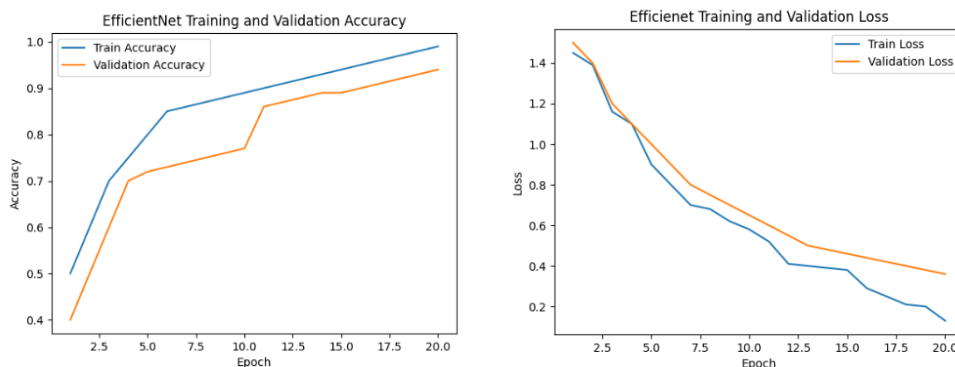


图 5.15 EfficientNet 模型准确率和损失率

EfficientNet—B0 的基线网络（表 5-16）中，每行用 L_i 层划分阶段为 Stage i ，其输出分辨率为 $(H_i$ 和 $W_i)$ ，输出通道为 C_i 。

表 5-16 EfficientNet—B0 模型的网络结构

Stage i	Operator	Resolution $H_i \times W_i$	Channels	Layers
1	Conv3*3	224*224	32	1
2	MBConv1,K3*3	112*112	16	1
3	MBConv6,K3*5	112*112	24	2
4	MBConv6,K5*5	56*56	40	2
5	MBConv6,K3*3	28*28	80	3
6	MBConv6,K5*5	14*14	112	3
7	MBConv6,K5*5	14*14	192	4
8	MBConv6,K3*3	7*7	320	1
9	Conv1*1&Pooling&FC	7*7	1280	1

Channel 的维度为 Width_coefficient，depth 的维度为 depth_coefficient。

表 5-17 中，在 B0 中的 Stage3 的 3*3 Channel 用到的卷积核个数为 24，对应的在 B7 中为 $24*2=48$ 。取整到最近的 8 的整数倍即 48，其他 Stage 同理。

Depth 的倍率因子仅 Stage2 到 Stage8 有效，比如在 B0 中的 Stage7 的 $L=4$ ，对应的在 B7 中为 $4*3.1=12.4$ ，向上取整 13。

表 5-17 channel 维度和 depth 倍率因子选取

Model	Input_size	Width_coef ficient	Deep_coef ficient	Drop_conne ct_rate	Dropout_ rate
EfficientNetB0	224*224	1.0	1.0	0.2	0.2
EfficientNetB1	240*240	1.0	1.1	0.2	0.2
EfficientNetB2	260*260	1.1	1.2	0.2	0.3
EfficientNetB3	300*300	1.2	1.4	0.2	0.3
EfficientNetB4	380*380	1.4	1.8	0.2	0.4
EfficientNetB5	456*456	1.6	2.2	0.2	0.4
EfficientNetB6	528*528	1.8	2.6	0.2	0.5
EfficientNetB7	600*600	2.0	3.1	0.2	0.5

