



本科毕业设计（论文）

基于深度学习的智能分类垃圾桶

Intelligent classification trash can based on deep learning

院（系）	计算机科学与技术
专 业	电子信息工程
班 级	4 班
学 号	16210420405
学生姓名	刘远旭
指导教师	杨斌
提交日期	年 月 日

毕业设计（论文）原创性声明

本人郑重声明：所呈交的毕业设计（论文），是本人在指导老师的指导下，独立进行的设计（研究）工作及取得的成果，论文中引用他人的文献、数据、图件、资料均已明确标注出，论文中的结论和结果为本人独立完成，不包含他人已经发表或撰写的作品及成果。对本文的研究作出贡献的个人和集体，均已在论文中作了明确的说明。本人完全意识到本声明的法律结果由本人承担。

毕业论文作者（签字）：

签字日期：

年

月

日

中文摘要

由于人们生活水平的提高和环保意识的增强，生活垃圾分类有效性成了人们关注的焦点。调查研究发现，在源头处对垃圾进行分类处理的方法是整个垃圾分类处理流程中最高效的也是分类最彻底的。

在本项目中，主要是设计一个使用深度学习技术来对垃圾进行图像识别分类，然后再将分类后的垃圾放置到指定的垃圾桶中的装置。

选择正确的图像识别分类算法是本项目的重中之重。因此，本文对过往的图像识别方面的突出贡献的算法做介绍和总结，把握其发展的脉络，从而说明选择 MobileNetV2 网络的原因。

经过实际测试，垃圾的类别判断正确率能在可接受范围内，而识别分类速度则在经过分类结果滤波后，平均成功识别一次垃圾的平均时间为 4 秒，这证明该项目在智能垃圾分类领域还是很有发展前景的。

关键词：垃圾分类,卷积神经网络,MobileNetV

Abstract

Due to the improvement of people's living standards and the enhancement of environmental awareness, the effectiveness of domestic waste classification has become the focus of attention. The investigation and study found that the method of sorting waste at the source is the most efficient and the most thorough in the entire waste sorting process.

In this project, it is mainly to design a device that uses deep learning technology to classify the garbage for image recognition, and then put the classified garbage into the designated garbage bin.

Choosing the correct image recognition classification algorithm is the top priority of this project. Therefore, this article introduces and summarizes the algorithms that have made outstanding contributions in the past in image recognition, grasping the development context, and thus explaining the reasons for choosing the MobileNetV2 network.

After actual testing, the classification accuracy rate of the garbage can be within the acceptable range, and the recognition and classification speed is filtered after the classification result, and the average time for successfully identifying a garbage is 4 seconds, which proves that the project is in the field of intelligent garbage classification. It is still very promising.

Key words: Garbage Classification; Convolutional Neural Network; MoblieNetV2

目录

中文摘要.....	I
Abstract.....	II
目录.....	III
图目录.....	VI
表目录.....	VIII
第一章 绪论	1
1.1 课题研究背景与意义.....	1
1.1.1 研究背景.....	1
1.1.2 意义.....	2
1.2 国内外垃圾分类及研究现状.....	3
1.2.1 国外现状.....	3
1.2.2 国内现状.....	3
1.3 本文研究内容	3
第二章 智能分类垃圾桶总体设计	5
2.1 概述	5
2.2 系统分析及设计思路.....	5
2.3 系统框架	5
2.4 系统总体运行流程	6
2.4.1 主处理器及主控制器处理流程	6
2.5 本章小结	7
第三章 智能垃圾分类图像处理原理.....	8
3.1 卷积神经网络对图像的基本操作.....	8
3.1.1 卷积操作.....	8
3.1.2 池化操作.....	9
3.2 LeNet 网络	10

3.2.1 综述.....	10
3.2.2 特点.....	12
3.3 AlexNet 网络.....	13
3.3.1 概述.....	13
3.3.2 特点.....	13
3.4 VGGNet 网络.....	14
3.4.1 特点.....	15
3.5 GoogleNet 网络及 Inception 架构	16
3.5.1 背景介绍.....	16
3.5.2 InceptionV1.....	16
3.5.3 InceptionV2.....	18
3.5.4 InceptionV3.....	20
3.6 ResNet 网络	21
3.6.1 综述.....	21
3.6.2 残差结构.....	22
3.7 MobileNet.....	23
3.7.1 背景介绍.....	23
3.7.2 MobileNetV1	24
3.7.3 MobileNetV2	24
3.8 本章小结	27
第四章 智能垃圾桶控制系统硬件设计	28
4.1 智能垃圾桶整体结构分布	28
4.2 STM32 硬件系统框架.....	28
4.3 STM32 微处理器.....	29
4.4 步进电机与步进电机驱动器.....	29
4.4.1 步进电机.....	29
4.4.2 步进电机驱动器.....	29

4.5 Jetson Nano	30
4.6 本章小结	31
第五章 系统实现及测试	32
5.1 识别算法选取与训练	32
5.1.1 MobileNetV2	32
5.1.2 TensorFlow 框架	32
5.1.3 训练 MobileNetV2 网络	33
5.2 串口通信测试	35
5.2.1 通信格式	35
5.2.2 通信测试	36
5.3 实际垃圾分类测试	37
5.4 本章小结	38
第六章 总结与展望	1
参考文献	2
致谢	4

图目录

图 1.1 2014-2019 年中国大、中城市生活垃圾产生量	1
图 2.1 系统总体框架.....	5
图 2.2 左 A 为主控制器处理流程 右 B 为主处理器处理流程	6
图 2.3 系统总体运行流程.....	6
图 3.1 卷积在图象中的计算图示.....	9
图 3.2 左为平均池化 右为最大池化.....	10
图 3.3 LeNet-5 神经网络模型	10
图 3.4 全连接层（从左到右分别为输入层、隐藏层、输出层）	11
图 3.5 激活函数.....	11
图 3.6 Sigmoid 函数(S 型函数).....	12
图 3.7 AlexNet 网络	13
图 3.8 ReLU 函数图像.....	13
图 3.9 Dropout 方法	14
图 3.10 左图为早期版本 右图为改进版本.....	16
图 3.11 通过不同的卷积核以及池化层实现的聚类.....	17
图 3.12 5×5 卷积被替换方式	17
图 3.13 完善后的 InceptionV1 结构.....	18
图 3.14 两个连续 3×3 卷积核替代 5×5 卷积核	19
图 3.15 InceptionV2 结构	19
图 3.16 BN 方法计算公式	20
图 3.17 BN 方法计算示例	20
图 3.18 使用 3×1 和 1×3 的非对称卷积替代 3×3 卷积.....	21
图 3.19 InceptionV3 结构.....	21
图 3.20 常规深度网络过深会导致深层网络训练效果反而不如浅层网络.....	22
图 3.21 残差块(Residual Block).....	22
图 3.22 Shortcut Connections 示例	22
图 3.23 图左为 Option A 图右为 Option B	23
图 3.24 深度卷积与逐点卷积.....	24
图 3.25 两种倒残差结构.....	25
图 3.26 ReLU6 函数图像.....	25
图 3.27 不同维度下使用 ReLU 激活函数后再还原的图像.....	25
图 3.28 MobileNetV2 结构	26

图 4.1 垃圾桶整体结构.....	28
图 4.2 STM32 主控模块	28
图 4.3 电机驱动器连接图.....	30
图 4.4 Jetson Nano B01 底板	30
图 4.5 神经计算性能对比.....	31
图 5.1 训练、验证、测试集.....	33
图 5.2 用来训练网络的数据集.....	33
图 5.3 训练集准确率.....	33
图 5.4 验证集准确率.....	34
图 5.5 训练集损失函数.....	34
图 5.6 验证集损失函数.....	34
图 5.7 实际测试材料.....	37
图 5.8 实际测试材料的识别效果概率分布.....	37

表目录

表 1.1 2014-2019 年信息发布城市数量	1
表 3.1 VGG 网络层数配置	15
表 3.2 VGGNet 各个深度下的单张图片的准确率	15
表 3.3 ResNet 网络层数配置	23
表 3.4 MobileNetV2 网络层数配置	26
表 3.5 MobileNetV2 网络在 ImageNet 数据集上性能和参数对比	27
表 3.6 MobileNetV2 网络在 COCO 数据集上性能和参数对比	27
表 4.1 步进电机参数表	29
表 4.2 树莓派 4 与 Jetson Nano 参数对比表	31
表 5.1 各个神经计算模型性能对比表	32
表 5.2 测试集图片识别成功率	35
表 5.3 串口通信格式表	35
表 5.4 主控制器串口测试表	36
表 5.5 主处理器串口测试表	36

第一章 绪论

1.1 课题研究背景与意义

1.1.1 研究背景

人类科技进步和城市化极大的改善了人们的生活质量，生活垃圾产量也随之水涨船高，“垃圾围城”这一景象也在各大城市中不断上演。

据《2019 年全国大、中城市固体废物污染环境防治年报》^[1]显示，2019 年，全国 202 个大、中城市的生活垃圾产生量为 21147.3 万吨。



图 1.1 2014-2019 年中国大、中城市生活垃圾产生量

表 1.1 2014-2019 年信息发布城市数量

发布年份	强制发布城市		自愿发布城市	总数
	重点城市	模范城市		
2014	47	54	162	263
2015	47	56	141	244
2016	47	56	143	246
2017	47	57	110	214
2018	47	57	98	202
2019	47	55	98	200

从上面数据可以看到，生活垃圾以每年一千万吨的量在增长，这是一个惊人数据。而在国外，梅波克洛夫指出：，美利坚合众国一年生成的城市固体废弃物高达 2.39 亿吨。这些垃圾，大多都是混合回收，在垃圾回收中心进行简单回收后，再填埋到指定地区。这种粗暴的垃圾处理方式，不仅占用珍贵的土地资源去填埋垃圾，又浪费了一些可回收的资源。

在 2019 年极具讨论性的上海垃圾分类话题中，上海市实行了史上最严垃圾分类管理规定。根据官方发布数据，截止至当年 8 月底，上海实现每天 4500 吨的可回收物回收，比去年年底多了 5 倍^[19]之多。由此可见，在源头对垃圾进行分类是垃圾分类与回收中极重要的一环。但是在上海实行的垃圾分类规定的管理中，可以看到很多都是通过人工去分类、监管的，这种方式给人们的日常生活还是造成一定的困扰，例如只能定时扔垃圾、垃圾分类过于复杂等问题。而且，人们虽然认识到垃圾分类的作用，也觉得应该进行，但是对垃圾分类方法却不是太了解^[3]，因此需要一种辅助手段去帮助他们。

另一方面，随着计算机技术的快速变革，深度学习作为机器学习的重要分支日益得到普及。如今世界处处都存在它的踪迹，如股票行情预测、人脸识别及智能机器人等。可以说，它在智能制造行业里面有着非常光明的前景，而其中的图像分类与检测领域更是出类拔萃。

1.1.2 意义

正如日本水泥大王浅野总一郎所说的“在这个世界上没有一件无用的东西,任何东西都是可以利用的”。如一般做填埋处理的湿垃圾可以通过厌氧发酵处理工艺^[2]等技术实现生物发电，提高了对资源的利用率。垃圾越早分拣,成分越简单,也就越容易分拣^[20]。然而传统的人工分拣方法，既对人健康不利，效率也不够高。

因此，本文所设计的基于深度学习的智能分类垃圾桶，通过深度学习算法去分类垃圾类别，并且能根据不同地区的要求进行不同种类划分，从而在源头处自动对垃圾进行分类和处理。

其优点在于：通过技术手段，既客观上减轻了人们在垃圾分类上面的负担，也让主观上缺乏分类垃圾意识的民众一起参与进来，这也符合当前的利用人工智能建设智能城市的市场发展趋势。既能促进环保发展，又能减轻人们的日常生活负担，使未来更美好。

1.2 国内外垃圾分类及研究现状

1.2.1 国外现状

.(1) 日本

日本对生活垃圾分类的处置采取精细化的态度，所以垃圾分类领域中名列各国第一。其分类最主要就是在源头处进行精细分类，共有 15 大类之多，并且对每一种垃圾都附有详细的处理要求。如废旧报纸要求要捆绑得整齐，而废电器的电线则要求绑在它自身上^[4]。但是这种方式，是日本从 90 年代就开始动员形成的，而且伴随着高额的罚款。它的发展不是一蹴而就，而是经历了多个阶段才最终形成现在的机制。

.(2) 美国

美国既是一个经济强国，也是一个垃圾产生大国。它本身有一套垃圾分类标准，而且还将垃圾处理形成了产业。在 2018 年，美国的清洁科技领域获得超过 40 亿美元的投资，其投资率相比 17 年增长了 54%。美国垃圾分类机器人初创公司 Clean Robotics 更是在 15 年便将一部分智能分类垃圾桶投入到了市场。但是其本身高昂的价格与分类效率低下使得其没有铺开来。而也有些公司采用终端处理方式，使用智能分类算法、机械臂和流水线结合的方式形式分类。

