



吉林大学

本科生毕业论文（设计）

（双学士学位学位）

中文题目 基于树莓派的垃圾识别系统

英文题目 Garbage Recognition System based on
Raspberry Pi

学生姓名 艾泽凯 教学号 学号 14190219

指导教师 董劲男 职称 讲师

双学士学位学院 计算机科学与技术学院

双学士学位专业 计算机科学与技术

主修学院 机械与航空航天工程学院

主修专业 机械工程

摘要

基于树莓派的垃圾识别系统

当前垃圾分类已经成为城市的主流趋势，但是目前垃圾分类仍然主要由人力完成，这样存在着人力成本高、分类慢、分拣人员易受伤等问题，为了处理这个问题，我设计了这个基于 SSD 算法 MobileNet V3 模型的垃圾识别系统，同时将这个系统运用在树莓派上，这样方便对现有垃圾桶进行改装，同时做到价格低廉。同时为了方便监测树莓派的状态，我设计了将手机与树莓派连接，从而可以通过手机查看树莓派里的文件，调用树莓派的摄像头以查看周围环境，监控树莓派的温度防止高温烧坏主板，监测树莓派 CPU 的使用情况等。本设计可以较好地做好垃圾识别工作，并且价格低廉，装置小巧，具有较好的市场前景。

关键字：树莓派，SSD 算法，MobileNet V3 模型，TensorFlow

Abstract

Garbage Recognition System based on Raspberry Pi

Garbage sorting has become the mainstream trend in cities, but currently, it is mainly done manually, which leads to problems such as high labor costs, slow sorting, and a higher risk of injuries to sorting personnel. In order to address this issue, I have designed a garbage recognition system based on the SSD algorithm and MobileNet V3 model. This system is implemented on a Raspberry Pi, which allows for convenient retrofitting of existing trash bins at a low cost. Additionally, I have designed a connection between a mobile phone and the Raspberry Pi to monitor the Raspberry Pi's status. This connection enables users to view files, access the Raspberry Pi's camera for environmental monitoring, prevent overheating of the mainboard by monitoring temperature, and track CPU usage. This design effectively improves garbage recognition, offers an affordable price, features a compact device, and has good market potential.

Keywords:Raspberry Pi, SSD algorithm, MobileNet V3 model,
TensorFlow.

目 录

第 1 章 绪论	1
1.1 研究背景与意义	1
1.2 国内外研究现状	1
1.3 论文研究内容与组织框架	3
第二章 深度学习基本介绍	4
2.1 卷积神经网络概述	4
2.1.1 卷积层	4
2.1.2 激活函数	5
2.1.3 池化层	5
2.1.4 全连接层	6
2.2 常见深度学习目标检测算法	7
2.2.1 R-CNN 算法	7
2.2.2 YOLO 算法	8
2.2.3 SSD 算法	8
2.3 常见卷积神经网络深度学习模型	9
2.3.1 AlexNet 算法	9
2.3.2 ResNet 算法	10
2.3.3 VGGNet 算法	10
2.3.4 LeNet-5 算法	11
2.3.5 MobileNet 算法	12
第三章 基于 SSD MOBILENET V3 神经网络的垃圾分类方法	13
3.1 SSD 算法详细介绍	13
3.1.1 SSD 算法的主要特点和结构	13
3.1.2 SSD 算法的工作原理和主要步骤	14

3.2 MOBILENET V3 模型详细介绍	15
3.2.1 网络结构设计	15
3.2.2 模型优化技术	15
3.3 模型训练过程	16
第四章 基于神经网络的垃圾分类方法的硬件实现	18
4.1 硬件介绍	18
4.1.1 树莓派 4B	18
4.1.2 树莓派摄像头	19
4.2 树莓派详细配置过程	20
4.2.1 树莓派系统烧录	20
4.2.2 树莓派与电脑有线连接	20
4.2.3 树莓派与电脑无线连接	24
4.2.4 树莓派 OpenCV 安装	26
4.2.5 手机端控制树莓派	27
第五章 实验检测	29
5.1 模拟检测实验	29
第六章 总结与展望	30
6.1 总结	30
6.2 展望	31
参考文献	32
致谢	34

第 1 章 绪论

1.1 研究背景与意义

当前垃圾分类已经在许多城市普及，但是很多小区应对垃圾分类的办法仅仅只是聘请了新的一些人来专门进行垃圾分类，这样提高了小区居民的生活成本，同时也浪费了一些不必要的劳动力。由于没有对垃圾进行初步分类，整个分类的过程将非常缓慢，同时垃圾里的一些比如剪刀、小刀、玻璃碎片等危险品又容易对垃圾分拣者造成伤害，所以我考虑设计了这款辅助分拣设备，可以协助识别出厨余垃圾、电子设备垃圾和容易划伤人的垃圾，这样可以大大提高分拣的效率，节约劳动力。

为了使此装置便于安装且成本不至于过高，我考虑以树莓派作为程序的载体，这样可以基本保证程序的运行速度，又不至于使得垃圾桶的外设过大。

1.2 国内外研究现状

当前深度学习与神经网络无疑计算机领域的大热门。

其中神经网络的早期发展可以追溯到感知机模型^[1]，感知机是由 Frank Rosenblatt 在 1957 年提出的简单的神经网络模型，只能解决线性可分问题，无法解决复杂的非线性问题；在 1980 年代，研究人员开始提出多层感知机，它通过引入一个或多个隐藏层来解决感知机的线性限制问题，从而能够处理更复杂的非线性模式；之后在 1986 年，David Rumelhart 等人提出了反向传播算法^[2]，该算法利用链式法则来计算神经网络中的梯度，从而进行有效的训练；在 1990 年代，由于计算能力和数据集规模的限制，神经网络的研究相对较少，支持向量机和核方法成为主流的机器学习方法。从 2006 年至今，Hinton 等人提出了深度学习信念网络^[3]，为深度学习提供了一种新的训练方法，之后又出现了许多突破，例如卷积神经网络在图像处理领域得到了广泛应用，循环神经网络在序列数据处理中得到了广泛应用。

而深度学习的发展历程可以追溯到上世纪 50 年代神经网络研究，但是直到最近几十年，随着计算能力和数据量的增加，深度学习才得以蓬勃发展。在 1980 年

代出现的多层感知机是深度学习的基础模型的其中之一。它引入了多个隐藏层从而解决了感知机的线性限制问题，从而能够处理复杂的非线性问题。但是受限制与计算能力和数据集规模，深层网络的训练受到了挑战；在 1986 年出现的反向传播算法是深度学习的重要突破之一，由 David Rumelhart 等人提出的反向传播算法^[4]利用链式法则计算神经网络中的梯度，从而实现深层网络的有效训练；于 1997 年产生了长短期记忆网络^[5]，长短期记忆网络是一种循环神经网络的变种，它的提出解决了传统循环神经网络在处理长序列时遇到的梯度消失和梯度爆炸问题，长短期记忆网络的引入使得深度学习在处理序列数据，例如机器翻译和语言模型方面取得了显著的突破；于 1998 年出现的卷积神经网络是一种专门用于处理视觉和图像任务的深度学习模型，同年 Yaan LeCun 等人提出了 LeNet-5 模型^[6]，将卷积层和池化层引入了神经网络，这使网络能够有效提取图像中的特征；2006 年出现的深度信念网络^[7]是深度学习发展的重要里程碑，Geoffrey Hinton 等人于 2006 年提出了深度信念网络，该网络利用有监督微调和无监督预训练的方式训练神经网络，为后来的深度学习算法奠定了基础；2012 年，Alex Krizhevsky 等人提出了 AlexNet 卷积神经网络模型^[8]，AlexNet 在 ImageNet 图像分类取得了非常显著优势，引起了深度学习在计算机视觉领域的广泛关注，从此，深度学习在图像分类、物体检测和图像生成等任务表现不断超越传统方法；深度学习在自然语言处理领域也取得了重要突破，2014 年，Google 研究团队提出了基于深度学习的神经机器翻译模型，这个模型的成功标志着深度学习在自然语言处理领域的广泛应用；深度学习也在强化学习领域取得了突破，在 2015 年，DeepMind 的 AlphaGo 击败了世界冠军李世石，AlphaGo 使用了深度神经网络和强化学习，体现了深度学习在复杂决策问题中的潜力；近年来，深度学习在自动驾驶、智能助手和语音识别等领域持续取得重大进展，自动驾驶技术借助了深度学习方法实现了感知、决策和控制等关键功能，这大大推动了自动驾驶车辆的发展，智能助手和语音识别系统也广泛使用深度学习来实现自然语言的理解和语音处理；近年来，迁移学习和预训练模型在深度学习中扮演着重要角色，迁移学习可以利用已经训练好的模型的知识来加速和改善新任务的训练，预训练模型可以通过大规模数据进行预训练，从而学习到丰富的语义和语言模式，在特定任务上进行微调，取得了显著的效果，这种方法在自然语言处理和计算机视觉等领域得到了广泛的应用；随着深度学习的发展，它在各个领域都取得了广泛应用，在医疗领域，深度学习在疾病诊断、医

学影像分析与药物研发等方面都发挥着重要作用，在金融领域，它在风险评估、欺诈检测和量化交易等方面也得到了应用，同时深度学习在推荐系统、广告投放和用户个性化等领域也发挥着重要的作用。

树莓派是一款基于单片机的小型计算机平台，被广泛用于教育、嵌入式系统和创客项目等领域。2012 年，初代树莓派由 Raspberry Pi 基金会开发，此款树莓派搭载 ARM11 处理器，初代树莓派得到了广泛关注和认可，因为它低廉的价格和开放的设计而收到欢迎；Raspberry Pi 2 于 2015 年发布，性能较初代树莓派有了显著提升；而 Raspberry Pi 3 于 2016 年发布，性能进一步增强，增加了无线网络和蓝牙功能；Raspberry Pi Zero 于 2015 年发布，采用了与 Raspberry Pi 1 相同的芯片，但是更小也更便宜，Raspberry Pi Zero 适用于资源受限和对尺寸有要求的项目；2019 年诞生的 Raspberry Pi 4 是目前最强大的树莓派版本，采用了 64 位的处理器，配备了 1GB、2GB 或 4GB 的 LPDDR4 内存，Raspberry Pi 4 还提供了双 HDMI 输出、USB 3.0 接口、Gigabit 以太网等更多的功能和扩展选项。