构建的最终框架如图 5.18 所示。

```
class EfficientNet(nn.Module):
    def __init__(self,
                 width_coefficient: float,
                 depth_coefficient: float,
                 num_classes: int = 1000,
                 dropout_rate: float = 0.2,
                 drop_connect_rate: float = 0.2,
                 block: Optional[Callable[..., nn.Module]] = None,
                 norm_layer: Optional[Callable[..., nn.Module]] = None
                 ):
        super(EfficientNet, self).__init__()
        #
        default_cfg = [[3, 32, 16, 1, 1, True, drop_connect_rate, 1],
                       [3, 16, 24, 6, 2, True, drop_connect_rate, 2],
                       [5, 24, 40, 6, 2, True, drop_connect_rate, 2],
                       [3, 40, 80, 6, 2, True, drop_connect_rate, 3],
                       [5, 80, 112, 6, 1, True, drop_connect_rate, 3],
                       [5, 112, 192, 6, 2, True, drop_connect_rate, 4],
                       [3, 192, 320, 6, 1, True, drop_connect_rate, 1]]
        def round_repeats(repeats):
            return int(math.ceil(depth_coefficient * repeats))
        if block is None:
            block = InvertedResidual
        if norm_layer is None:
            norm_layer = partial(nn.BatchNorm2d, eps=1e-3, momentum=0.1)
        adjust_channels = partial(InvertedResidualConfig.adjust_channels,
                                width_coefficient=width_coefficient)
        bnneck_conf = partial(InvertedResidualConfig,
                              width_coefficient=width_coefficient)
```