1.2.2 国内现状

国内在垃圾分类处理方面虽然立法和建立试点城市的方法实施已久，但是由于宣传和成本等问题，在国内没有形成普遍的垃圾分类意识。同时，政府方面在政策上也没有足够支持，很多地方的分类垃圾桶，在进行垃圾回收时，采用的是混合回收，哪怕是在源头进行分好，最终依然是混在一起。当然，这也与民众平时也不按垃圾分类标准进行投放有关。2019 年的上海最严垃圾分类规定的执行，一方面看到了垃圾分类的必要性，另一方面也对民众生活造成了一定困扰，甚至一度在微博等社交网络平台形成了讨论热潮。

在目前的做法中，国内依然采用分类垃圾桶的方式，通过人工分类，手工投放的方式进行投放。当然，为了辅助人工分类，如阿里巴巴集团等公司也上线了手机 app 分类小程序对垃圾进行分类，还有是基于 NB-IOT 技术在本地拍照上传到远程服务器上识别，再返回分类结果后对垃圾进行分类投放到对应垃圾桶的方式^[5]。

1.3 本文研究内容

本文主要是基于深度学习的智能分类垃圾桶研究与设计，通过基于深度学习的移动终端的轻量级卷积神经网络，针对移动终端或嵌入式设备，使用深度学习的图

像分类技术对垃圾桶内的垃圾进行检测分类，然后通过电机结构将其投放到对应的桶体中。

本文各章节安排如下：

第一章：绪论。讲述垃圾分类研究的背景和意义，接着介绍国内外在垃圾分类处理这方面的主要方法和现状。

第二章：智能分类垃圾桶的总体设计。主要是通过分析需求从而确定系统的整体流程，然后分别表述了系统各个模块和总体的运行流程。

第三章：介绍了卷积神经网络的一些基本术语和操作，同时介绍这一领域算法的逐步发展，由此引出本身使用 **MobileNetV2** 模型的原因。

第四章：智能垃圾桶控制系统硬件设计。主要是描述了垃圾桶的机械结构和硬件电路以及电机等内容。

第五章：系统实现及测试。介绍了模型的训练过程和模块之间的通信编码以及通信测试等。最后对整个智能分类垃圾桶进行总体测试，证明分类的准确性。

第二章 智能分类垃圾桶总体设计

2.1 概述

本项目中所设计的智能分类垃圾桶是基于深度学习实现的智能分类。通过安装在垃圾桶中的 USB 摄像头采集图像传送到作为处理中心的 Jetson Nano 上，然后 Jetson Nano 对图像信息进行分类处理，再将处理后的结果通过串口传输到 STM32 中，由 STM32 根据分类结果将垃圾投放到对应的桶体中。

2.2 系统分析及设计思路

考虑到垃圾桶对垃圾分类并将其投放到相应的垃圾桶中是一个实时性要求很高的任务，而由终端拍摄照片或者传输视频流的方式将采集样本发送到云端，再由云端处理完返回结果给终端再进行处理，这种方式耗时太长。而且在网络波动较大的地方，更是难以使用。

本项目采用的是本地进行处理的方式。由于有 Google 公司提出的 MobileNetV2 算法跟 NVIDIA 公司开发的 Jetson Nano 平台，使得在本地进行高效的图像检测分类成为了可能。这种方法避免了因为网络波动而造成的识别时间过久甚至是无法识别的问题。

本项目中，采用的是模块化设计的思路，分为识别分类模块和控制模块。这样分类主要原因是 Jetson Nano 平台是带操作系统的，虽然本身有 PWM 输出，但是其精准度依旧比不上 MCU。因此在控制部分中采用 STM32F103 作为主控芯片，在接收到 Jetson Nano 的分类信号后，控制步进电机去投放垃圾到对应的垃圾桶。Jetson Nano 和 STM32 之间由于距离近，所以直接采用串口通信方式进行，避免了无线信号会出现的干扰问题。

2.3 系统框架

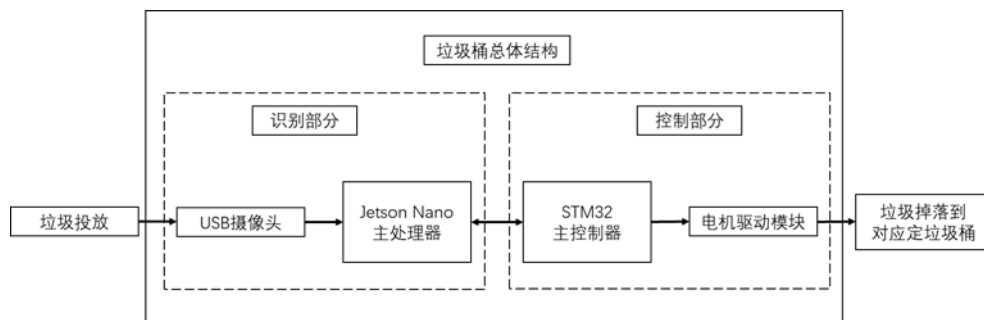
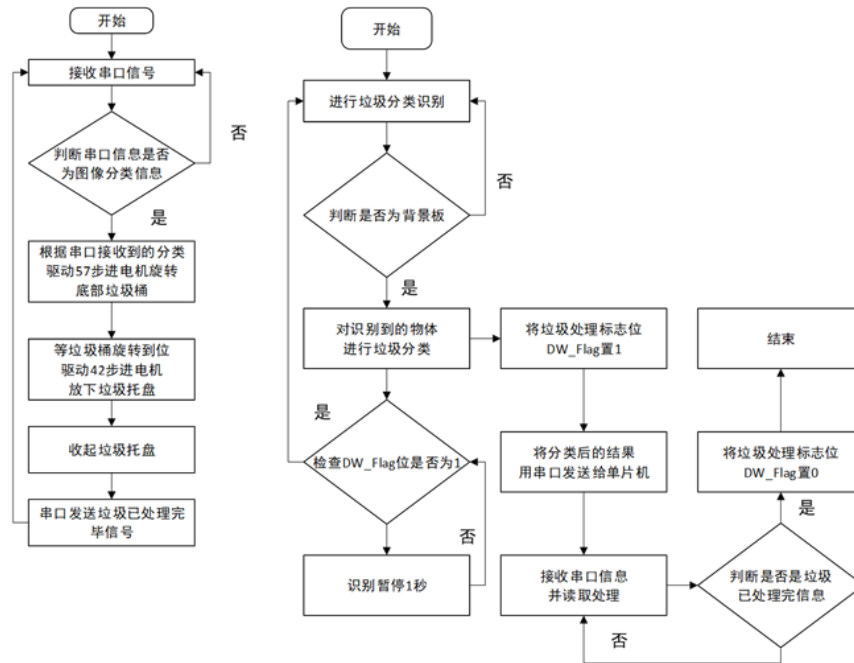


图 2.1 系统总体框架

2.4 系统总体运行流程

2.4.1 主处理器及主控制器处理流程



A: 主控制器处理流程

B: 主处理器处理流程

图 2.2 左 A 为主控制器处理流程 右 B 为主处理器处理流程

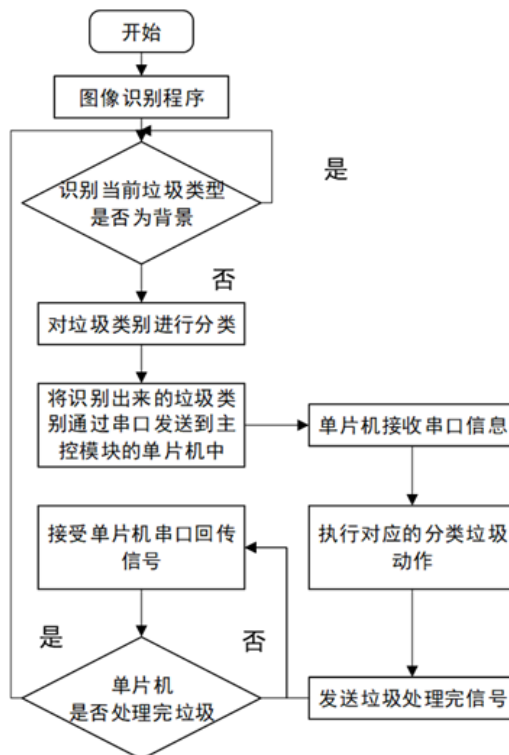


图 2.3 系统总体运行流程

2.5 本章小结

本章主要分析了本项目的项目需求，然后根据项目需求去制定项目的运作流程。通过运作流程将系统分为控制模块和处理模块两大模块，通过分开模块化操作，实现系统的稳定运行。同时也介绍了系统各个模块和总体的运行流程。

第三章 智能垃圾分类图像处理原理

3.1 卷积神经网络对图像的基本操作

3.1.1 卷积操作

其实质是图像矩阵和权值矩阵的点乘

卷积运算公式为

$$y(n) = x(n) * h(n) = \sum_{i=0}^N x_i h(n-i) \quad (3.1)$$

卷积计算的相关术语：

.(1) Filter

过滤器，也称卷积核，它的本质是卷积神经网络中的权值矩阵，通过在“二维平面”中运动，将卷积核包裹住的部分元素实行矩阵乘法，接着将计算得到的结果求和得到单个输出像素值，从而实现对图像进行特征提取。

.(2) Stride

步长，一般小于卷积核的尺寸，主要用来驱动卷积核在图像进行上滑动的长度。如果步长等于 1，则向右滑动一格，步长为 k，滑动 k 格，具体如图 3.1 卷积在图像中的计算图示所示。

.(3) Padding

填充，就是使用额外的自定义像素（通常值为 0，因此也称零填充）填充图像的边缘。在不使用填充的情况下，卷积核很难采集到位于边缘的像素的特征，边缘的特征很容易就被忽略掉。而填充通过在边缘外增加一圈假像素，使得卷积核的可以让边缘区域的像素处于卷积核的中心位置。这样不仅能在卷积操作结束之后提取到边缘像素的特征，也能产生与输入矩阵尺寸相同的输出矩阵。

.(4) Feature Map

特征图，卷积计算之后产生的图像

.(5) 卷积后图像尺寸

h_{in} 为输入矩阵的高度， w_{in} 为输入矩阵的宽度， h_{out} 为输出矩阵的高度， w_{out} 为输出矩阵的宽度， F 为卷积核的尺寸， P 为填充值， S 为步长

$$h_{out} = \frac{h_{in} - F + P}{S} + 1 \quad w_{out} = \frac{w_{in} - F + P}{S} + 1 \quad (3.2)$$

.(6) 图像中的卷积计算步骤

可分为以下三步：

求点积：将 5×5 的输入矩阵中 3×3 深蓝色区域中每个元素分别与其对应的权值矩阵相乘，然后再相加，得到输出矩阵的第一个元素

滑动窗口：若步长为 1，则将 3×3 权值矩阵向右移动一个格

重复操作：重复执行“求点积-滑动窗口”操作，直到输出矩阵被填满

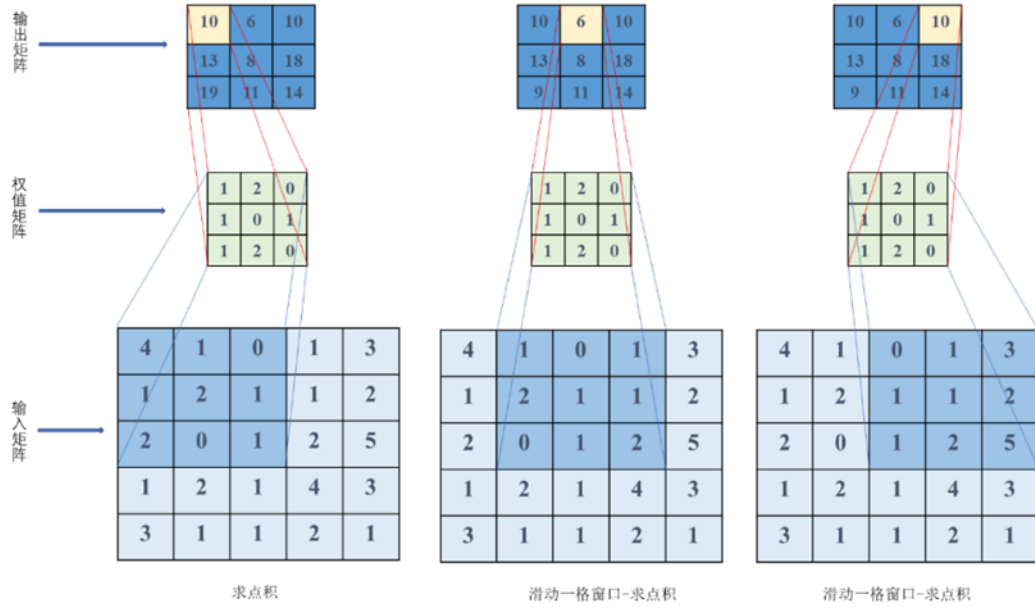


图 3.1 卷积在图象中的计算图示

$$y_{1,1} = 4 \times 1 + 1 \times 2 + 0 \times 0 + 1 \times 1 + 2 \times 0 + 1 \times 1 + 2 \times 1 + 0 \times 2 + 1 \times 1 = 10 \quad (3.3)$$

$$y_{1,2} = 1 \times 1 + 0 \times 2 + 1 \times 0 + 2 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 1 + 1 \times 2 + 2 \times 0 = 6 \quad (3.4)$$

$$y_{1,3} = 0 \times 1 + 1 \times 2 + 3 \times 0 + 1 \times 1 + 1 \times 0 + 2 \times 1 + 1 \times 1 + 2 \times 2 + 5 \times 0 = 10 \quad (3.5)$$