1.3 论文研究内容与组织框架

本文由六个章节组成，第一个章节为绪论，介绍了研究背景与意义，国内外的研究现状以及论文研究内容与组织框架。

第二章是对深度学习的基本介绍，简要介绍了卷积神经网络，还简要介绍了一些常见的深度学习目标检测算法以及常见卷积神经网络深度学习模型。

第三章是对基于 SSD MobileNet V3 神经网络的垃圾分类方法的详细介绍，详细介绍了 SSD 算法以及 MobileNet V3 模型，同时还对模型训练过程进行了一定的介绍。

第四章是关于基于神经网络的垃圾分类方法的硬件实现，先简要介绍了一下树莓派和树莓派摄像头，之后介绍了树莓派上所需要的配置过程例如烧录系统、连接电脑、安装 OpenCV、使用手机控制树莓派。

第五章是关于识别效果的检测，对分类效果进行了模拟实验。

第六章是总结与展望，对本设计的优势与不足进行了总结，同时提出了展望，指出了未来改进的方向！

第二章 深度学习基本介绍

2.1 卷积神经网络概述

卷积神经网络是一种在计算机视觉和图像处理领域中广泛应用的人工神经网络模型，卷积神经网络以其在图像识别、目标检测和图像分类等任务中的出色表现而闻名，它的核心思想是模拟人脑视觉系统中的处理方式，通过多个卷积层、池化层和全连接层构成，从输入图像中提取特征，并通过训练过程学习对图像进行分类或其它任务的模式。

2.1.1 卷积层

卷积层是卷积神经网络的核心组件之一，主要负责从输入数据中提取特征并生成特征图。卷积操作是种局部感受野的运算，通过使用一组可学习的滤波器对输入数据以滑动窗口的方式进行计算，每个滤波器都有一组权重参数，它定义了特定的特征检测器，滤波器可以在输入数据上进行逐元素相乘并求和的操作，从而产生一个输出值。

卷积层中，各个滤波器都会与输入数据的进行卷积运算。通过滤波器在输入数据上的滑动，卷积层可以检测到不同位置的局部特征，这些特征可以是边缘、纹理或颜色，在卷积操作中，滤波器的权重参数会根据训练结果进行学习调整，这使得卷积层能够自动提取有效特征。

卷积操作的输出成为特征图，其尺寸取决于输入数据尺寸、滤波器大小以及卷积的步幅和填充设置，其中步幅定义了滤波器每次移动时的跨度，填充可以在输入数据周围添加额外像素，从而保持输出特征图尺寸与输入数据一致。卷积层中通常包含了多个滤波器，每个滤波器可以检测得到不同特征，使用多个滤波器可以使卷积层生成生成多个特征图，每个特征图都可以对应一个滤波器，通过这种方式，卷积层就可以同时检测多个不同的特征，并且每个特征图中的像素值可以表示相应位置特征在输入数据中的存在程度。

2.1.2 激活函数

激活函数在神经网络中扮演着重要角色，它引入了非线性变换，这使得神经网络可对复杂数据进行建模，激活函数引入了非线性变换，这使得神经网络可以对复杂数据进行建模学习。在神经网络中，常见的激活函数有 ReLU、sigmoid 和 tanh 等。

ReLU 函数：ReLU 函数定义为 $f(x) = \max(0, x)$ ，即输入小于 0 时输出为 0，输入大于等于 0 时输出等于输入值。ReLU 函数具有简单的计算形式和非线性特性，它能够在保留正数输入的信息的同时，抑制负数输入，增强网络的表达能力。此外，ReLU 函数的导数在正数输入时恒定为 1，避免了梯度消失的问题。

Sigmoid 函数：Sigmoid 函数定义为 $f(x) = 1 / (1 + \exp(-x))$ ，它将输入映射到 0 和 1 之间的连续值。Sigmoid 函数在早期的神经网络中经常使用，它具有良好的平滑性和可导性。然而，Sigmoid 函数在输入较大或较小时，梯度接近于 0，导致梯度消失的问题，限制了网络的深度学习能力。

Tanh 函数：Tanh 函数定义为 $f(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x))$ ，它将输入映射到 -1 和 1 之间的连续值。Tanh 函数具有 Sigmoid 函数的平滑性和可导性，但相比 Sigmoid 函数，Tanh 函数在输出值的范围上更大，并且具有零中心性质（即均值接近于 0），有利于网络的训练和收敛。

选择适当的激活函数取决于具体的任务和网络架构。通常情况下，ReLU 函数是最常用的激活函数，因为它具有较好的性能和计算效率。当存在输出需要限制在特定范围内时，例如二分类问题中的输出概率，可以使用 Sigmoid 函数。而在需要对输出进行归一化或对称处理时，可以选择 Tanh 函数。

2.1.3 池化层

池化层（Pooling Layer）是卷积神经网络中的一种常用层，用于减小特征图的空间尺寸、降低计算复杂度并提取最显著的特征。池化操作通过在输入数据的局部区域上进行采样或统计汇聚，来减少特征图的维度。常见的池化操作有最大池化和平均池化。

最大池化是从输入数据的局部区域中选择最大的特征值作为输出。它的主要目的是保留最显著的特征，同时减小特征图的空间尺寸。最大池化通过滑动固定大小的窗口（通常为 2×2 ）在输入数据上进行采样，然后取窗口内的最大值作为输出。这样可以减小特征图的尺寸，同时保留最强的特征。

平均池化是从输入数据的局部区域计算平均值作为输出。它的主要目的是对特征进行平滑和降维，减少计算量。平均池化操作与最大池化类似，不同之处在于取窗口内的平均值作为输出。

池化层通常在卷积层之后应用，可以多次堆叠使用。池化操作具有一定的不变性，即对输入数据的微小变化具有一定的鲁棒性。这使得池化层能够对输入特征进行不变性的学习，从而提高模型的鲁棒性和泛化能力。

池化层的一些重要参数包括池化窗口大小、步幅和填充方式。池化窗口大小决定了池化操作采样的区域大小，步幅定义了窗口滑动的跨度，而填充可以在输入数据周围添加额外的像素，以保持输出特征图的尺寸或进行调整。

2.1.4 全连接层

全连接层，也称为密集连接层，是卷积神经网络中的一种基本层类型。全连接层的目的是将前一层的所有神经元与当前层中的每个神经元相连接，实现特征的组合和分类。在全连接层中，每个神经元都与前一层的所有神经元相连。每个连接都有一个权重参数，用于调整输入特征对当前层的影响。此外，每个神经元还有一个偏置项，用于引入偏移量。全连接层的输入是前一层的输出，通常是经过卷积层和池化层处理后的特征图。输入特征被展平成一维向量，以便与全连接层中的神经元进行连接。每个神经元都执行以下操作：接收来自前一层的输入特征向量；将输入特征向量与相应的权重参数进行元素乘法，并将结果累加；添加偏置项；应用激活函数对结果进行非线性变换；将变换后的结果作为当前层的输出。

全连接层的输出可以作为最终的分类结果或传递给其他层进行进一步处理。通常，在全连接层之后会添加一个或多个全连接层，以逐渐提高模型的表达能力和复杂性。全连接层的主要作用是实现特征的组合和高级表示，从而使网络能够学习更复杂的模式和关系。通过大量的参数学习和非线性激活函数的引入，全连

接层能够处理输入数据之间的复杂关系，并生成适合特定任务的输出。然而，全连接层的参数数量很大，导致模型的计算量和内存需求较高。为了解决这个问题，常常采用降维技术（如卷积层和池化层）来减少特征维度，以及正则化方法来控制过拟合。

2.2 常见深度学习目标检测算法

2.2.1 R-CNN 算法

R-CNN^[9]是一种经典的深度学习目标检测算法。它是目标检测领域的里程碑之一，为后续的目标检测算法奠定了基础。R-CNN 算法的核心思想是将目标检测问题分解为两个子任务：候选区域提取和目标分类与定位。它通过候选区域建议（Region Proposal）方法选择一组可能包含目标的区域，并使用卷积神经网络（CNN）进行特征提取和目标分类。

R-CNN 算法的详细步骤如下：使用选择性搜索（Selective Search）等传统计算机视觉方法，从输入图像中提取一组可能包含目标的候选区域。这些候选区域通常是不同尺度和长宽比的矩形框；对于每个候选区域，将其裁剪并调整为固定大小的图像块。然后，将这些图像块输入预训练的卷积神经网络（如 AlexNet、VGG 等）中，提取相应的特征表示；使用提取的特征作为输入，通过全连接层进行目标分类和边界框的回归。全连接层包括分类器和回归器，分类器用于预测每个候选区域的目标类别，回归器用于预测目标边界框的位置；根据目标分类的置信度，对候选区域进行筛选和排序，选择置信度较高的候选区域作为最终的目标检测结果。

R-CNN 算法的优点在于它能够通过使用预训练的卷积神经网络从图像中提取有用的特征，从而在目标分类和定位任务中取得较好的性能。它还能够处理不同尺度和长宽比的目标，并且在多个目标之间共享计算，提高了检测速度。然而，R-CNN 算法也存在一些缺点。首先，它的训练过程比较复杂和耗时，因为需要进行多个阶段的训练。其次，由于需要对每个候选区域进行独立的特征提取和分类，导致算法在速度上相对较慢。此外，候选区域提取阶段使用传统方法，可能存在一定的误检和漏检问题。尽管 R-CNN 算法在目标检测领域取得了显著的进展，但后续算法如 Fast R-CNN、Faster R-CNN 和 Mask R-CNN 等对其进行了改进和优