图 5.18 主要函数组成

6 实物识别

6.1 整体工作流程

工作流程图如图 6.1 所示。

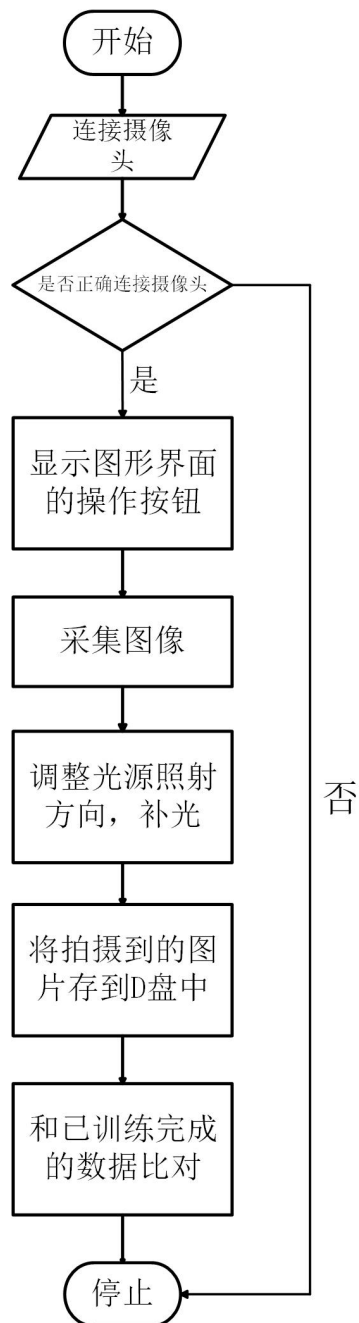


图 6.1 识别流程图

第一步，将猪肉放入检测盒当中,如图 6.2 与图 6.3 所示。



图 6.2 硬件外形



图 6.3 检测盒内部外形

检测控制界面如图 6.4 所示。



图 6.4 控制界面

将肉放入检测盒当中，此时显示图像比较暗按下显示图像如图 6.5 所示。

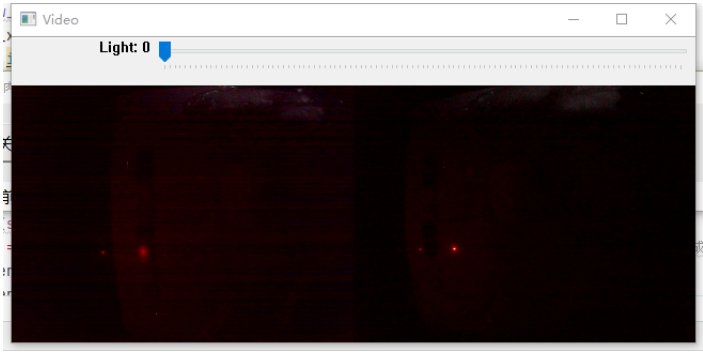


图 6.5 放入猪肉检测图像

调整盒内光的亮度，此时按下显示图像如图 6.6 所示。

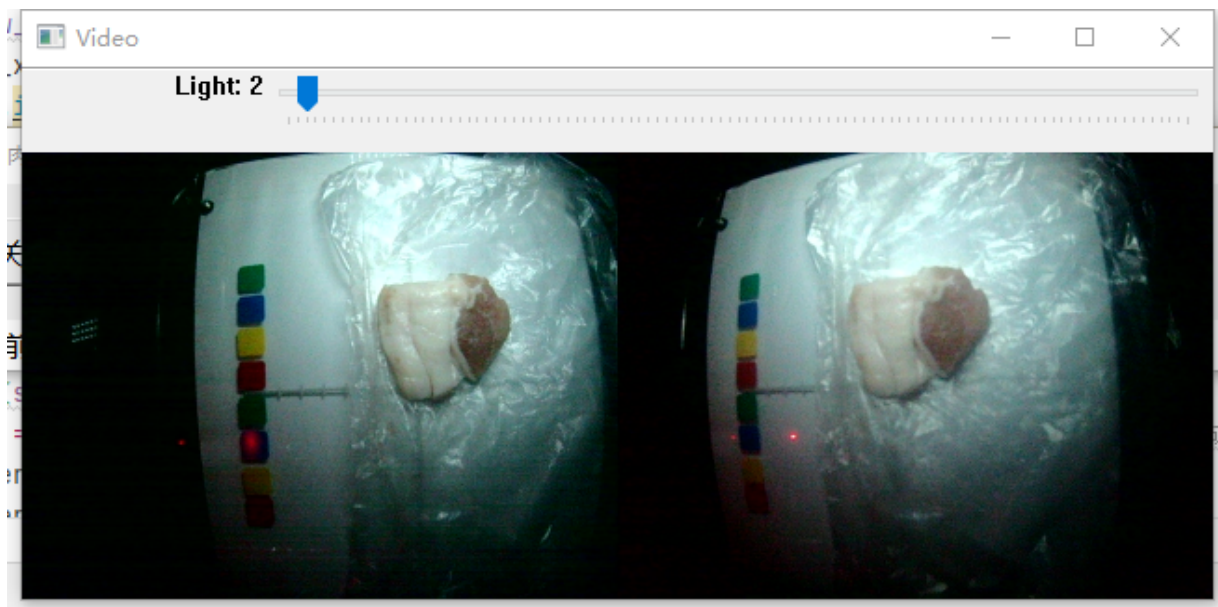


图 6.6 调整亮度

识别结果如图 6.7 所示。



图 6.7 识别结果

结束语

随着当下社会节奏加快,同时老百姓对是食品安全要求的进一步提高,我们生活的方式和环境有着巨大的变化,各方面的要求都有了进一步的提高,又因为猪肉等肉类作为日常的大量消耗品,安全问题近年来尤为突出,传统的人工检测显然已经不能满足人民所需,目前各类检测领域不断推出各种肉类品质检测工具,致力于提升肉类检测的速度,同时考虑到实际要求的稳定性和便捷性,经探索后,了解到机器视觉系统由于其种类多样分类功能体系、便民的成本和利民的控制,其猪肉检测系统在社会上得到了高度的关注及业内专业人士深入研发和投入使用,纵观当下,选取机器学习来检测肉类已经在所难免。