3.1.2 池化操作

在卷积层之后通常紧挨着一个降采样层，通过减少矩阵的高度和宽度，从而减少计算参数，加快网络运行的速度，这一操作就是池化操作。而降采样就是降低特定信号的采样率的过程。

池化包括均值池化和最大池化两种：

.(1) 均值池化

对池化区域内的像素点取均值，这种方法得到的特征数据对背景信息更敏感。

.(2) 最大池化

对池化区域内所有像素点取最大值，这种方法得到的特征数据对纹理特征信息

更敏感，示例如图 3.2 所示

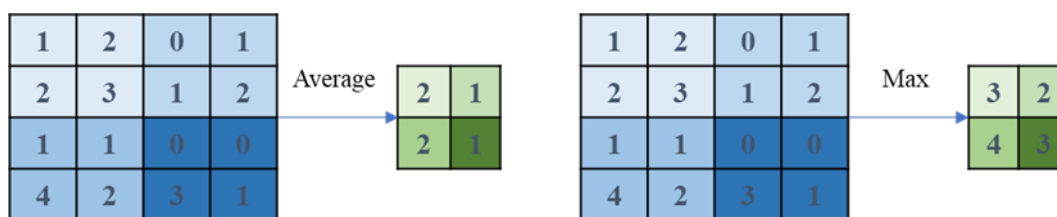


图 3.2 左为平均池化 右为最大池化

3.2 LeNet 网络

3.2.1 综述

Yann LeCun 提出的第一个训练成功的 CNN 模型：LeNet-5。该模型包含有输入层、卷积层、池化层、全连接层和 Softmax 层。

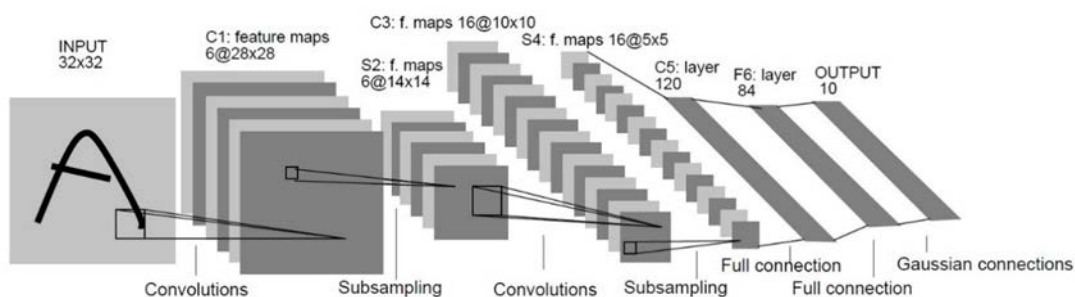


图 3.3 LeNet-5 神经网络模型

.(1) 输入层

输入层是整个神经网络的输入节点。在 LeNet 网络使用灰度图来训练和识别，所以其输入为 $[32, 32, 1]$ ，其中 32×32 是图像的尺寸，而 1 代表的是图像的通道数。

.(2) 卷积层

在本层使用卷积操作对输入的图像矩阵进行处理，以此来提取图像特征。卷积操作增强特定的输入特征，并且降低原始图像的噪音。

.(3) 降采样/池化层

本层采用的是平均池化，通过利用图像的局部相关性原理，对图像进行子抽样。这种方法在保留有用信息的前提下，不仅能有效的降低神经网络的计算量，还能减少过拟合。

.(4) 全连接层

全连接层的本质为矩阵向量乘积计算，该计算使用特征空间变换方法对输入的特征图进行处理，从而集成有用的信息。它的每一层的每个神经元都和上一层所有神经元相连接。该层结构如图 3.4 所示,其函数表达为

$$a_i = w_{x_1i}x_1 + w_{x_2i}x_2 + \cdots + w_{x_ni}x_n + b_i \quad (3.6)$$

$$y_i = w_{a_1i}a_1 + w_{a_2i}a_2 + \cdots + w_{a_ni}a_n + b_{ai} \quad (3.7)$$

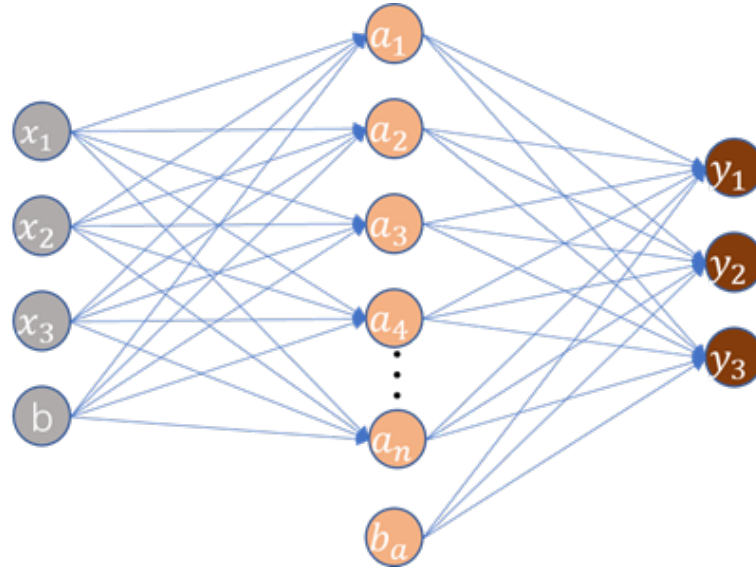


图 3.4 全连接层（从左到右分别为输入层、隐藏层、输出层）

.(5) 激活函数

在神经网络中，激活函数作为一种特殊的映射法则，将特定的输入映射到对应的输出上。如图 3.5 所示，通过对上一层所有的节点的输出值求加权和，然后生成一个非线性的输出值，并将其传递给下一层节点作为输入值。LeNet 中采用的激活函数是 Sigmoid 函数，也称 S 型函数它本身可导且导数非零，容易计算。

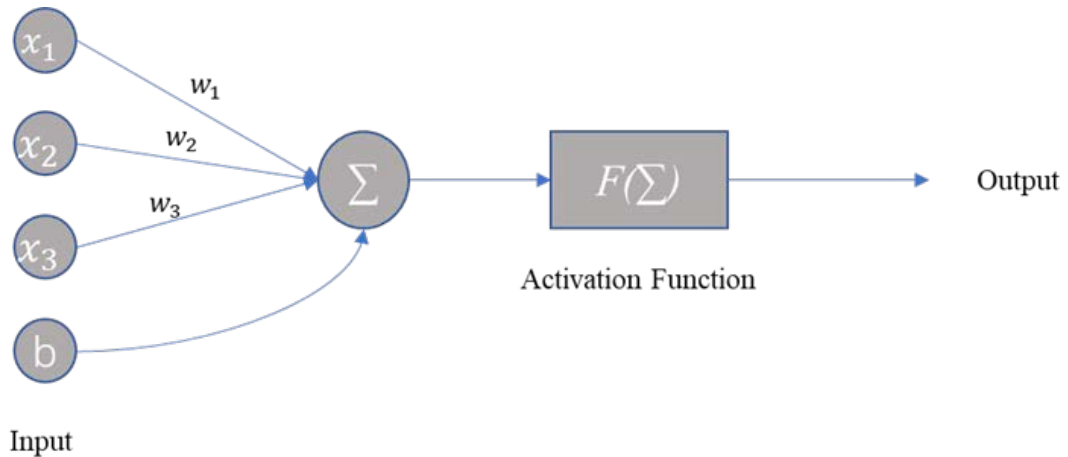


图 3.5 激活函数

Sigmoid 函数定义

$$y = \frac{1}{1 + e^{-(z)}} \quad (3.8)$$

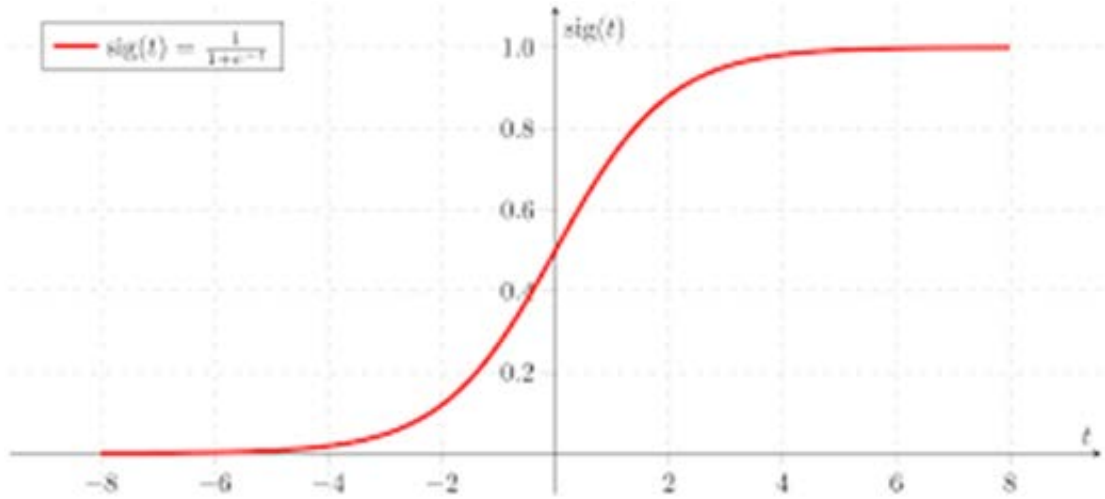


图 3.6 Sigmoid 函数(S 型函数)

.(6) 损失函数

损失函数是用来估算预测值与真实值之间的不一致程度，它是一个非负实值函数。损失函数越小，模型的鲁棒性就越好。而在目前的卷积神经网络中，通常是使用交叉熵作为损失函数。

$$\begin{aligned}
 H(p, q) &= - \sum_x p(x) \log q(x) \\
 &= \begin{cases} p = 0, q \rightarrow 0 \Rightarrow H(p, q) \rightarrow 0 \\ \text{其他} & \Rightarrow H(p, q) > 0 \\ p = 1, q \rightarrow 1 \Rightarrow H(p, q) \rightarrow 0 \end{cases} \quad (3.9)
 \end{aligned}$$

交叉熵描述的是两个概率分布之间的距离， p 代表样本的标签值，而 q 代表预测值。通过交叉熵方法计算出来的值越小，即两个概率分布的值越趋近，则预测越准确。交叉熵损失函数定义如下所示，其中 y_i 为标签值，而 y_i' 为预测值

$$Loss = - \sum_{i=1}^n y_i \times \log y_i' \quad (3.10)$$

3.2.2 特点

LeNet 模型首次使用了卷积方法去提取图像的空间特征，还使用了平均池化方式的池化层来降采样，而在降采样层后面使用了 S 型激活函数。

3.3 AlexNet 网络

3.3.1 概述

AlexNet 网络是 ILSVRC-2012 挑战赛的冠军，也是首个在大规模图像问题中取得突破性进展的深度学习神经网络^[9]。下图为 AlexNet 网络结构

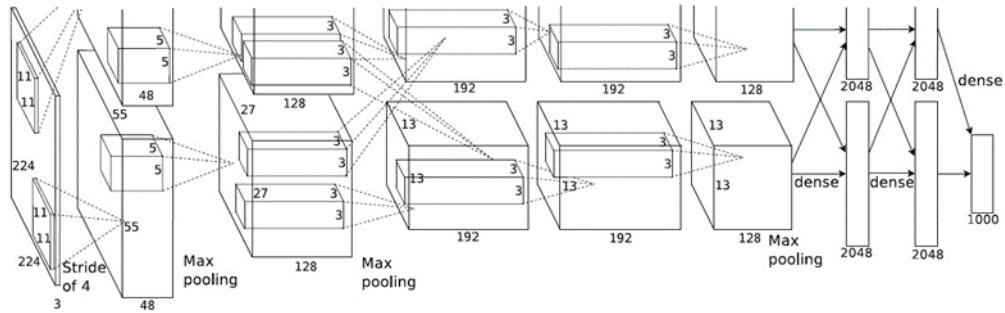


图 3.7 AlexNet 网络

卷积神经网络的一大特点是模块化的设计思维，通过卷积层-激活函数-池化层-标准化的有机组合方式，可以形成多种类型的模块设计。如图 3.7 中的其中一种模块设计就是：卷积层-激活函数(ReLU)-最大池化层。