化，进一步提高了目标检测的准确性和效率。

2.2.2 YOLO 算法

YOLO^[10]是一种快速而准确的深度学习目标检测算法。相比于传统的目标检测算法，YOLO 将目标检测任务转化为一个回归问题，通过单个神经网络直接在图像上进行密集预测，从而实现实时目标检测。

YOLO 算法的详细步骤如下：网络划分：将输入图像划分为固定大小的网格单元，每个网格单元负责检测该单元中是否存在目标。通常，YOLO 将图像划分为 $S \times S$ 的网格；锚框设计：对于每个网格单元，预定义一组锚框（Anchor Box），用于检测不同尺度和长宽比的目标。每个锚框与该网格单元关联，负责预测该网格单元中的目标；特征提取：通过卷积神经网络（CNN）对输入图像进行特征提取。YOLO 通常使用卷积层和池化层构建特征提取网络，以获取图像的高级语义特征；特征映射：将特征图与网格单元关联起来，每个网格单元负责预测锚框中目标的类别和位置信息。通过卷积层输出的特征图可以计算每个网格单元的目标置信度（包括目标类别的概率）和边界框的位置和尺寸；预测解码：根据特征映射的结果，对目标置信度进行筛选和非极大值抑制，以获取最终的目标检测结果。非极大值抑制排除重叠较大的候选框，并保留具有较高置信度的框。

YOLO 算法的特点在于其实时性和高效性。由于 YOLO 在单个神经网络中进行密集预测，避免了候选区域提取和多次推理的过程，因此具有更快的检测速度。此外，YOLO 在处理小目标方面具有优势，能够捕捉到小尺寸目标的细节信息。然而，YOLO 算法也存在一些限制。由于网络划分和锚框设计的限制，YOLO 可能在检测小尺寸目标和目标密集场景中表现不佳。此外，由于 YOLO 的密集预测方式，它在处理重叠目标时可能存在定位偏差。为了改进 YOLO 算法，研究者提出了一系列改进版，如 YOLOv2、YOLOv3 和 YOLOv4。这些改进版在网络架构、特征提取、损失函数等方面进行了创新，以提高目标检测的准确性和性能。

2.2.3 SSD 算法

SSD^[11]是一种流行的深度学习目标检测算法，它能够快速且准确地检测图像中的目标。SSD 通过在不同层级的特征图上应用不同尺度的预定义锚框，实现了多

尺度的目标检测。SSD 算法的优点在于它能够在单次前向传播中实现目标检测，因此具有较快的检测速度。同时，SSD 能够处理不同尺度和长宽比的目标，并且在多个特征层级上进行检测，提高了检测的准确性。然而，SSD 算法也存在一些限制。由于锚框的固定设计，可能导致在小目标和大目标之间存在检测不均衡的问题。此外，较低分辨率的特征图可能对小目标的定位精度不够准确。

2.3 常见卷积神经网络深度学习模型

2.3.1 AlexNet 算法

AlexNet 算法^[12]是由 Alex Krizhevsky、Ilya Sutskever 和 Geoffrey Hinton 在 2012 年提出的卷积神经网络模型。它是深度学习在计算机视觉领域取得突破性成果的重要里程碑。AlexNet 在 ImageNet 图像分类挑战赛上取得了显著的优势，大大推动了深度学习的发展。AlexNet 的主要特点和结构如下：

深层网络结构：AlexNet 是一个较为深层的卷积神经网络模型，包含 8 个卷积层和 3 个全连接层。相较于之前的模型，AlexNet 引入了更多的层级，提升了网络的表示能力；**卷积层和池化层：**AlexNet 的卷积层采用较大的卷积核尺寸，如 11×11 和 5×5 ，以便捕捉更大范围的特征。在卷积层之后，使用最大池化层来减小特征图的尺寸，并且引入局部响应归一化进行特征增强；**ReLU 激活函数和 Dropout：**在网络的每个卷积层和全连接层之后，采用了非线性激活函数 ReLU 来引入更好的非线性表达能力。此外，在全连接层之间添加了 Dropout 层，以减轻过拟合问题；**数据增强和训练技巧：**为了增加数据的多样性和模型的泛化能力，AlexNet 采用了数据增强技术，如随机裁剪、水平翻转和色彩变换等。此外，使用了随机梯度下降（SGD）进行网络训练，并应用了学习率衰减和动量等技巧；**多 GPU 并行训练：**AlexNet 是在多个 GPU 上并行训练的，这种多 GPU 并行的训练方式加速了网络的训练过程，并且能够处理更大规模的图像数据集。

AlexNet 的突破性在于它展示了深度学习在计算机视觉任务中的巨大潜力。通过使用更深的网络结构、更大的数据集和更强的计算能力，AlexNet 在 ImageNet 图像分类挑战赛上取得了显著的准确率提升，标志着深度学习在计算机视觉领域的广泛应用。其结构和思想也对后续的深度学习模型设计产生了重要影响，激发了更多的研究人员对深度学习的兴趣和探索。AlexNet 的成功不仅是由于其网络结

构的创新，还得益于其训练技巧和数据增强策略的运用。AlexNet 作为一种经典的卷积神经网络模型，为深度学习在计算机视觉领域的发展铺平了道路，为后续更深、更强大的网络模型奠定了基础，对于深度学习的研究和应用具有重要的意义。

2.3.2 ResNet 算法

ResNet 算法^[13]是由 Kaiming He 等人在 2015 年提出的深度残差网络模型，它在解决深层神经网络训练中的梯度消失和梯度爆炸问题上取得了重大突破。ResNet 的设计思想是通过引入残差连接（residual connection），允许网络中的信息直接跳过一个或多个层级，从而有效地训练更深的网络。

ResNet 的基本组成单元是残差块。一个残差块由两个或三个卷积层组成，每个卷积层之后都有一个跳跃连接将输入直接添加到输出中。这样的设计使得网络可以学习残差映射，即网络只需要学习输入与输出之间的残差部分，简化了网络的学习过程。残差连接允许信息在网络中直接跳过一些层级，避免了信息在网络中传递时的衰减。通过跳跃连接，残差块可以学习到输入的细微变化，从而更好地保留了输入的信息。

为了减少参数量和计算量，ResNet 采用了瓶颈结构。瓶颈结构通过在残差块中引入 1x1 的卷积核来减少通道数，然后再进行 3x3 的卷积操作。这样可以在保持网络深度的同时减少计算复杂度。ResNet 的网络层级可以根据任务的复杂性进行选择，常见的有 ResNet-18、ResNet-34、ResNet-50、ResNet-101 和 ResNet-152 等。这些模型的深度不同，其中的数字代表了网络中残差块的数量。

ResNet 的创新之处在于通过残差连接解决了深层网络中的梯度消失和梯度爆炸问题，使得可以训练非常深的网络。相比于之前的模型，ResNet 在 ImageNet 图像分类挑战赛上取得了更高的准确率，并且在其他计算机视觉任务中也取得了重要的成果。

2.3.3 VGGNet 算法

VGGNet^[14]是由 Simonyan 和 Zisserman 在 2014 年提出的卷积神经网络模型。它在深度学习领域具有重要的影响，其特点是网络结构相对简单而均匀，通过增加网络深度来提升模型的性能。

VGGNet 的网络结构非常规整，使用了相同大小的卷积核（通常是 3×3 ）和池化层（通常是 2×2 的最大池化），并且连续堆叠多个卷积层。这种简单而均匀的结构使得网络更容易理解和实现。VGGNet 中的池化层起到了降低特征图尺寸和参数的作用，帮助提取更高级别的特征。通常在相邻的卷积层之后进行最大池化操作，将特征图的尺寸减半。

VGGNet 提供了不同深度的模型选择，包括 VGG-11、VGG-13、VGG-16 和 VGG-19 等。这些模型的数字表示网络中卷积层的数量，数字越大表示网络越深。深层的模型可以提取更丰富的特征，但也增加了计算复杂度。在卷积层之后，VGGNet 使用了三个全连接层来进行最终的分类。这些全连接层具有较大的参数量，因此 VGGNet 的模型较大。

VGGNet 在 ImageNet 图像分类挑战赛上取得了很好的成绩，并成为深度学习中的重要里程碑。它的设计思想简单而直观，通过增加网络深度来提升模型性能，这种思路对后续的深度卷积神经网络模型设计产生了重要影响。

2.3.4 LeNet-5 算法

LeNet-5^[6]是由 Yann LeCun 等人在 1998 年提出的卷积神经网络模型，是深度学习在计算机视觉领域的里程碑之一。LeNet-5 的设计目标是实现手写数字识别，其结构简单而有效。

LeNet-5 由多个卷积层和池化层交替组成。卷积层使用小尺寸的卷积核（通常是 5×5 ）来提取特征，然后通过池化层进行下采样。这样可以逐渐减小特征图的尺寸，并且增强对平移不变性的学习。LeNet-5 使用 Sigmoid 作为激活函数。虽然在后续的模型中更常使用 ReLU 激活函数，但在 LeNet-5 中，Sigmoid 被广泛应用。

LeNet-5 引入了局部响应归一化（Local Response Normalization）层，在卷积层之后进行特征增强。该操作可以抑制相邻神经元的竞争，增强特征的鲁棒性。在卷积层和池化层之后，LeNet-5 使用了多个全连接层来进行最终的分类。全连接层连接到每个神经元，并通过激活函数进行非线性变换。

LeNet-5 的创新之处在于引入了卷积神经网络在计算机视觉任务中的应用，提出了卷积层、池化层和局部响应归一化的概念，为后续深度学习模型的发展奠定了基础。虽然 LeNet-5 相对较浅且用于手写数字识别等简单任务，但它证明了卷积神经网络的有效性和潜力，为后续更复杂的网络模型的设计提供了启示。尽管

LeNet-5 的设计思想和结构相对简单，但它在计算机视觉领域的成功应用标志着深度学习崛起，并为后续更深、更强大的网络模型的发展铺平了道路。

2.3.5 MobileNet 算法

MobileNet^[15]是由 Google 的研究团队在 2017 年提出的一种轻量级卷积神经网络模型，旨在在移动设备等资源受限的环境下进行高效的图像识别和处理。MobileNet 通过精心设计的深度可分离卷积结构，在减少参数量和计算复杂度的同时，保持较高的模型准确性。