本试验所设计的系统在研究上外观结构可塑性强、快速性投入使用效率高、稳定性较高等特点,是一种非常贴切生活实际的应用型的设计。

致 谢

本次论文的完成，首先我要感谢我的指导老师刘超对我在完成论文设计中提供的大力支持，在撰写论文期间，刘超老师在本文的课题的选题，内容排版，文献框架，行文布局等方面给我提出了建设性的帮助。刘超老师尽管还要给班级上课，但是在我遇到问题的时候总能及时的得到刘超老师的及时帮助，在刘超老师的悉心指导下，我完成论文的进度有了很大的提高，在这期间，我学会了绘图软件 AD，和学会了编写 Python，以及多所学的专业课知识有了更进一步的认识。最终在刘超老师抽出大量时间帮我多次批阅论文后，不断完善，最终完成定稿。然后还要感谢我的舍友，在本次设计过程中，给我提供了一些论文文献参考和设计思路，和督促我去及时的完成任务，给本论文的顺利完成带来了极大的帮助。最后感谢我的家人，是他们培养了我，感谢长久以来他们对我默默的支持，他们的支持给了我坚强的后盾，使我能够全力以赴的去提升自我，我铭记在心。再次对本次论文的的指导老师刘超老师致以崇高的谢意以及对所有帮助过我的老师致以崇高的谢意。

参 考 文 献

- [1] Marta Castro-Giráldez, Patricia Botella, Fidel Toldrá, Pedro Fito. Low-frequency dielectric spectrum to determine pork meat quality[J]. Innovative Food Science and Emerging Technologies, 2010, 11(2).
- [2] Antonio Girolami, Fabio Napolitano, Daniela Faraone, Ada Braghieri. Measurement of meat color using a computer vision system[J]. Meat Science, 2013, 93(1).
- [3] 陈洵凇, 黎小巨, 许梓烁, 谢小鹏. 基于 OpenCV 的工件尺寸与划痕检测算法研究[J]. 机床与液压, 2020, 48(08):109-111+131.
- [4] 石磊. 基于 CNN 的猪胴体组成测定方法研究[D]. 西北农林科技大学, 2019. .
- [5] 王嘉俊. 基于 STM32 的智能小车控制系统设计[J]. 电子制作, 2018(17):21-22+25. DOI:10.16589/j.cnki.cn11-3571/tn.2018.17.008.7.
- [6] 李翠玲. 基于多光谱成像的猪肉新鲜度便携式快速无损检测技术及装置开发[D]. 中国农业大学, 2015.
- [7] 焦俊, 王文周, 侯金波, 孙裴, 何屿彤, 辜丽川. 基于改进残差网络的黑毛猪肉新鲜度识别方法[J]. 农业机械学报, 2019, 50(08):364-371.
- [8] 孙瑾, 顾宏斌. 计算机视觉系统框架结构研究[J]. 计算机工程与应用, 2004(12):44-47.
- [9] Feifei Tao, Yankun Peng. A method for nondestructive prediction of pork meat quality and safety attributes by hyperspectral imaging technique[J]. Journal of Food Engineering, 2014, 126.
- [10] J. -H. Liu, X. Sun, J. M. Young, L. A. Bachmeier, D. J. Newman. Predicting pork loin intramuscular fat using computer vision system[J]. Meat Science, 2018, 143.
- [11] 赵红霞, 周丰昆, 刘静波, 孙佳石. 基于计算机视觉的肉色量化研究(英文)[J]. 食品科学, 2004(10):95-100.
- [12] 洪嵘, 吴常信, 张勤, 肖宁, 王爱国. 动物胴体组成图像分析系统的研制[J]. 中国畜牧杂志, 1996(06):5-7.

-
- [13] B. P. Demos, D. E. Gerrard, X. Gao, J. Tan, R. W. Mandigo. Utilization of image processing to quantitate surface metmyoglobin on fresh beef[J]. Meat Science, 1996, 43(3).
- [14] Cheng-Jin Du, Da-Wen Sun, Patrick Jackman, Paul Allen. Development of a hybrid image processing algorithm for automatic evaluation of intramuscular fat content in beef M. longissimus dorsi[J]. Meat Science, 2008, 80(4).
- [15] M. G. O'Sullivan, D. V. Byrne, J. H. Nielsen, H. J. Andersen, M. Martens. Sensory and chemical assessment of pork supplemented with iron and vitamin E[J]. Meat Science, 2003, 64(2).
- [16] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6).
- [17] 孙宝. 基于机器视觉的猪肉品质检测技术研究[D]. 西南科技大学, 2019.
- [18] 张永伟. 融合图像和嗅觉特征的猪肉新鲜度分级[D]. 东北电力大学, 2018.
- [19] 王丹凤, 王锡昌, 刘源, 赵勇, 顾赛麒, 谢晶, 潘迎捷. 电子鼻分析猪肉中负载的微生物数量研究[J]. 食品科学, 2010, 31(06):148-150.
- [20] 韩旭. 基于运动趋势估计的人脸跟踪技术研究[D]. 东北大学, 2009.