3.3.2 特点

.(1) 使用 ReLU 函数作为激活函数

ReLU 函数定义

$$f(x) = \max(0, x) \quad (3.11)$$

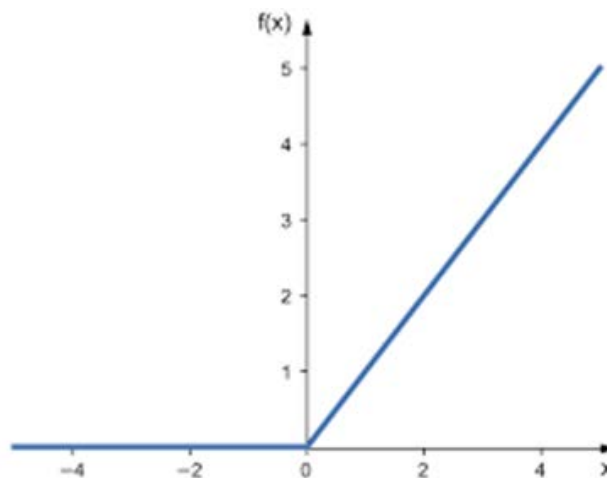


图 3.8 ReLU 函数图像

ReLU 函数有效规避了 S 型函数在神经网络层次加深的时候出现的梯度弥散问题。由于采用梯度下降法去训练 AlexNet 网络，因此饱和和非线性激活函数的学习速度要慢于不饱和和非线性激活函数。

.(2) Dropout 方法

在全连接层中，使之任意忽略掉一部分神经元，从而在一定程度上遏制了过拟合的出现。

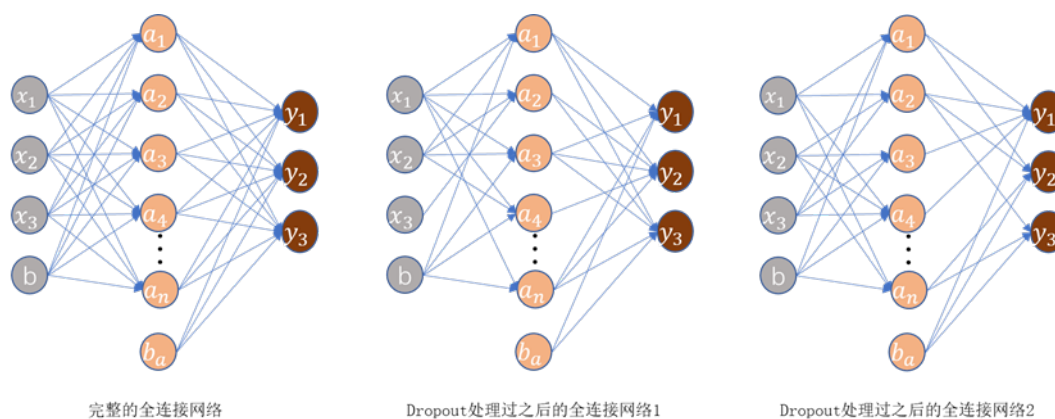


图 3.9 Dropout 方法

.(3) 重叠的最大池化操作

尽管常规的池化操作不会重叠，然而 Alex-Net 使用到了以下方法，其中卷积核尺寸大于移动的步长，从而达到了池化层重叠操作。如图 3.2 所示的就是池化层重叠。采用最大池化方法是为了除去平均池化带来的模糊化结果，这也是一种避免过拟合的手段。

.(4) 使用 GPU 和 CUDA 加速神经网络的训练

.(5) 数据增强

该技术主要是对图像进行剪裁、翻转、镜像、放缩等操作，在数据量较小的情况下增加数据量，从而减少过拟合

3.4 VGGNet 网络

从文章名^[10]就可以看出来，VGGNet 主要是尝试了更深的结构。VGGNet 有深度为 16 的 VGG-16 和深度为 19 的 VGG-19。前者计算量相对较少的情况下获得的准确率比较高，而后者计算量庞大，但是准确率最高。VGG-16 的参数个数高达 1.36 亿，而 VGG-19 更是达到了 1.44 亿。

表 3.1 VGG 网络层数配置

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

表 3.2 VGGNet 各个深度下的单张图片的准确率

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	25.5	8.0

3.4.1 特点

.(1) VGGNet 网络取消了 LRN 层

主要原因是如表 3.2 所示,经过作者在 ILSVRC dataset 上面进行验证,发现 LRN 并不能提升模型的性能,反而会还会增加不少计算量^[10]。

.(2) VGGNet 网络使用单一模块

相比于后面的网络,超参数稀少,网络结构单一。但是相比于经过 Inception、深度可卷积网络等方法优化过的结构,它本身的训练参数过于庞大。

.(3) 使用 3 个 3×3 卷积核来代替 7×7 卷积核

根据“NTN(Network in Network)”的结果,使用三个 3×3 网络来替代 7×7 的大卷积核,既不损失精度,又能提高性能。

.(4) 随着网络加深,图片的高度和宽度不断的以一定的规律缩减

3.5 GoogleNet 网络及 Inception 架构

3.5.1 背景介绍

近些年来，为了获得高质量的模型的最佳做法是增加模型的深度（即层数）和宽度（即每一层的卷积核的数目或神经元的数目）^[11]。但是，这种做法会带来以下问题：

- a) 参数增加过多，若训练集的数据有限，则容易导致过拟合
- b) 网络越大，计算复杂度越大，难以实际应用
- c) 梯度消失会随着网络的加深而愈加严重，最终影响训练的成败。

但是，另一个问题又出现了，那就是计算机的硬件对于非均匀稀疏数据的计算效率很差。

文章中提出使用从稀疏的连接结构的方法去解决上述问题，此方法的依据是 Hebbian 准则。然后为了解决稀疏连接带来的计算效率降低的问题，提出了通过使用多个稀疏矩阵并将其聚集为相关的密集子矩阵来提高计算性能的想法。

3.5.2 InceptionV1

(1) 概述

Inception 结构主要是使用密集组件来替代最优的局部稀疏结构。

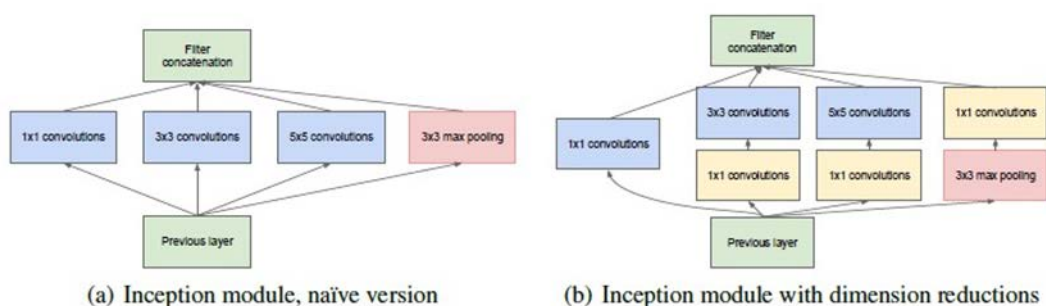


图 3.10 左图为早期版本 右图为改进版本

(2) 不完善的 InceptionV1 结构

在上文中提到，卷积网络是模块化组成的。在高度稀疏的大型深层神经网络中，可以通过分析每个模块最后一层的相关性，逐层构建局部最优的网络拓扑结构，从而将高度相关输出的神经元聚合起来。在深度较浅的层，相关的特征信息都集中在局部，并且将这些信息输出到下一层。而这些信息再被 1×1 的卷积核再次提取。

这种做法对应图 3.11，通过加大单个卷积层的宽度，也就是说在一个卷积层里，使用不同大小的卷积核。而且由于池化层在别的文献中都有不错的表现，以此也在该模块中加入了一个池化层。再通过网络自行调整内部参数，来决定使用哪个过滤器，和是否使用池化层等，增强了网络的自我学习效果

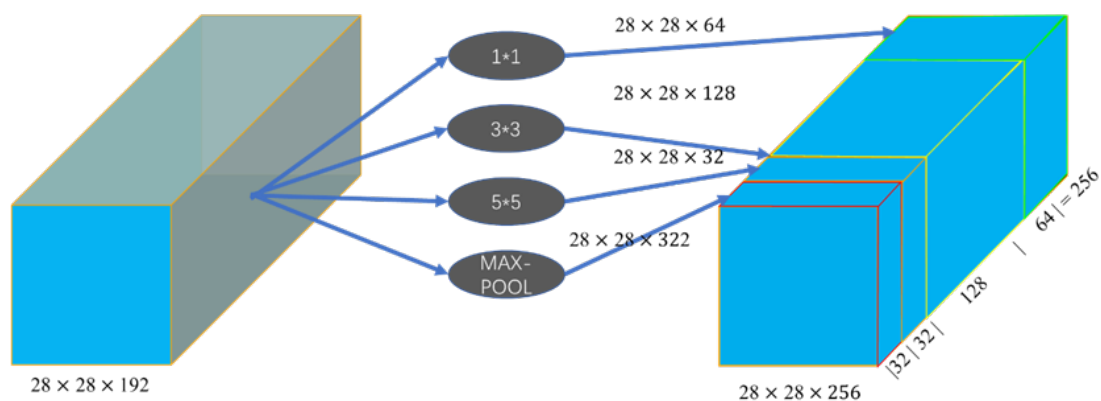


图 3.11 通过不同的卷积核以及池化层实现的聚类

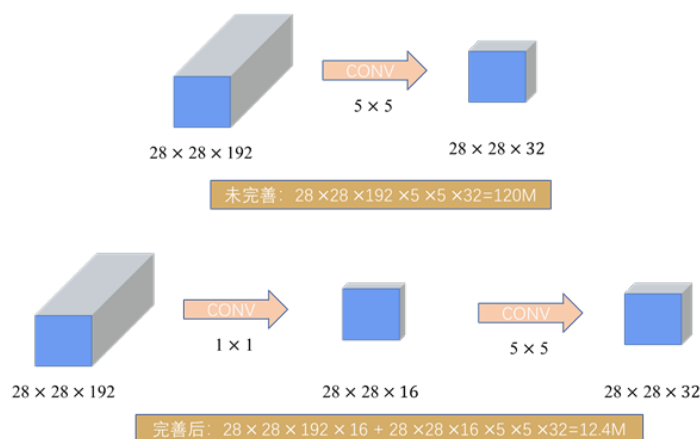
采用尺寸不同的卷积核意味着有不同的感受野，从而可以提取到多种不同尺寸的特征图。而在最后的拼接层中，将不同尺寸的特征图进行融合。

在该论文中采用的卷积核大小尺寸为 1×1 、 3×3 、 5×5 ，是为了更容易对齐。将步长设为 1 后，再分别将填充设为 0、1、2，这样卷积后得到的特征图的大小就一致了，再将这些特征图堆叠聚合。虽然这些特征图的维度一致，但是由于 3×3 和 5×5 的卷积核有填充操作，因此他们提取到的特征的值是不一样的，因此特征图也就不一样。

.(3) 完善后的 InceptionV1 结构

上述的模型存在一个很重要的问题就是，计算量过大，其中 5×5 的卷积共需要 1.2 亿次的计算量，这与这个网络的初衷（减少计算量）是不一致的。

针对这个问题，通过 V1 结构的作者的实践，发现以下网络替代可以有效的解决上述问题。示意图如下所示：

图 3.12 5×5 卷积被替换方式

通过上图可看到，完善后 5×5 卷积核的模型，为完善前的 $1/10$ 。同理，也可以在 3×3 卷积核上应用这一种方法。同时，此结构中的 1×1 卷积层又被称为“bottleneck”（瓶颈层）。因此，完善后的 InceptionV1 结构如下所示