MobileNet 使用了深度可分离卷积来替代传统的标准卷积操作。深度可分离卷积分为两个步骤：深度卷积和逐点卷积。深度卷积在每个输入通道上独立进行卷积操作，然后通过逐点卷积将通道维度的特征混合在一起。这种结构可以显著减少参数量和计算量。MobileNet 中的瓶颈结构类似于 ResNet 中的思想，通过在深度可分离卷积之前和之后添加 1×1 的卷积操作，减少通道数，并降低计算复杂度。

MobileNet 通过调整网络的宽度参数来平衡模型的性能和计算复杂度。通过控制通道数的比例，可以在准确性和模型大小之间进行权衡。为了进一步提升模型性能，MobileNet 引入了快捷连接（Shortcut Connection）来帮助信息的流动。在网络中的某些层级上，将输入直接连接到输出，保留更多原始信息。

MobileNet 的设计目标是在资源受限的环境下提供高效的图像处理。相比于传统的卷积神经网络模型，MobileNet 具有更小的模型尺寸和更低的计算复杂度，适合在移动设备等资源受限的场景中应用。它在图像分类、物体检测和语义分割等任务上取得了很好的性能。后续的 MobileNet V2 和 MobileNet V3 在 MobileNet 的基础上进行了进一步的改进和优化，提升了准确性和效率。MobileNet 系列模型的成功证明了轻量级卷积神经网络在资源受限环境下的价值，为移动端和嵌入式设备上的深度学习应用提供了重要参考。

第三章 基于 SSD MobileNet V3 神经网络的垃圾分类方法

3.1 SSD 算法详细介绍

SSD 是一种用于目标检测的卷积神经网络算法，由 Wei Liu 等人于 2016 年提出。相对于传统的目标检测算法，SSD 具有高效性和准确性的特点，能够在单个前向传播过程中实现目标检测，因此称为"Single Shot"。

3.1.1 SSD 算法的主要特点和结构

SSD 通过在不同层级的特征图上进行检测，能够有效地检测不同尺寸的目标。它从骨干网络（如 VGGNet 或 ResNet）中提取多个特征图，每个特征图具有不同的分辨率和语义级别。

SSD 使用了锚框来定位和识别目标物体。在每个特征图上，SSD 通过在不同位置和比例上生成一组预定义的锚框，用于表示不同大小和形状的目标物体。这些锚框与实际目标进行匹配，预测目标的位置和类别。SSD 通过在不同层级的特征图上进行多层预测，能够对不同大小的目标进行检测。较低层级的特征图适用于检测较小的目标，而较高层级的特征图适用于检测较大的目标。这种多层预测的策略使得 SSD 能够处理各种尺寸的目标物体。

SSD 使用了多个损失函数来训练网络，包括目标类别损失和目标框位置损失。目标类别损失使用交叉熵损失函数，用于预测目标的类别标签。目标框位置损失使用 Smooth L1 损失函数，用于准确地预测目标的位置。在目标检测的最终阶段，SSD 使用非极大值抑制来消除重叠的候选框，保留最具代表性的目标框。这样可以提高检测结果的准确性，并减少重复检测的问题。

SSD 算法在目标检测领域取得了很好的性能，具有快速和准确的特点。它能够实现端到端的目标检测，不需要额外的候选框生成过程，从而提高了检测速度。SSD 在各种应用场景中都表现出色，如行人检测、车辆检测、物体识别等。相比于其他目标检测算法，SSD 在处理小目标和多尺度目标时具有明显的优势。其多层级的特征提取和多尺度预测能力使其能够捕捉不同大小和形状的目标。

SSD 算法的一个主要优点是速度快，能够实时处理图像或视频流。这得益于

其使用的卷积神经网络结构和多尺度特征图检测策略，使得目标检测可以在单个前向传播中完成，而无需额外的候选框生成过程。这使得 SSD 成为在嵌入式设备、移动设备和实时应用中广泛使用的目标检测算法。然而，SSD 也存在一些限制。由于其设计的深度可分离卷积结构，SSD 在处理更细节和更复杂的场景时可能会丧失一定的准确性。此外，SSD 对于目标的密集排列和重叠较多的情况，如群体场景或遮挡情况，可能会有一定的检测漏报或误报。

3.1.2 SSD 算法的工作原理和主要步骤

SSD 使用一个预训练的卷积神经网络（如 VGGNet 或 ResNet）来提取图像的特征。这些特征图具有不同的分辨率和语义级别，对于不同大小和形状的目标物体具有不同的表示能力。

SSD 对特征图进行处理，以获得不同尺度的特征图用于目标检测。通过在原始特征图上应用一系列卷积层和池化层，可以生成一系列具有不同尺度的特征图。在每个特征图上，SSD 通过在不同位置和比例上生成一组锚框来表示目标的位置和形状。每个锚框都与特征图的一个像素位置相关联，并且具有预定义的不同尺度和长宽比。对于每个锚框，SSD 通过卷积操作来预测目标的类别和位置。分类操作用于确定锚框中是否存在目标物体以及其所属的类别，回归操作用于精确定位目标的边界框。

SSD 通过在不同层级的特征图上进行多层预测，能够检测不同大小的目标。较低层级的特征图适用于检测较小的目标，而较高层级的特征图适用于检测较大的目标。同时，SSD 还通过特征融合的方式，将不同层级的特征图进行结合，以提高目标检测的准确性和鲁棒性。在目标检测的最终阶段，SSD 使用非极大值抑制来消除重叠的候选框，保留最具代表性的目标框。NMS 通过计算候选框之间的重叠程度，并基于一定的阈值来判断是否保留候选框。这样可以消除重复检测和减少误报。

通过以上步骤，SSD 能够实现快速而准确的目标检测。它在多个数据集上取得了很好的性能，在物体检测、行人检测、车辆检测等领域具有广泛应用。同时，SSD 的结构灵活性较高，可以根据需求进行定制化的修改和扩展，例如通过改变特征提取网络、调整锚框的比例和数量，或者引入其他改进技术来进一步提升性能。

3.2 MobileNet V3 模型详细介绍

MobileNet V3 是一种轻量级的卷积神经网络模型，用于图像分类和目标检测任务。它是 Google 团队于 2019 年提出的，旨在保持高准确性的同时，减少模型的计算量和参数数量。MobileNet V3 的设计主要注重于两个方面：网络结构的设计和模型优化技术。

3.2.1 网络结构设计

MobileNet V3 采用了一种基于倾斜分离卷积的模块化设计。这种卷积操作分为两步：深度卷积和逐点卷积。在深度卷积阶段，MobileNet V3 使用轻量级的深度卷积核来对每个输入通道进行卷积操作，减少了计算量。逐点卷积是一个 1×1 卷积操作，用于将深度卷积输出的特征图进行通道间的线性组合，以提取更高维度的特征。MobileNet V3 通过多个模块的堆叠来构建整个网络。其中，每个模块都包括多个倾斜分离卷积操作，配合激活函数和规范化操作，构成了一个非常轻量级但有效的网络结构。

3.2.2 模型优化技术

MobileNet V3 引入了一些模型优化技术，以进一步减少模型的计算量和参数数量，同时提高模型的性能。

MobileNet V3 通过调整网络的宽度参数，可以在准确性和计算量之间进行权衡。通过控制每个层级的通道数，可以灵活地调整模型的大小和计算量。它使用了一种自适应线性单元激活函数，称为 Hard-Swish。它在计算效率和准确性之间找到了一个平衡点，能够提供更好的非线性表示能力。

MobileNet V3 中的某些模块采用了倒残差结构，即先通过逐点卷积将输入通道扩展，然后再进行深度卷积操作，最后通过逐点卷积将通道降维。这种结构有助于提升特征表达能力，并减少参数数量和计算量。

MobileNet V3 在图像分类和目标检测任务中都取得了很好的性能。它在准确性方面与其他更复杂的模型相媲美，但具有更小的模型尺寸和计算复杂度。相对于前几个版本的 MobileNet，MobileNet V3 进一步提高了模型的准确性和效率。

MobileNet V3 还提供了多个不同的变体，以适应不同的应用场景和资源限制。其中包括 MobileNetV3-Large 和 MobileNetV3-Small 等变体，分别针对高准确性和低计算量的需求。

MobileNet V3 在计算机视觉领域的许多任务中都具有广泛的应用，例如图像分类、目标检测、语义分割等。它在移动设备、嵌入式设备和边缘计算等资源受限的环境中表现出色，能够在保持较高准确性的同时，实现实时的推理速度。

3.3 模型训练过程

为了训练基于 SSD 算法的 MobileNet V3 模型，首先我从 COCO 官方网站上下载了 COCO 数据集的图像和标注文件，COCO 数据集是一个广泛使用的计算机视觉数据集，用于目标检测、图像分割和关键点检测等任务。

在训练之前，我还对图像数据进行了预处理，预处理过程包括了图像缩放、均值减去、标准化等，同时我利用图像缩放将图像调整到了固定的尺寸，以适应模型的输入要求。其中均值减去是指从图像的每个通道中减去均值，以减少输入数据的偏差。标准化可以将图像的像素值进行归一化，使其落在一个较小的范围内，有助于提高模型的训练效果。

第二步是模型构建，由于树莓派算力较低，基于轻量化的想法，我最终选择了 MobileNet V3 模型以及 SSD 算法，使用了 TensorFlow 来构建模型，安装了 TensorFlow Object Detection API，之后我配置了包括数据路径、模型路径和训练参数等，选择了 SSD 模型的配置文件，其中指定了训练参数、模型结构和数据路径等信息，之后就是具体的训练过程，训练完成后使用了 TensorFlow 提供的 `export_inference_graph.py` 脚本导出模型。