附录 A 代码文件

```
#coding=utf8
import cv2
from cam_reader import Cam_Read
import tkinter as tk
import numpy as np
import time
import tkinter.messagebox as Msg
class Img_Processor:
    def __init__(self):
        self.cam=Cam_Read()
        self.prime_Dat=None
        self.hsv_Dat=None
        self.Extracted_Box=None
        self.Eroded=None
        self.left_Open=tk.IntVar()
        self.right_Open = tk.IntVar()
        self.alive=True
    def Refresh_Dat(self): pass
```

```
def Show_Prime(self):  
    #if self.left_Open.get()==0:  
        cv2.imshow('Left Camera',self.cam.frame_L)  
    #if self.right_Open.get() == 1:  
        cv2.imshow('Right Camera', self.cam.frame_R)  
  
def Show_HSV(self):  
    #if self.left_Open.get() == 1:  
        hsv_l = cv2.cvtColor(self.cam.frame_L, cv2.COLOR_BGR2HSV)  
        cv2.imshow('Left HSV', hsv_l)  
    #if self.right_Open.get() == 1:  
        hsv_r = cv2.cvtColor(self.cam.frame_R, cv2.COLOR_BGR2HSV)  
        cv2.imshow('Right HSV', hsv_r)  
  
def Show_tx(self):  
    #读取图片  
    hsv_x = cv2.imread('D:/1253.png')  
    cv2.imshow('results', hsv_x)  
    Msg.showinfo('猪肉测试', '猪肉等级为\n\n 次鲜肉二级')  
    Msg.showinfo('氨气测试', '检测到存在氨气')  
  
def Flick(self):  