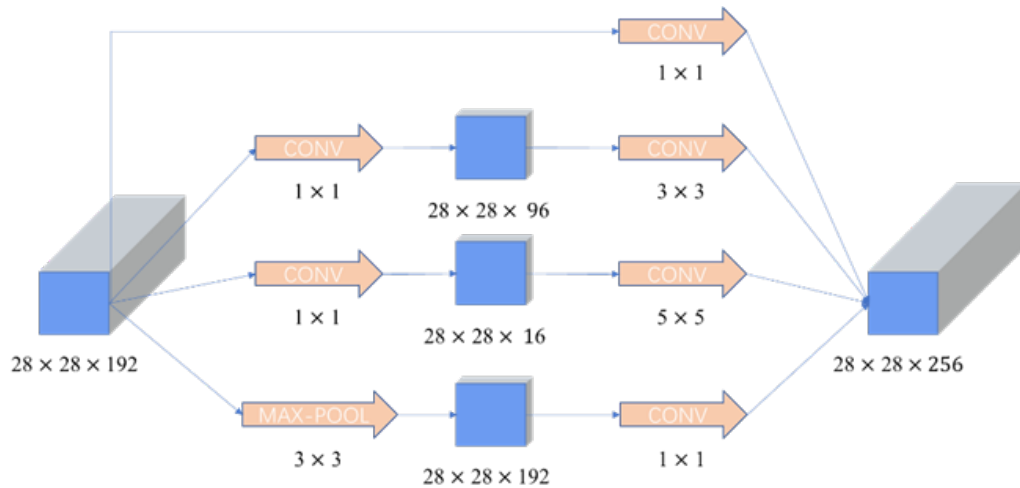


图 3.13 完善后的 InceptionV1 结构

.(4) 特点

a) 深度和宽度：模型总层数达到 22 层，增加了多种卷积核， 1×1 、 3×3 、 5×5 ，以及 MAX-POOL

b) 为了避免上述增加宽度导致的计算量巨大的问题，在 3×3 、 5×5 卷积核前分别加上 1×1 卷积核，来降低计算量。

c) 使用 Average Pooling 代替全连接层，AlexNet 的参数量是 GoogleNet 的 12 倍之多，GoogleNet 的 Top5 的错误率为 6.67%，相比 AlexNet 提高了 10 个百分点。但在最后仍添加一个全连接层，以便以后微调分类输出。

d) 添加了两个辅助分类器。经过论文作者验证，辅助分类器在训练早期对精度的提升不大，但是在训练快要结束的时候，相比于没有添加辅助分类器的模型准确率会更高。

3.5.3 InceptionV2

.(1) 概述

在 InceptionV1 中为了获得不同的感受野，而大的卷积核能带来大的感受野，因此采用了 5×5 的大卷积核。虽然经过了 InceptionV1 结构的优化后，参数减少了很多。但是经过计算，在卷积核数目相同的情况下， 5×5 的卷积核的计算成本高出 3×3 卷积核 2.78 倍。

改进后的结构被称为 InceptionV2 结构。它用两个级联的 3×3 卷积核代替庞大的 5×5 的卷积核。同时，论文提出了一个非常强大的算法，即 Batch Normalization(BN 方法)。

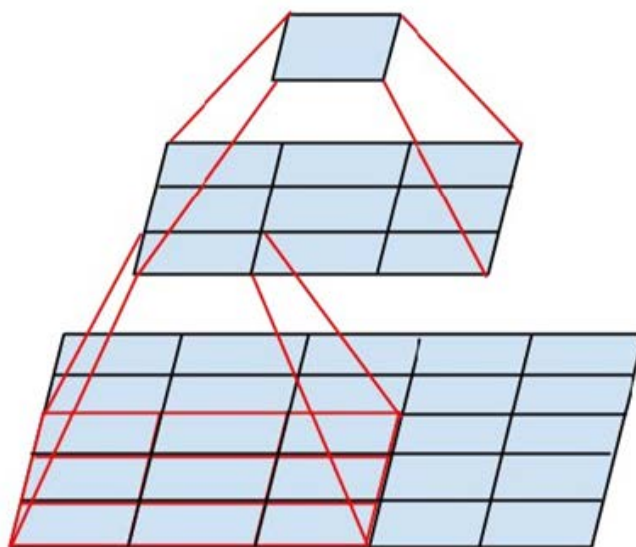
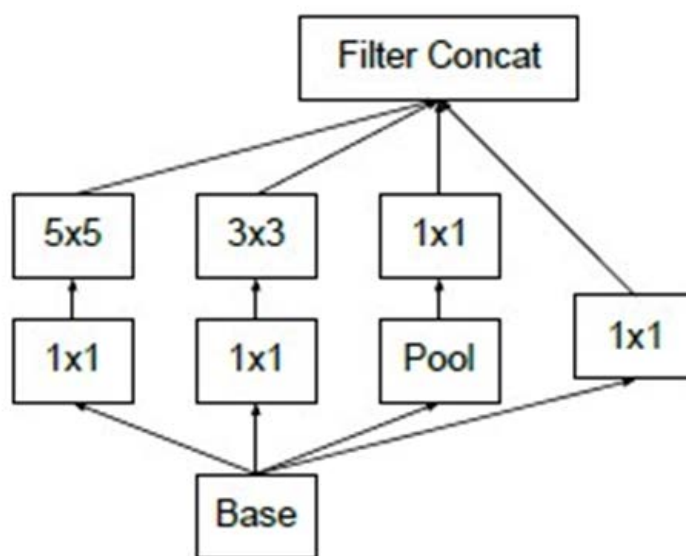
图 3.14 两个连续 3×3 卷积核替代 5×5 卷积核

图 3.15 InceptionV2 结构

.(2) BN 方法

BN 方法的灵感来源于机器学习中的常用的规范化数据的方法：白化。但是这种方法计算成本过高，因此该文提出了 BN 方法，希望既能减少计算量，又能让数据尽可能保留原始的表达能力。

BN 是一类强大的正则化的处理方法，它的作用范围是神经网络中一层所有的 feature map，即它不是对某一张图片的 feature map 进行标准化处理，而是一个小规模的数据样本的所有 feature map，从而使一整层的输出契合 $N(0,1)$ 的正态分布。

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_1 \dots x_m\}$;
 Parameters to be learned: γ, β
Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

图 3.16 BN 方法计算公式

原文中描述：对于一个拥有 d 维的输入 x ，将其所有维度进行标准化处理。例如：输入图像是 RGB 三通道，则 d 等于图像的通道数等于 3，而 $x=(x_1, x_2, x_3)$ ，其中 x_1 就代表 R 通道所对应的特征矩阵。

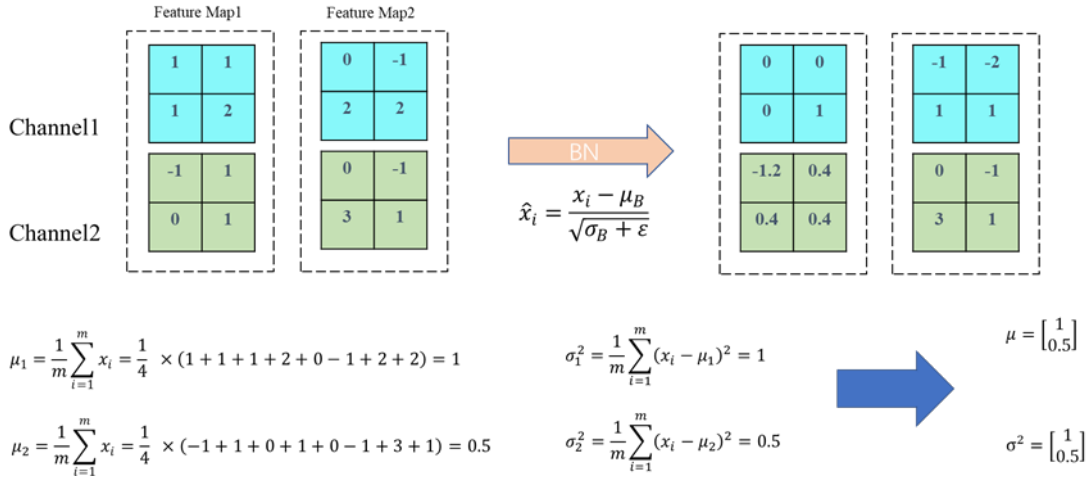


图 3.17 BN 方法计算示例

3.5.4 InceptionV3

(1) 概述

该模型依旧是以 Inception 系列模型为基础进行优化。在 InceptionV2 网络中，我们发现一个 5×5 的卷积核可以被两个 3×3 的卷积核给替换掉，从而明显的减少了参数量。作者在此基础上思考，是否还能对此进行进一步的拆分。从而提出了空间不对称卷积分解的思路，即两个相对小的一维卷积替换掉一个大的二维卷积。

(2) 大卷积核分解

该方法是使用不对称卷积思想，经过作者^[12]实际验证，使用非对称卷积去对大的卷积核进行拆分，即 $n \times 1$ 的卷积核，实际效果比 2×2 的卷积核效果更好。

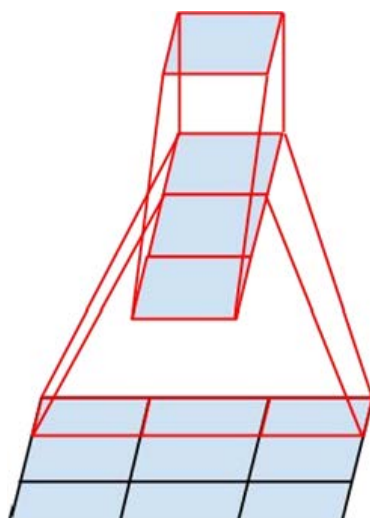
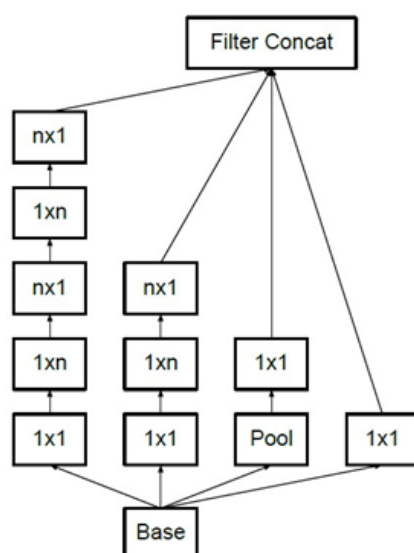
图 3.18 使用 3×1 和 1×3 的非对称卷积替代 3×3 卷积

图 3.19 InceptionV3 结构

如上图所示，使用一个 3×1 和 1×3 的卷积的级联去替代一个 3×3 的卷积。这种方式可以削减一部分网络参数，增加计算速度和降低过拟合程度。而且由于多了一个卷积层，模型的表达能力得到增强。

3.6 ResNet 网络

3.6.1 综述

卷积神经网络的层数对分类识别效果中起很大作用。然而一般的神经网络层数不断堆叠加深，不一定会带来好的效果。下图为何凯明博士针对不同层数的模型做的测试。发现 56 层的深度网络时，表现还不如 20 层的深度。这种称为网络梯度消失或者网络退化。

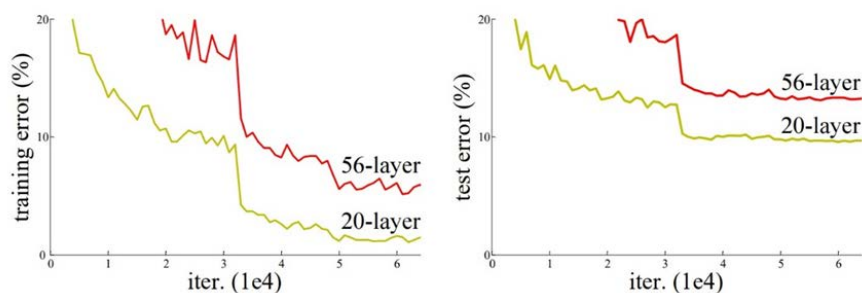


图 3.20 常规深度网络过深会导致深层网络训练效果反而不如浅层网络

3.6.2 残差结构

如图 3.21 所示，作者提出了一个残差函数 $F(x)=H(x)-x$ ， x 为该结构的深入，期望输出是 $H(x)$ ，而堆叠的非线性层结果 $F(x)$ 即为我们学习的目标。以下结构也被称为残差学习单元。

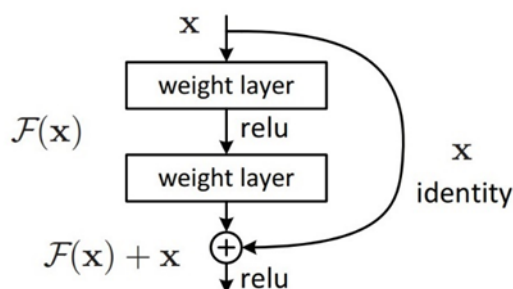


图 3.21 残差块(Residual Block)

文中将输入 x 传输到输出 $H(x)$ 中的方法使用的是“Shortcut Connections”（捷径连接），指的是在网络的输入层添加一个线性层直接连接到输出层。在 ResNet 网络中，shortcut Connections 将输入 x 与组合而成的非线性层的输出相加，即允许原始图像的输入信息直接传输到后面的层中。