第三步是模型转换，为了将模型转化为可以被树莓派调用，需要将 TensorFlow 训练出来的模型转换为.pb 文件和.pbtxt 文件，我使用 TensorFlow 提供的 `exporter_main_v2.py` 脚本完成了此操作。完整过程如图 3-1 所示。

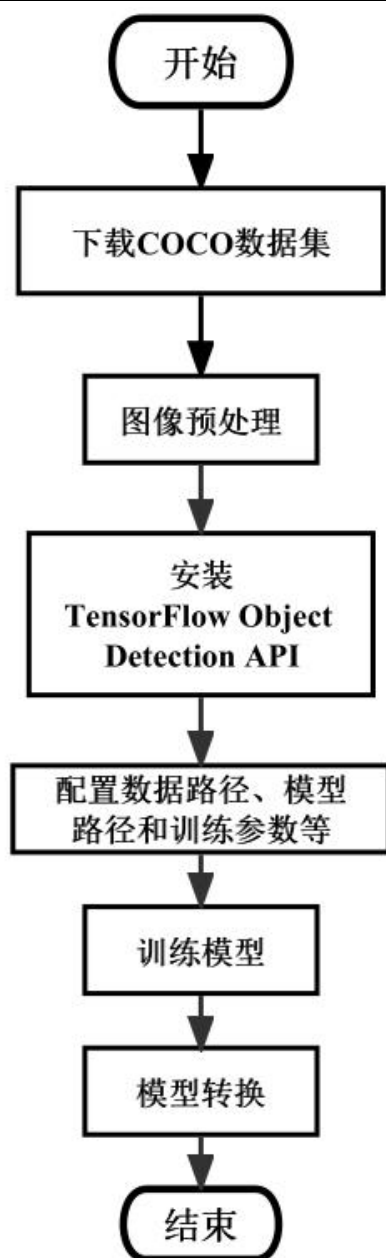


图 3-1

第四章 基于神经网络的垃圾分类方法的硬件实现

4.1 硬件介绍

4.1.1 树莓派 4B

树莓派是一款小型的单板计算机，由英国的 Raspberry Pi 基金会开发。它的设计旨在为教育和创造者提供一个低成本、高性能的平台，用于学习计算机科学、物联网、嵌入式系统等领域。树莓派的外观类似一块小型电路板，尺寸较小且易于携带。它包含一个主处理器、内存、输入输出接口等核心硬件组件。树莓派的硬件版本有多个，每个版本在处理器性能、内存容量、接口等方面可能会有所不同。树莓派采用 ARM 架构的处理器，具有较强的计算能力。根据不同的型号，处理器的性能和速度也会有所不同。较新的型号通常配备了更快的处理器和更大的内存容量，使其能够处理更复杂的任务。

树莓派具有丰富的接口和扩展选项，包括 HDMI 接口用于连接显示器、USB 接口用于连接外部设备、GPIO（通用输入输出）接口用于连接传感器和其他电子元件。这些接口和扩展选项使树莓派成为一个强大的嵌入式计算平台，可用于构建各种创意和项目。

树莓派的设计初衷是为了教育和学习用途。它提供了丰富的学习资源和教育项目，如编程教育、物联网项目、电子制作等。树莓派简单易用，适合初学者入门，同时也提供了高级功能和扩展性，能够满足专业开发者的需求。

树莓派 4B 是树莓派系列中的一款单板计算机，是目前最新和最强大的树莓派型号。树莓派 4B 采用了 Broadcom BCM2711 四核 64 位 ARM Cortex-A72 处理器，主频 1.5GHz，相较于前一代的树莓派 3B+，处理器性能大幅提升，能够更快地处理复杂的计算任务。树莓派 4B 有多个内存配置可选，包括 1GB、2GB 和 4GB LPDDR4 SDRAM。内存容量的增加使得树莓派 4B 能够更好地支持多任务处理和运行内存要求较高的应用程序。

树莓派 4B 通过 MicroSD 卡扩展存储空间，用户可以根据自己的需求选择合适的存储容量。此外，树莓派 4B 还提供两个 USB 3.0 接口和两个 USB 2.0 接口，可以连接外部存储设备。同时它还支持双显示器输出，具备两个微型 HDMI 接口，

可以同时连接两个高清显示器。它支持 4K 分辨率的视频播放和显示,提供更清晰、更细腻的图像效果。树莓派 4B 还具备千兆以太网接口,支持有线网络连接,能够提供更稳定和快速的网络传输速度。此外,树莓派 4B 还提供了无线局域网(Wi-Fi)和蓝牙 5.0 功能,可以无线连接到网络和外部设备。同时还提供了 40 个 GPIO(通用输入输出)接口,用于连接各种传感器、执行器和其他电子元件。这些接口可以扩展树莓派的功能,支持各种创意和项目的开发。树莓派 4B 兼容多种操作系统,包括 Raspberry Pi 官方提供的 Raspbian 系统、Ubuntu、Windows 10 IoT Core 等。用户可以根据自己的需求选择合适的操作系统,并在其上运行各种应用程序和开发工具。

树莓派 4B 适用于各种应用场景,它作为一款强大的嵌入式计算平台,可以用于开发各种嵌入式系统和设备。它可以与传感器、执行器、摄像头等外部硬件组件结合使用,用于构建物联网设备、智能家居系统、机器人等;同时它还非常适合用于电子制作和创客项目。它的丰富接口和扩展性使得用户可以将其与各种电子元件和传感器组合,实现自己的创意和想法。例如,可以构建智能小车、智能气象站、智能音乐播放器等。但需要注意的是,树莓派 4B 作为一款单板计算机,其处理能力和存储容量相对有限。对于一些特别要求高的计算任务或大规模的应用,可能需要考虑更为高性能的计算平台。

4.1.2 树莓派摄像头

树莓派摄像头是专为树莓派设计的一种摄像模块,它可以通过树莓派的 CSI 接口连接到树莓派上,提供高质量的图像和视频捕捉功能。以下是对树莓派摄像头的详细介绍:树莓派摄像头有两种主要类型可供选择:CSI 摄像头和 USB 摄像头。CSI 摄像头是最常用的类型,通过 CSI 接口直接连接到树莓派的摄像头端口。CSI 摄像头具有更高的图像质量和性能,适用于高要求的摄像应用。USB 摄像头则通过 USB 接口连接到树莓派,更为常见且易于使用,适用于一般的摄像需求。

树莓派摄像头提供高质量的图像和视频捕捉能力。CSI 摄像头通常具有更高的分辨率和更好的图像传感器,能够捕捉更清晰、更细腻的图像。图像质量的好坏还受摄像头模块的品牌和型号影响,不同型号的树莓派摄像头有不同的性能和特点。树莓派摄像头可以通过相关的软件设置进行配置和控制。用户可以调整图像的亮度、对比度、饱和度等参数,以及设置曝光时间、白平衡等功能,以满足不

同场景下的摄影需求。摄像头还可以进行实时预览、拍照和录制视频等操作。同时树莓派摄像头还广泛应用于各种项目和应用领域，可以用于构建简单的视频监控系統，实时监视室内或室外环境；通过树莓派摄像头，可以进行机器视觉应用的开发，如人脸识别、物体检测、图像处理等。

4.2 树莓派详细配置过程

4.2.1 树莓派系统烧录

首先需要在树莓派上将系统镜像文件写入 MicroSD 卡,可以使用树莓派官方网站提供的镜像文件自行安装，也可以使用树莓派的烧录工具安装，在烧录系统前，要先使用适当的工具，对 SD 卡进行格式化，以确保正确的文件系统格式。

之后在计算机上安装一个可用的烧录工具，例如 Raspberry Pi Imager，这个树莓派官方工具提供了直观的界面，可以帮助用户选择镜像文件并将其写入 SD 卡。然后选择想要烧录的树莓派系统版本以及安装路径。这里需要注意的是，在 2020 年 7 月之后的树莓派版本中，不再能直接默认以 pi 为账号，raspberrry 为密码，必须要设置自己的账号密码，否则树莓派不能正常登录。烧录完成后，即可将 SD 卡从计算机拔出并将其插入树莓派的上的 SD 卡槽中。

4.2.2 树莓派与电脑有线连接

为了能够尽量节约使用成本，同时提高树莓派与电脑之间文件传输的速度，我考虑使用计算机以有线连接的方式控制树莓派，首先需要开启 SSH，即在 SD 卡根目录中新建 SSH 文件，之后用网线将树莓派与笔记本连接，将树莓派通电，在电脑上安装 Putty 软件，之后在笔记本上打开命令提示符，输入 arp-a 并按回车键查找树莓派的 ip 地址，如下图所示，树莓派的地址通常以 192.168.137 开头，所以只需要尝试下图中的以 192.168.137 开头的地址,输入命令 ping (每一个具体的 ip)，就可以查看到哪个 ip 地址有响应，有响应的地址即为树莓派的 ip 地址。

```

C:\Users\Administrator>arp -a

接口: 192.168.137.1 --- 0x7
Internet 地址      物理地址      类型
192.168.137.84      dc-a6-32-d3-77-2c 静态
192.168.137.85      dc-a6-32-d3-77-2c 静态
192.168.137.255     ff-ff-ff-ff-ff-ff 静态
224.0.0.22          01-00-5e-00-00-16 静态
224.0.0.251         01-00-5e-00-00-fb 静态
224.0.0.252         01-00-5e-00-00-fc 静态
239.255.255.250     01-00-5e-7f-ff-fa 静态
255.255.255.255     ff-ff-ff-ff-ff-ff 静态

接口: 172.25.9.99 --- 0xd
Internet 地址      物理地址      类型
172.25.9.165        18-26-49-af-6d-c4 动态
172.25.11.165        90-78-41-e5-94-58 动态
172.25.12.153        5c-e4-2a-01-35-f6 动态
172.25.13.228        e8-b1-fc-cc-01-eb 动态
172.25.15.250        1c-7d-22-28-79-d8 动态
172.25.15.254        58-66-ba-81-93-a4 动态
172.25.15.255        ff-ff-ff-ff-ff-ff 静态
224.0.0.22          01-00-5e-00-00-16 静态
224.0.0.251         01-00-5e-00-00-fb 静态
224.0.0.252         01-00-5e-00-00-fc 静态
239.255.255.250     01-00-5e-7f-ff-fa 静态
255.255.255.255     ff-ff-ff-ff-ff-ff 静态

C:\Users\Administrator>