    #存入图像  
    str=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())  
    cv2.imwrite("D:\%s.jpg"%str,self.cam.frame_L)  
    Msg.showinfo('OK','图像已经写入 D:\%s.jpg"%str)  
  
def Rec(self):  
    #肉质识别  
    hsv = cv2.cvtColor(self.cam.frame_L[100:450,230:600],  
cv2.COLOR_BGR2HSV)#将 BGR 格式转换成 hsv 图片  
    lower_blue = np.array([120, 50, 50])  
    upper_blue = np.array([180, 255, 255])  
    mask = cv2.inRange(hsv, lower_blue, upper_blue)  
    # Bitwise-AND mask and original image  
    res = cv2.bitwise_and(self.cam.frame_L[100:450,230:600],  
self.cam.frame_L[100:450,230:600], mask=mask)
```

```
img=cv2.resize(res, None, fx=2, fy=2)

cv2.putText(img,'Fat: 37% Lean: 63% Color Score:65',(100,100),cv2., 0.5,
(0, 255, 0), 2)

cv2.putText(img, 'Total Score: Class 2', (100, 300),
cv2.FONT_HERSHEY_COMPLEX,
1.0, (0, 255, 0), 2FONT_HERSHEY_COMPLEX)

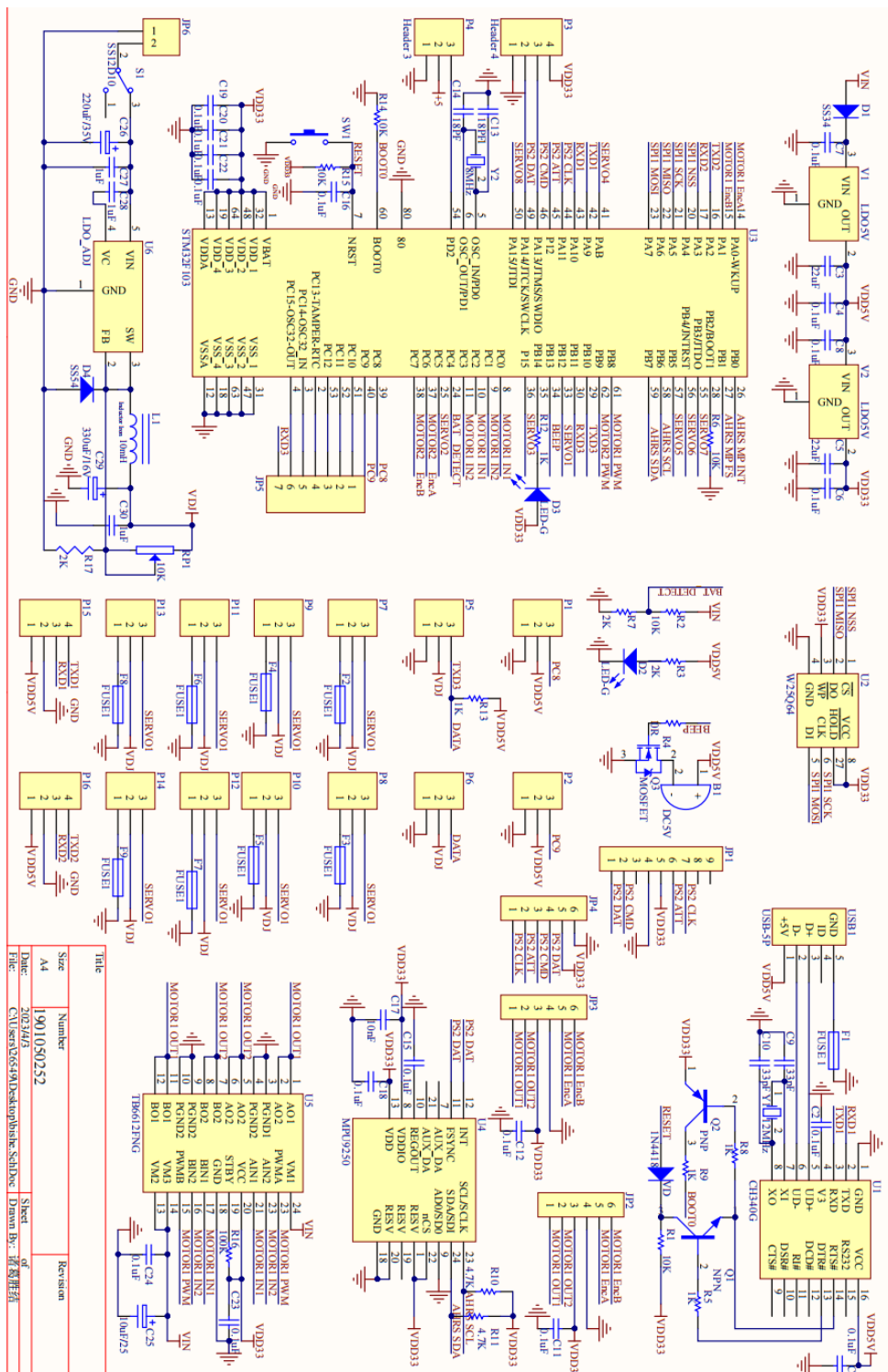
cv2.imshow("Data",img )

kernel = np.ones((5, 5), np.uint8)

closing = cv2.morphologyEx(cv2.resize(res,None,fx=2,fy=2),
cv2.MORPH_CLOSE, kernel)

#cv2.imshow("肉质提取", closing)
```

附录 B 单片机系统原理图



Title			
Size	Number	Revision	
A4	1901050252		
Date:	2022/4/3	Sheet	of
File:	C:\Users\6649\Desktop\shishu.schDoc	Drawn by:	胡嘉祥