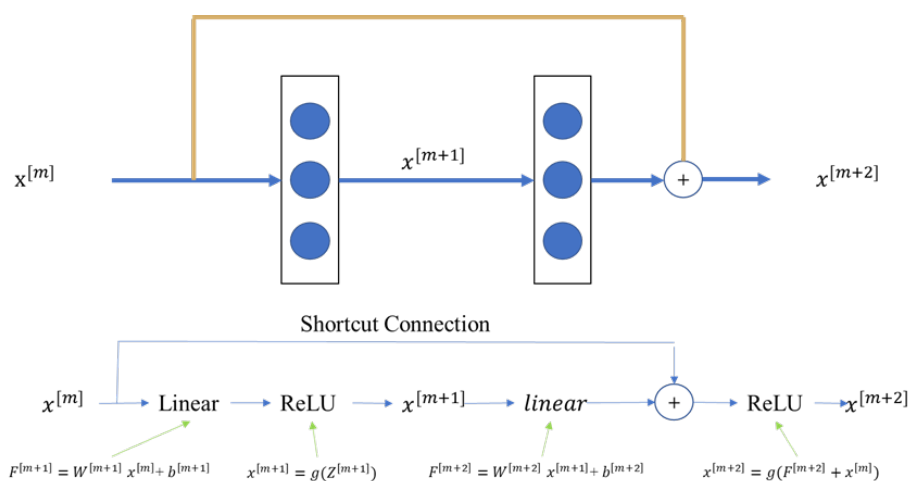


图 3.22 Shortcut Connections 示例

在残差网络中，将 $x[m]$ 复制到更深层的网络节点中，然后在后续节点的激活层前面加上 $x[m]$ ，这样原始信息就可以直接传达到更深层网络。残差块使更深层的网络构造成为可能。因此，可以使用残差块叠加构造 ResNet 神经网络。

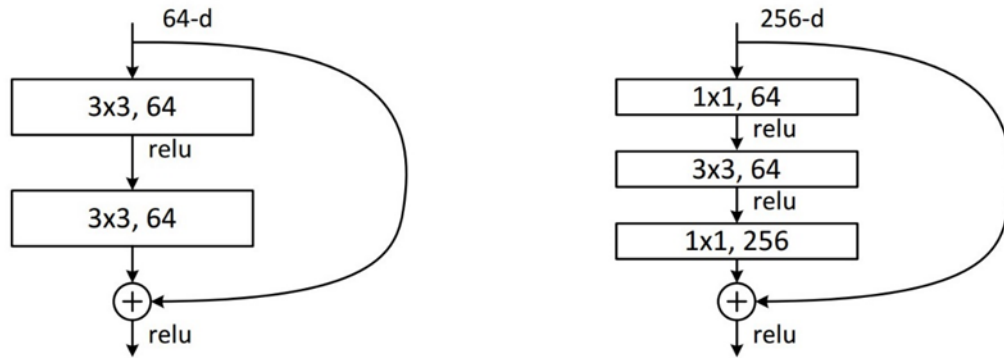


图 3.23 图左为 Option A 图右为 Option B

表 3.3 ResNet 网络层数配置

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

在 ResNet 网络中，深度较浅的网络 ResNet-34 采用的是 Option A 方式构造残差模块，而 ResNet-50 及以上则采用 Option B 方式构建残差模块。其中， 1×1 的卷积核主要是用来降维以及升维使用。堆叠的非线性层与 Shortcut 的输出矩阵必须一致。

3.7 MobileNet

3.7.1 背景介绍

在现实中，如无人驾驶汽车、机器人等产品，使用的往往是性能受限的嵌入式设备，这要求神经网络模型的体积和计算量不能太大。因此，我们需要一个既能保

持高准确率又能在嵌入式设备中高速运行的网络模型^[14]。

3.7.2 MobileNetV1

(1) 深度可分离卷积(Depthwise Separable Convolution)

MobileNet 所用的深度可分离卷积是可分离卷积^[12]的一种变形，它将一般卷积分为深度卷积和 1×1 的逐点卷积两个部分。

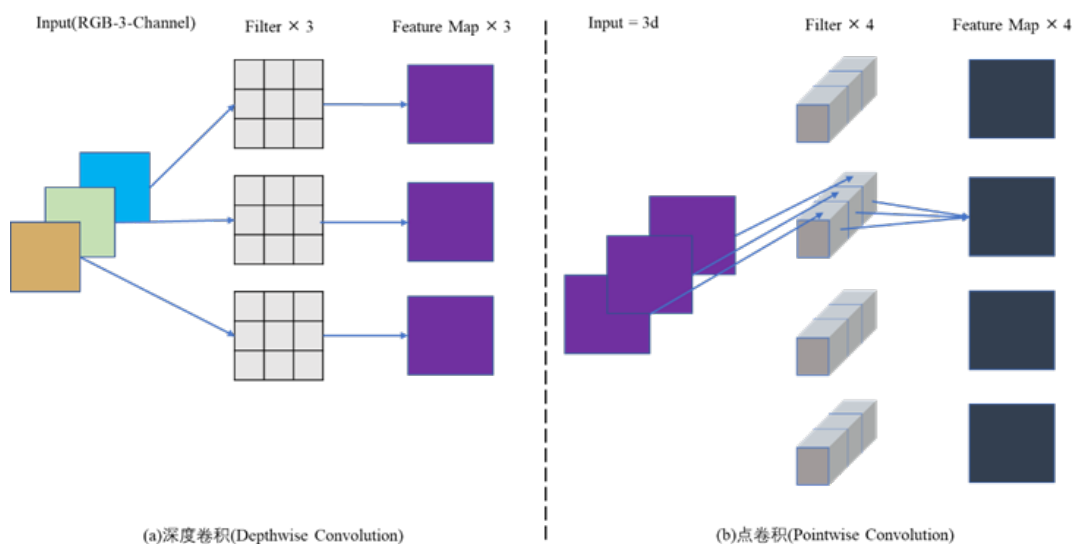


图 3.24 深度卷积与逐点卷积

深度卷积中的每一个卷积核负责一个输入矩阵的 channel，所以输出矩阵的 channel 等于卷积核的个数等于输入矩阵的 channel。而点卷积则是传统卷积，只不过特定卷积核尺寸为 1×1 。通过深度可分离卷积，可以有效的减少模型的参数量，从而提高计算速度。在文中，经过计算后得出的结论是，深度可分卷积的计算速度是普通卷积的 8~9 倍。

(2) 模型收缩超参数(Model Shrinking Hyperparameters)

- 为了使模型的计算量更少，作者增加了两个超参数， α (Width Multiplier)和 β (Resolution Multiplier)。
- 宽度乘数 α (Width Multiplier): 用来改变输入输出的 channel，减少 feature map 的数量，减少网络的厚度。其中 α 的取值为[0,1]。
- 分辨率乘数 β (Resolution Multiplier): 用来改变输入层的分辨率，同样可以减少网络的参数量

3.7.3 MobileNetV2

(1) 倒残差结构(Inverted Residuals)

MobileNetV2 主要是基于 MobileNetV1 进行改进的。为了提升梯度的跨层传播能力，因此将残差结构添加到了 MobileNetV2 网络中，提出了一种新的结构，倒残差结构。

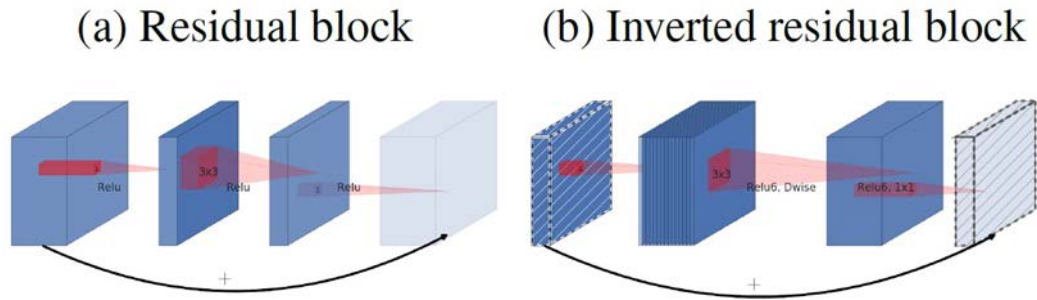


图 3.25 两种倒残差结构

在引用残差结构的时候，作者提出了两种残差结构模型，如图 3.25 所示。经过作者的实验验证，(b)中的倒残差结构有着更好的内存利用效率。

同时为了在嵌入式设备中也能使用 16 位较低精度的浮点数去精确的描述数值，作者使用 ReLU6 替换了 ReLU 非线性激活函数。

ReLU6 函数定义为

$$ReLU6 = \min(\max(0, x), 6) \quad (3.12)$$

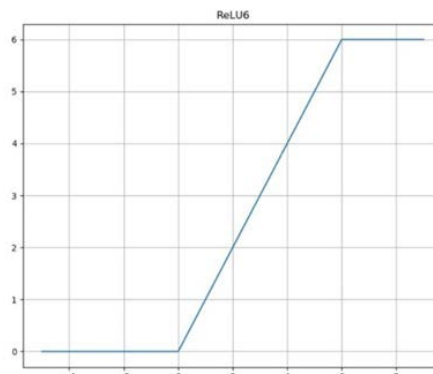


图 3.26 ReLU6 函数图像

.(2) Linear Bottlenecks

作者做了个实验，使用一个二维的图像做输入，然后采用不同维度的矩阵 T 对该图像进行卷积操作并提取相关的图像特征。再使用 ReLU 激活函数得到它的输出值，接着使用矩阵 T 的逆矩阵 T^{-1} 将之前得到的输出矩阵还原。得出如图 3.27 所示结果。从图中可以看出，ReLU 激活函数会对低维特征信息造成较大损失，而高维特征造成的损失不大。

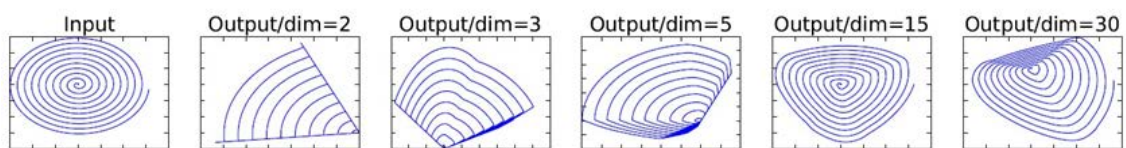


图 3.27 不同维度下使用 ReLU 激活函数后再还原的图像

基于上述所说的问题，因此在 MobileNetV2 结构中的第二个逐点卷积层中去掉 ReLU6 激活函数，直接使用线性连接。

.(3) MobileNetV2 结构及模型

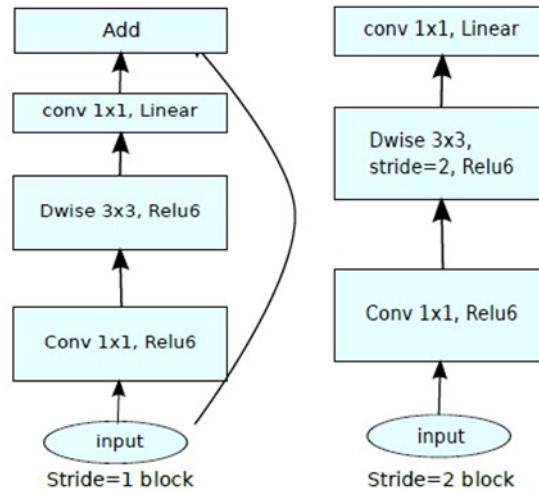


图 3.28 MobileNetV2 结构

在上图中提到在 Stride=1 的时候就需要使用捷径连接，但是在实际上是需要同时满足 stride=1 且输入矩阵与输出矩阵的 shape 相同两个条件时才能使用捷径连接，不然 shape 不同的矩阵根本无法进行加法操作的。

表 3.4 MobileNetV2 网络层数配置

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

表 3.4 中的 t 是拓展因子， c 是输出特征矩阵的深度 channel， n 是 bottleneck 的重复次数， s 是步距（仅针对第一层，其他层均为 1）。

表 3.5 MobileNetV2 网络在 ImageNet 数据集上性能和参数对比

Network	Top 1	Params	MAdds	CPU
MobileNetV1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	3.4M	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NasNet-A	74.0	5.3M	564M	183ms
MobileNetV2	72.0	3.4M	300M	75ms
MobileNetV2 (1.4)	74.7	6.9M	585M	143ms

表 3.6 MobileNetV2 网络在 COCO 数据集上性能和参数对比

Network	mAP	Params	MAdd	CPU
SSD300[34]	23.2	36.1M	35.2B	-
SSD512[34]	26.8	36.1M	99.5B	-
YOLOv2[35]	21.6	50.7M	17.5B	-
MNet V1 + SSDLite	22.2	5.1M	1.3B	270ms
MNet V2 + SSDLite	22.1	4.3M	0.8B	200ms

3.8 本章小结

本章主要介绍了卷积神经网络在图像识别领域的发展脉络。首先从 Alex 这个 ILSVRC-2012 挑战赛的冠军开始，分别介绍了 AGGNet、GoogleNet、ResNet 这些给深度卷积神经网络带来显著进步的网络模型。最后通过对比前面的庞大的卷积神经网络，从而体现了 MobileNet 及 MobileNetV2 网络体积小的优点。