```

图 4-1

```

C:\Users\Administrator>ping 192.168.137.84

正在 Ping 192.168.137.84 具有 32 字节的数据:
来自 192.168.137.84 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.137.84 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.137.84 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.137.84 的回复: 字节=32 时间=1ms TTL=64

192.168.137.84 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 1ms, 最长 = 1ms, 平均 = 1ms

C:\Users\Administrator>ping 192.168.137.85

正在 Ping 192.168.137.85 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.137.85 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

```

图 4-2

在获取了树莓派的 ip 之后,即可通过 Putty 软件登录树莓派,如图所示在软件

中输入 ip 地址，点击 open，再在之后的页面中输入树莓派的账号密码，即可成功登录树莓派。

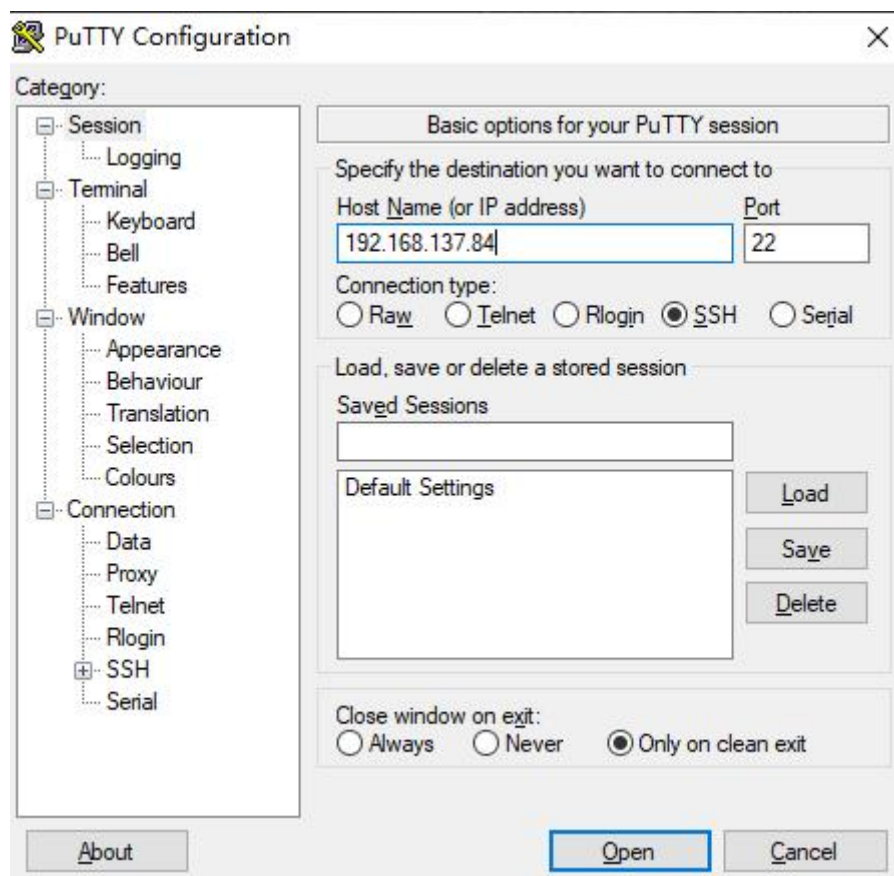


图 4-3

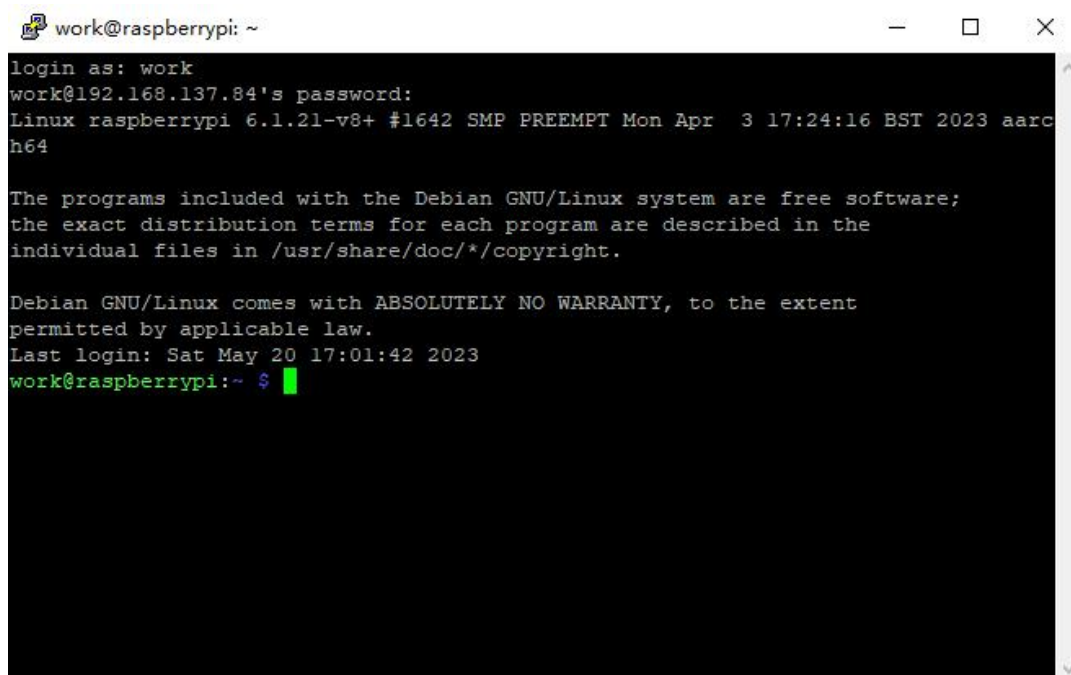


图 4-4

通过 Putty 软件 SSH 远程连接树莓派之后，可以使用 VNC 软件在笔记本上配置树莓派的虚拟桌面，从而更方便计算机对树莓派进行操作，首先现在计算机上下载 VNC Viewer 软件，并在软件中树莓派的 ip 地址准备连接，如图所示。

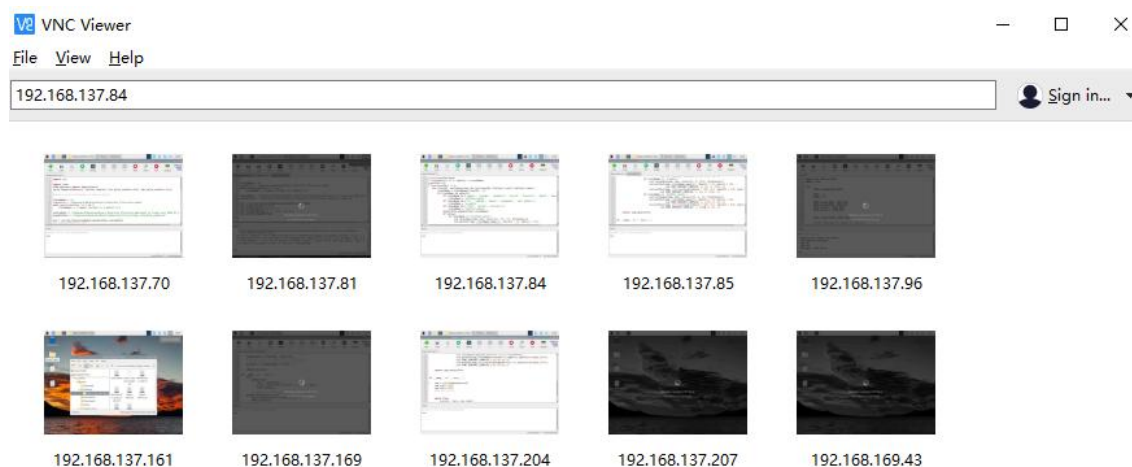


图 4-5

之后在弹出的窗口中输入账号密码，点击 ok，如图所示。这样就连接成功了，连接之后显示的窗口如图所示

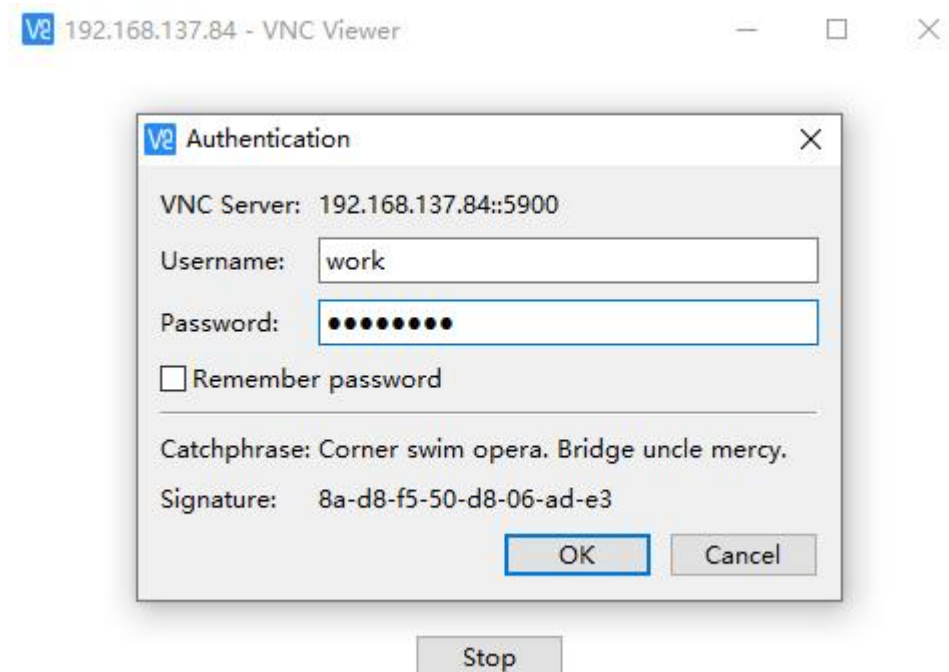


图 4-6

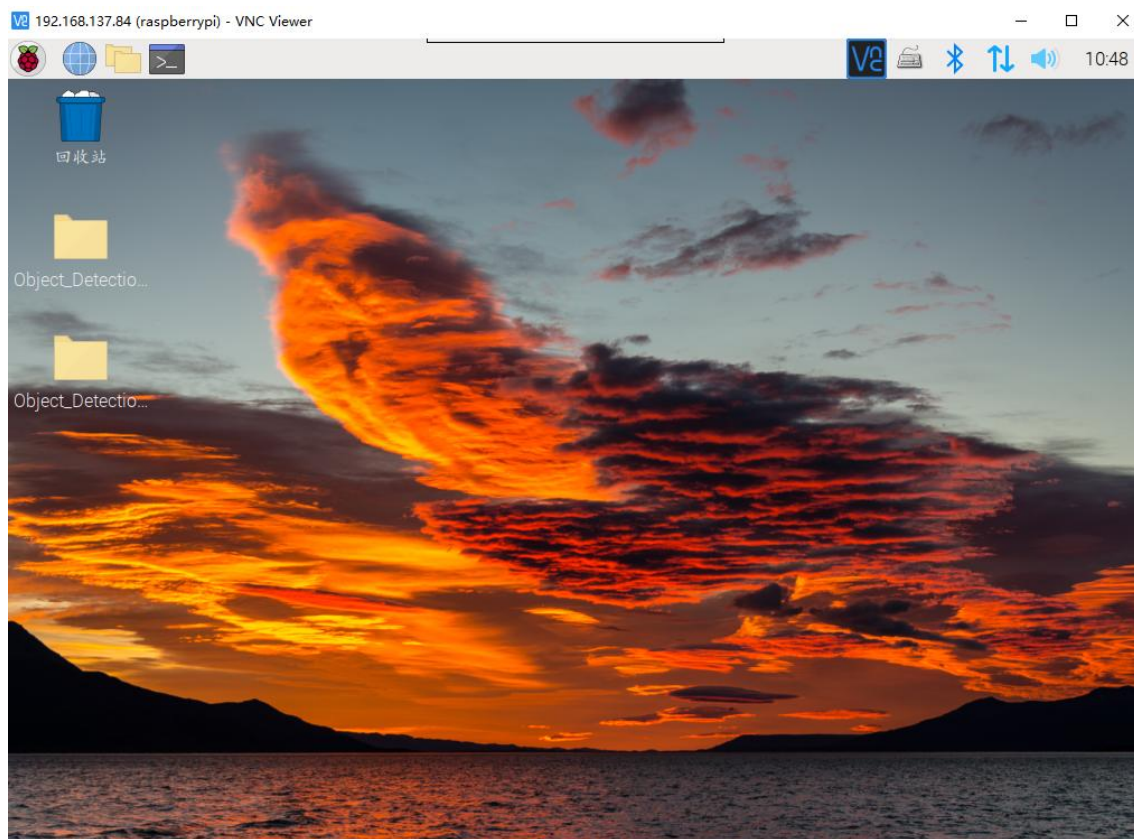


图 4-7

4.2.3 树莓派与电脑无线连接

有时为了提供树莓派的便携性，考虑将树莓派和电脑进行无线连接，首先与有线连接一样，要先配置 SSH 远程登录文件，即在 SD 卡中新建 SSH 文件，与有线连接不同的是，无线连接还需要新建一个 `wpa_supplicant.conf` 文件，在该文件中需要输入如下内容：

```
country=CN

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev

update_config=1

# 配置多个 WiFi

network={

ssid="WiFi 链接名称"

psk="WiFi 密码"
```



```
key_mgmt=WPA-PSK
priority=5
}

network={
ssid="WiFi 链接名称"
psk="WiFi 密码"
key_mgmt=WPA-PSK
priority=3
}
```

其中 WiFi 链接名称和 WiFi 密码需要填写为使用的 WiFi 名称和密码，若只使用一个 WiFi，第二段的有关第二个 WiFi 的配置可以省略。笔记本电脑要与树莓派连接同一个 WiFi。下一步需要获取树莓派的 IP 地址，我采用的方法是将树莓派连接上手机开启的热点，之后打开手机上的终端模拟器，在终端模拟器中输入 `ip neigh`，之后会跳出两个 IP 地址，排在前面的那一个即为树莓派的 IP 地址，如图所示。

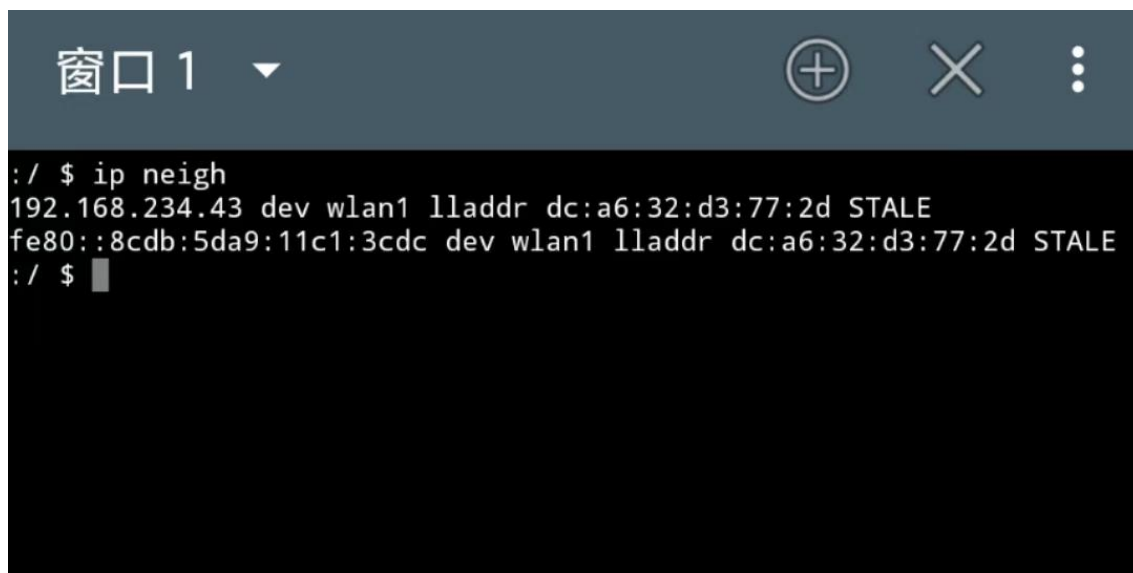


图 4-8

之后即可通过 Putty 与 VNC Viewer 建立电脑与树莓派的连接，此过程与有线连接一致，在此不再赘述。

4.2.4 树莓派 OpenCV 安装

首先需要更新一下树莓派，在 Putty 窗口中输入

```
sudo apt-get update&&sudo apt-get upgrade
```

之后安装所需要的各类包，在 Putty 窗口中输入

```
sudo apt-get install build-essential cmake pkg-config
```

```
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
```

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
```

```
sudo apt-get install libxvidcore-dev libx264-dev
```

```
sudo apt-get install libgtk2.0-dev libgtk-3-dev
```

```
sudo apt-get install libatlas-base-dev gfortran
```

```
sudo pip3 install numpy
```

接下来获取 OpenCV 安装包并解压安装包，在窗口中输入

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.4.0.zip
```

```
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.4.0.zip
```

```
unzip opencv.zip
```

```
unzip opencv_contrib.zip
```

接下来安装 OpenCV，在窗口中输入

```
cd ~/opencv-4.4.0/
```

```
mkdir build
```

```
cd build
```