第四章 智能垃圾桶控制系统硬件设计

4.1 智能垃圾桶整体结构分布

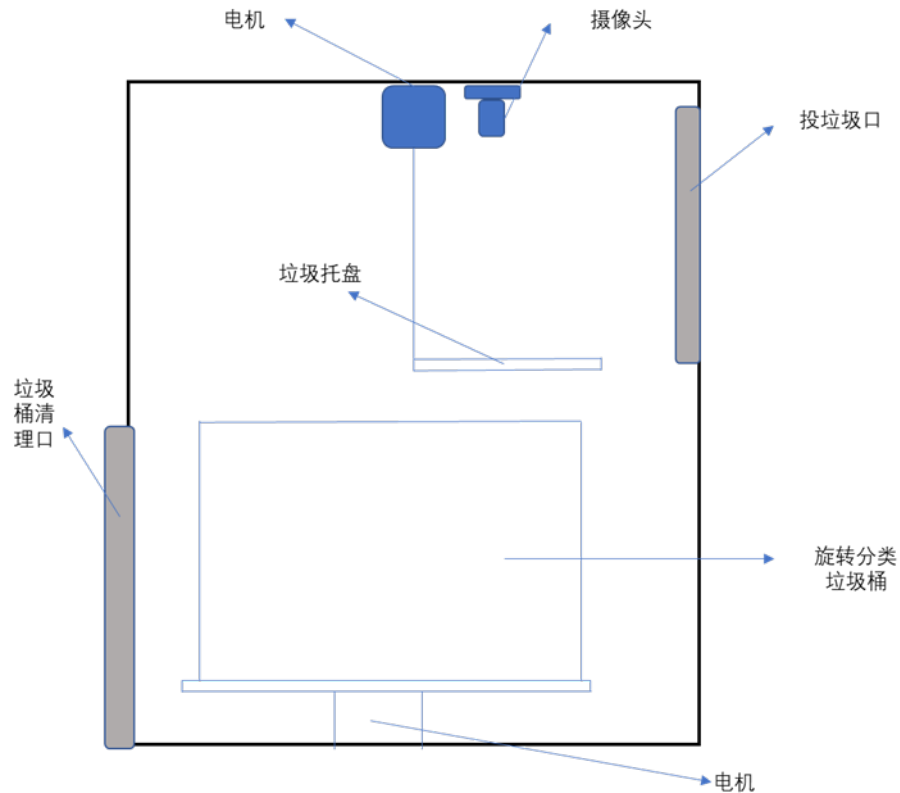


图 4.1 垃圾桶整体结构

4.2 STM32 硬件系统框架

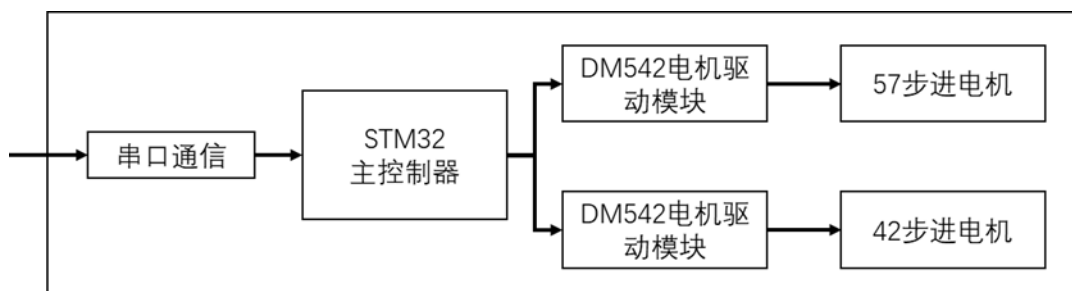


图 4.2 STM32 主控模块

STM32 接收 Jetson Nano 通过串口发送过来的垃圾分类信号，接收到信号之后，判断接收到的垃圾类别。然后根据垃圾类别去驱动 57 步进电机转动，从而带动垃圾桶旋转到对应的分类垃圾桶体。接着驱动 42 步进电机将暂时存放垃圾的托盘拉起，使垃圾下落到垃圾桶中。再将垃圾托盘收起，垃圾桶转回原来位置。

4.3 STM32 微处理器

微处理器也可称单片机，它相比于原来的通用计算机而言，有着成本低廉、使用方式灵活和易于产品化等特点。在本项目中，选择 STM32F429 作为整个主控模块的核心处理器，它是一个 32 位的微处理器，本身主频高达 180M，本身外设资源及其丰富。

4.4 步进电机与步进电机驱动器

4.4.1 步进电机

步进电机是一种能够产生电磁转矩的电磁铁，它接收外部的电脉冲信号并将其转换为角度位移或者直线位移，电磁脉冲信号又被称为数字控制信号^[16]。当它每接收到一个脉冲的信号的时候，便会转动一个角度，这个角度就叫做步距角。因此，如果要它的旋转角度和旋转速度，只需要控制单位时间内发送的脉冲信号的个数即可。

由于本项目要在底部旋转分类垃圾桶，需要一个强有力的电机，因此本项目中采取了 57 步进电机作为动力源。而收放托盘需要的力相对较小，因此可以使用 42 步进电机。

表 4.1 步进电机参数表

步进电机参数								
电机型号	步距角	机身长	相电流	相电阻	相电感	引线数	转动惯量	扭矩
\	°	L(mm)	A	Ω	mH	NO.	g.cm ²	N.m
42BYGH28-401A	1.8	28	1.5	3.0	2.3	4	34	0.28
57HS22	1.8	82	3	1.18	4.6	4	460	2.2

4.4.2 步进电机驱动器

由于本文中所用的步进电机运行所需要的电流比较大，尤其是 57 步进电机需要 3A 的电流，STM32 最小系统本身无法驱动这么大的电流。步进电机驱动器是驱动步进电机运行的功率放大器，它可以提供步进电机所需要的能源。而驱动器性能的优劣会直接影响到步进电机性能的测试结果^[17]。在项目开发中，在前期由于驱动器选取不当，导致步进电机经常发生丢步现象，后来更换了驱动器之后才正常运转。

本文中选取的电机驱动器采用 DM542 芯片，能够提供最大 5A 的电流供给电机

使用。

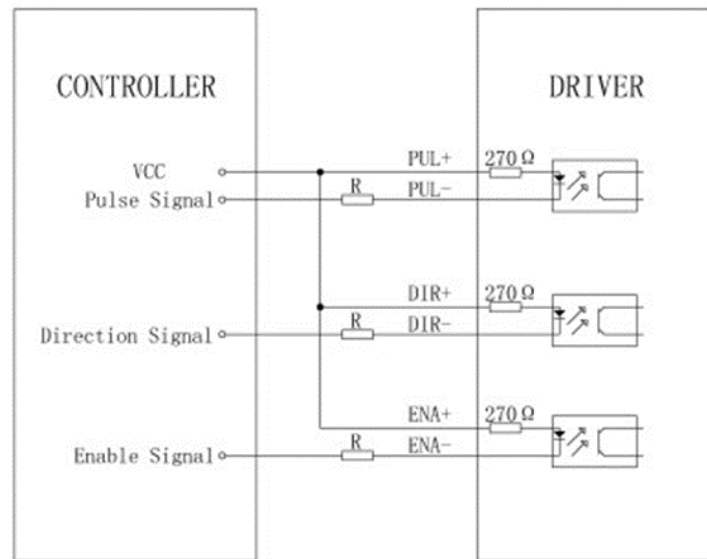


图 4.3 电机驱动器连接图

4.5 Jetson Nano

Jetson Nano 是一款专门为 AI 应用而设计的开发板。采用 1.4Ghz 的 ARM-A57 芯片，128 核 Nvidia Maxwell GPU 和 4GB RAM。它运行的官方系统是 Ubuntu 18.04 的一个定制版本，名称为 Linux4Tegra，主要设计在 NVIDIA 的硬件上运行。该系统预配置了 CUDA 跟 TensorRT 组件，因此可以使用 CUDA 对卷积神经网络计算进行加速。它本身也集成了 UART、I2C 等外设通信组件。

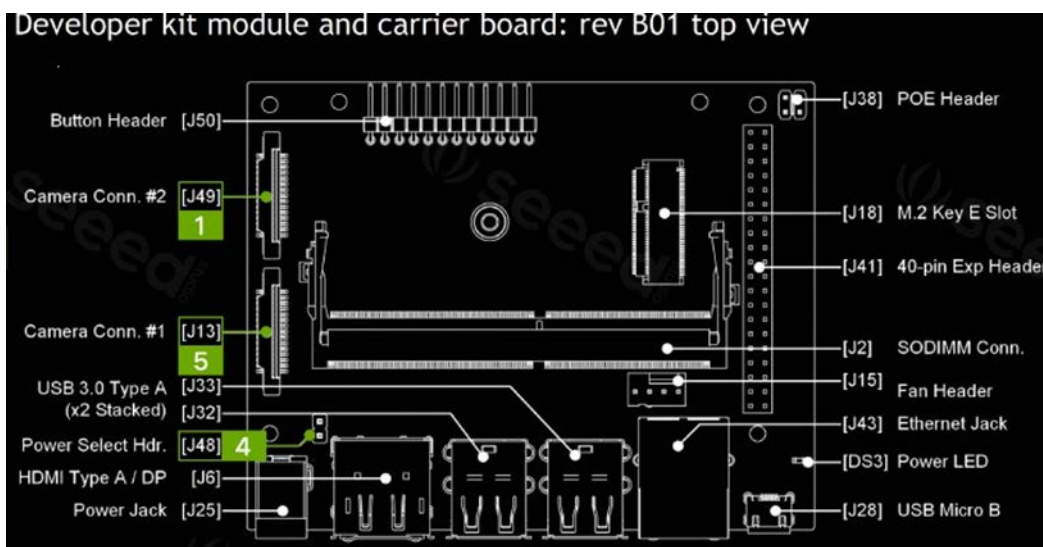


图 4.4 Jetson Nano B01 底板

如图 4.4 所示，Jetson Nano 引出来了 40 个 GPIO 口，布局与树莓派一致。但是 Jetson Nano 的 GPIO 口的电流特别弱，因此使用的时候需要使用额外的供电电路去驱动电子元器件。

本项目最初考虑使用的主处理器是树莓派，但是树莓派本身只有一个 ARM 的 CPU 能够用作神经网络计算，而 Jetson Nano 除了有 CPU 还配置了专为 AI 设计的 GPU。而且显卡上附带的显存对于本身最大内存只有 4G 的机器来说重要性不言而喻。

表 4.2 树莓派 4 与 Jetson Nano 参数对比表

	Raspberry Pi 4	NVIDIA Jetson Nano
CPU	Quad-core ARM Cortex-A72 64-bit @ 1.5 Ghz	Quad-Core ARM Cortex-A57 64-bit @ 1.42 Ghz
GPU	Broadcom VideoCore VI (32-bit)	NVIDIA Maxwell w/ 128 CUDA cores @ 921 Mhz
Memory	4 GB LPDDR4**	4 GB LPDDR4

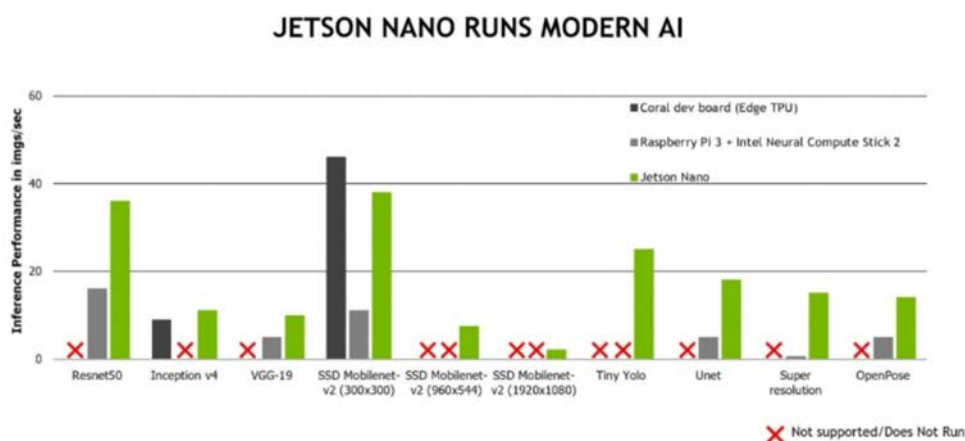


图 4.5 神经计算性能对比

通过上图可以看到，在各种模型计算中，Jetson Nano 的运算性能完胜树莓派加英特尔第二代神经网络计算棒的组合。

4.6 本章小结

本章主要介绍了垃圾桶的物理结构以 STM32、电机和 Jetson Nano 等硬件设备。

第五章 系统实现及测试

5.1 识别算法选取与训练

5.1.1 MobileNetV2

因为要在嵌入式设备上运行，所以模型的计算量和体积是首要的考虑因素，实际使用中，不可能让机器运行好几分钟去分辨识别一次垃圾。而选择进行本地识别，也有一部分原因是基于设备运行时间考虑的。因此，选择计算模型，不仅要考虑模型的准确率，更要考虑模型的计算速度。