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
```

```
    -D CMAKE_INSTALL_PREFIX=/usr/local \
```

```
    -D INSTALL_PYTHON_EXAMPLES=ON \
```

```
    -D
```

```
OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-4.4.0/modules \
```

```
    -D BUILD_EXAMPLES=ON ..
```

```
make -j $(nproc)
```

等待一小时左右，OpenCV 就安装好了。

4.2.5 手机端控制树莓派

为了方便树莓派的使用，我还尝试了使用手机连接树莓派，这个过程中在手机上使用了 RaspController 软件和终端模拟器。

首先先配置树莓派的 WiFi，在 wpa_supplicant.conf 文件中输入手机开的热点的账号和密码，这样只要启动了树莓派，树莓派就会自动连接上手机所开的热点。之后打开终端模拟器，输入命令 ip neigh，就能获取树莓派的 IP 地址，此过程与树莓派与电脑的无线连接类似，之后打开软件 RaspController 软件，点击添加设备，输入设备名称，IP 地址，账号与密码，如图所示，这样就将树莓派与手机连接成功了。

图 4-9

通过手机与树莓派的连接，可以利用手机查看树莓派的很多信息，例如可以

查找树莓派上的文件，打开树莓派的相机以查看周围环境，查看树莓派 CPU 的使用状况以及系统的温度，如图所示。



图 4-10

第五章 实验检测

5.1 模拟检测实验

为了确保该模型能够在真实环境使用，我针对可识别的三类物品都进行了检测实验，在厨余垃圾里，我识别了橘子、香蕉、胡萝卜、披萨，在电子垃圾里，我识别了鼠标、电脑、手机、平板，在危险垃圾中，我识别了剪刀和小刀，详细识别结果见下图，由于部分物品在附近没有实物，我采用了在平板上打开相关物品照片的方法，但是这样识别系统同时也会检测到平板并输出电子垃圾的结果。

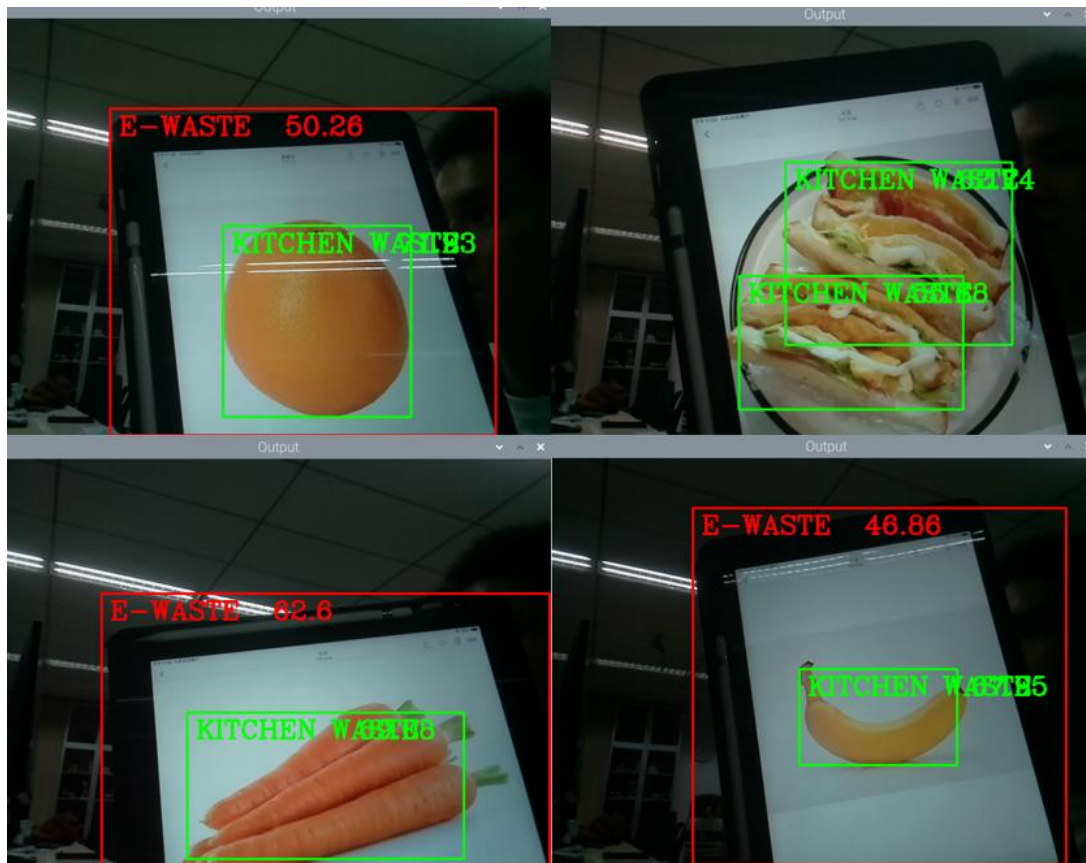


图 5-1

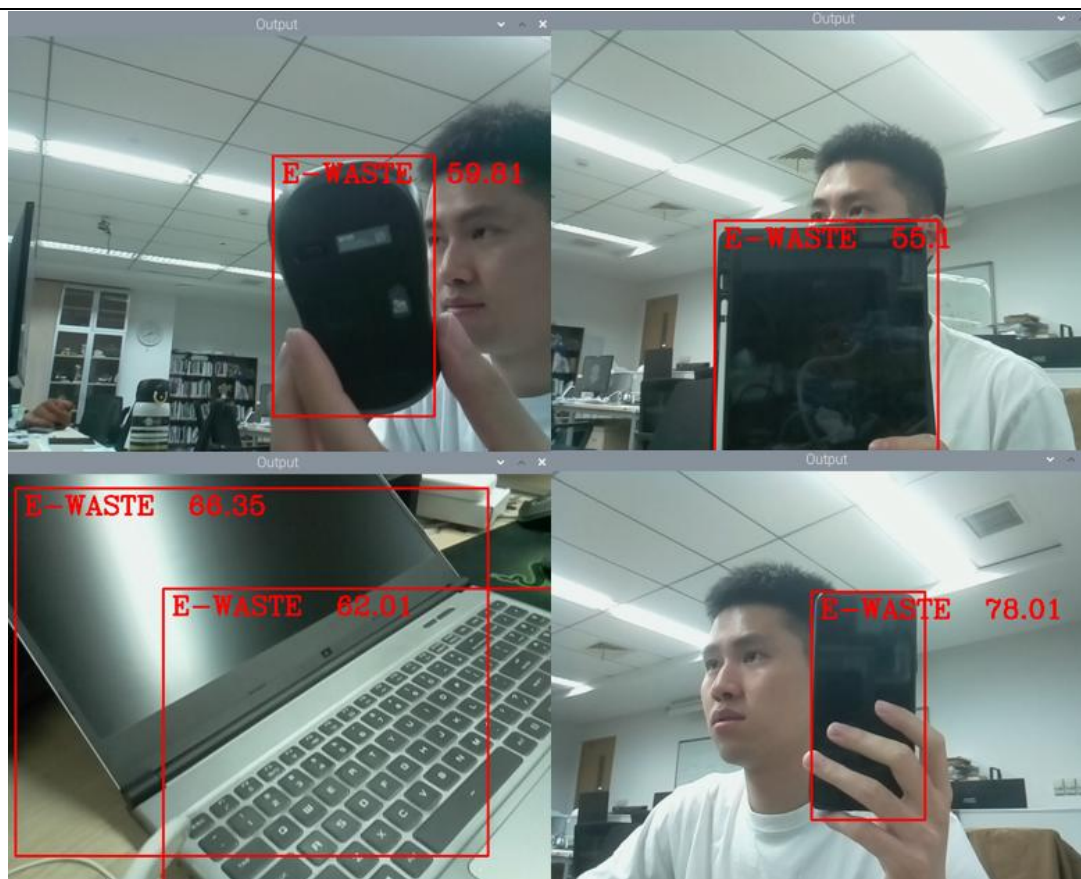


图 5-2

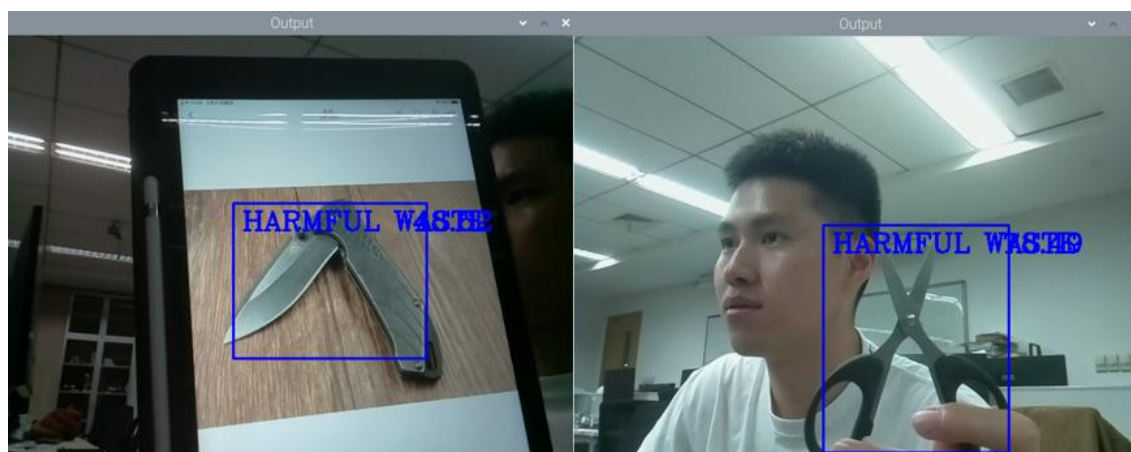


图 5-3

第六章 总结与展望

6.1 总结

垃圾分类现在已经是我国城市的主流趋势，为了提高垃圾分拣的效率，我设计了这款辅助垃圾分拣系统，可以识别出厨余垃圾、电子垃圾和危险垃圾，帮助

使用者进行一个垃圾预分类，这样可以大大提高分拣效率，同时能尽量保护使用者不受到例如小刀剪刀等易划伤物品伤害。同时我使用了树莓派作为处理器，其体积相对较小，价格相对较便宜，同时方便在现有垃圾桶上进行改装，具有不错的市场前景。

6.2 展望

当前对此垃圾分类系统的研究还仅仅停留在识别上，以后我考虑加装舵机以实现垃圾分类的全过程，同时当前代码运行的延时较大，识别还存在一定的不准确几率，后期考虑继续优化算法以实现低延时精准识别！

参考文献

- [1] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408.
- [2] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. *Nature*, 1986, 323(6088): 533-536.
- [3] Hinton, G. E.; Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 2006, 313(5786): 504-507.
- [4] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. *Nature*, 1986, 323(6088): 533-536.
- [5] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- [6] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- [7] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504-507.
- [8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.
- [9] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 580-587.
- [10] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788.
- [11] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *European Conference on Computer Vision (ECCV)*, 21-37.
- [12] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.

- [13]He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.
- [14]Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- [15]Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.0486

致谢

不知不觉，我的大学四年生活走到了尾声，这篇毕业论文也算是为大学生活画了一个句号，在这个过程中我有非常多的人想感谢。

首先我想感谢董劲男老师，刘思远老师和张云辉老师，董老师是我毕业论文的负责老师，他不厌其烦地解答了我许多关于毕业论文的问题，同时还对我论文的修改提供了宝贵的意见。刘思远老师是我本专业的班主任，同时指导我参加了一些比赛，取得了一些成绩，对我的专业认知和人生规划都有很大的影响。张云辉老师是我的本科生导师，她是一位非常认真负责的教师，同时对学生特别关心，对待专业严谨的态度也对我有着很大的启发。

然后我想感谢我的父母，他们永远非常支持我的决定，永远给我他们能给予我的最好的，永远对我充满信心。同时我想感谢陪伴在我身边的同学们，很多同学都非常乐于助人，还有一些同学给我树立了很好的榜样，让我有了前进的方向。

最后我想感谢默默坚持的自己，虽然目前在很多方面都没有做到比较出色，但我还会继续努力的。未来或许会有坎坷，但向前走便是了。