表 5.1 各个神经计算模型性能对比表

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet val

如表 5.1 所示，MobileNetV2 以 14MB 的模型尺寸位列列表中最小的计算模型，经典的分类网络如 ResNet、InceptionV3 缩小到了六分之一的大小。而且在 Top1 的准确率上只差了 0.03 左右，Top5 也一样。而这就符合了模型选取的原则，既能高速运行又能保持高准确率。

5.1.2 TensorFlow 框架

TensorFlow 是由 Google 公司开发的机器学习框架，在学术和工业上被广泛使用。在早期的 1.x 版本存在接口复杂且经常变动的问题，使用它构建深度学习网络的需要复杂且反复的编程过程。而在 2019 年 10 月 1 日发布的 TF2.0，则通过多级封装的

形式，简化了 API 接口，使得构建模型时使用的代码更加简洁易懂。TF 自诞生起就存在有训练与部署分离的思想，可以将高性能机器上训练好的模型导出到移动端上部署。而本项目选取该框架就是看重 TF 的训练与部署分离的特性，从而可以在 PC 上快速的训练调优之后，再将模型部署到嵌入式设备中运行。

5.1.3 训练 MobileNetV2 网络

.(1) 样本训练集

样本数据集可分为训练集、验证集和测试集。

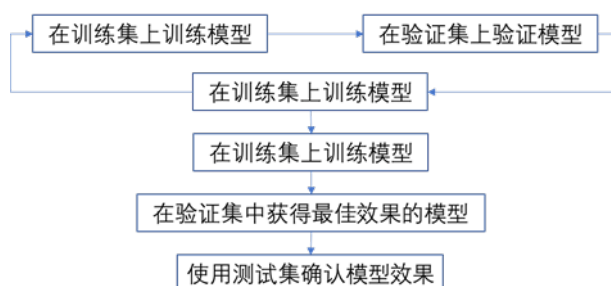


图 5.1 训练、验证、测试集

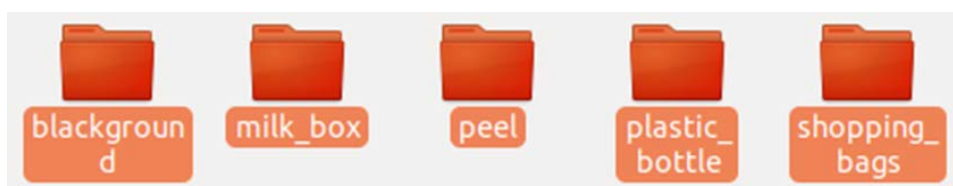


图 5.2 用来训练网络的数据集

如图 5.2 所示，样本数据集有 4 类样本标签和一个背景标签，共 2327 张图片，在本项目中，训练集和验证集比例为 8:2，测试集由预先挑选出的图片进行测试。

.(2) 训练结果

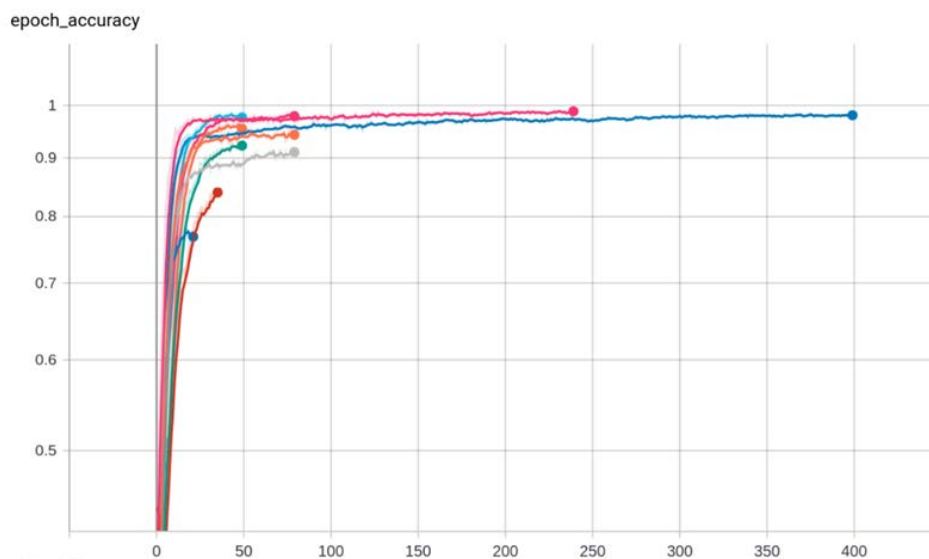


图 5.3 训练集准确率

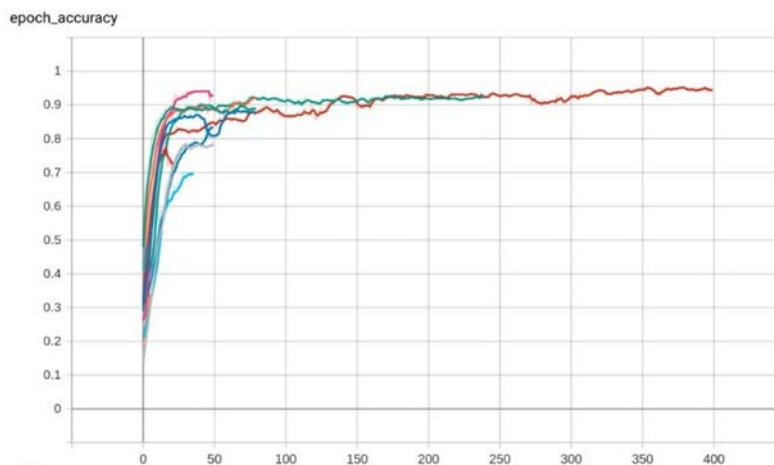


图 5.4 验证集准确率

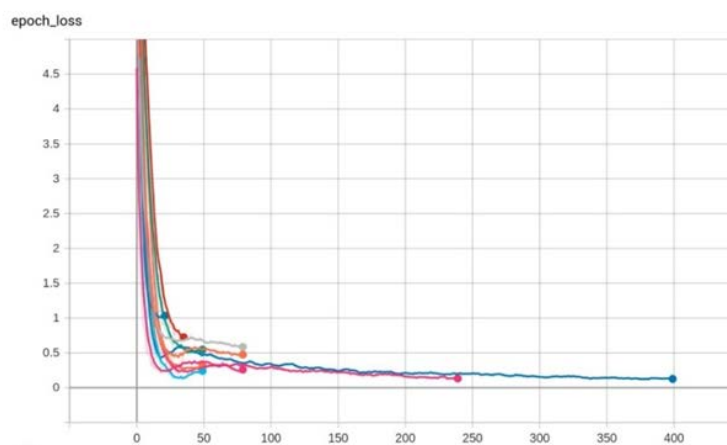


图 5.5 训练集损失函数

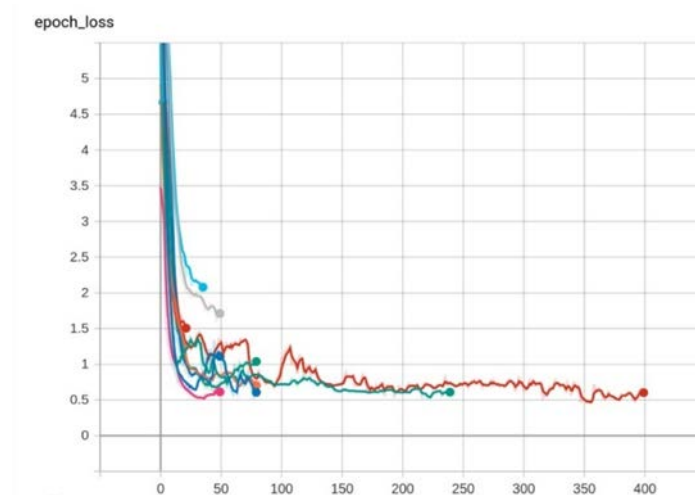


图 5.6 验证集损失函数

图 5.3 图 5.4 图 5.5 图 5.6 是经过多次训练后，选出的几组效果相对较好的训练参数，通过上面图像的比较，选出了最适合使用的一组模型，即训练集正确率 98%，验证集准确率 92%的这组模型。

接着在 PC 电脑上使用事先挑选出来的三十张图片进行测试，结果如下

表 5.2 测试集图片识别成功率

对训练出来的模型进行测试		
测试种类	成功率	误检率
塑料瓶	100%	0
牛奶盒	100%	0
香蕉皮	100%	0
塑料袋	100%	0

5.2 串口通信测试

5.2.1 通信格式

在本项目中，由于处理模块和控制模块是通过串口进行传输信息的，因此需要统一的通信编码标准。由于串口通信会受到如硬件电路等干扰，这有可能会造成信息缺失甚至是信息错误，如信息内容 00 突变为 01 等。所以要加上一个校验位，而奇偶校验在本项目中，由于发送的都是固定的信息，因此可以减少系统处理信息的花销，直接根据信息内容查看校验位即可。

本项目的两个模块之间的串口通信格式为“发送者-接收者-信息内容-校验码”。如 Jetson Nano 要发送对有害垃圾分类处理的信息给 STM32，则串口通信内容为“000100001”。

表 5.3 串口通信格式表

命令内容	发送者	接收者	信息内容	奇偶校验
Jetson Nano 发送垃圾分类处理 信息，垃圾类别为 塑料瓶	00(JN)	01(32)	00(进行垃圾分类处理) 00(塑料瓶)	1
Jetson Nano 发送垃圾分类处理 信息，垃圾类别为 香蕉皮	00(JN)	01(32)	00(进行垃圾分类处理) 01(香蕉皮)	0
Jetson Nano 发送垃圾分类处理 信息，垃圾类别为 塑料袋	00(JN)	01(32)	00(进行垃圾分类处理) 10(塑料袋)	0
Jetson Nano 发送垃圾分类处理 信息，垃圾类别为 牛奶盒	00(JN)	01(32)	00(进行垃圾分类处理) 11(牛奶盒)	1
STM32 发送垃圾处理完 塑料瓶	01(32)	00(JN)	11(垃圾处理完) 00(塑料瓶)	1
STM32 发送垃圾处理完 香蕉皮	01(32)	00(JN)	11(垃圾处理完) 01(香蕉皮)	0
STM32 发送处理完 塑料袋	01(32)	00(JN)	11(垃圾处理完) 10(塑料袋)	0
STM32 发送垃圾处理完 牛奶盒	01(32)	00(JN)	11(垃圾处理完) 11(牛奶盒)	1

5.2.2 通信测试

.(1) 通信注意事项

由于 Jetson Nano 本身系统的问题，将波特率设为 115200 时，Jetson Nano 使用串口发送一连串的数据的时候，PC 平台是能正确接收串口信息的。但是 STM32 单片机在接收 Jetson Nano 发送的信号的时候，不能连续接收一连串的数据，这些数据接收后读取出来发现是乱码。经过多次测试后，发现只能选择每次发送一个字节的的信息，发送一次就要延迟一段时间再发送第二个。在指令内容的最后添加一个 '@' 符号作为结束符，以此标志该次内容发送完毕。

.(2) 主控制器测试

表 5.4 主控制器串口测试表

主控制器串口指令测试			
接收数据指令	预期效果	实际效果	结论
000100001@	垃圾桶旋转至塑料瓶分类格，垃圾托盘放下再拉起	垃圾桶旋转至塑料瓶分类格，垃圾托盘放下再拉起	通过
000100010@	垃圾桶旋转至香蕉皮分类格，垃圾托盘放下再拉起	垃圾桶旋转至香蕉皮分类格，垃圾托盘放下再拉起	通过
000100100@	垃圾桶旋转至塑料袋分类格，垃圾托盘放下再拉起	垃圾桶旋转至塑料袋分类格，垃圾托盘放下再拉起	通过
000100111@	垃圾桶旋转至牛奶盒分类格，垃圾托盘放下再拉起	垃圾桶旋转至牛奶盒分类格，垃圾托盘放下再拉起	通过
100000001@	无反应	无反应	通过
000100000@	无反应	无反应	通过

.(3) 主处理器测试

表 5.5 主处理器串口测试表

主处理器串口指令测试			
接收数据指令	预期效果	实际效果	结论
010011001@	屏幕打印数据指令 010011001	屏幕打印数据指令 010011001	通过
010011010@	屏幕打印数据指令 010011010	屏幕打印数据指令 010011010	通过
010011101@	屏幕打印数据指令 010011101	屏幕打印数据指令 010011101	通过
010011111@	屏幕打印数据指令 010011111	屏幕打印数据指令 010011111	通过
010011000@	无反应	无反应	通过
000100100@	无反应	无反应	通过

5.3 实际垃圾分类测试

将控制模块跟处理模块连接后，实际投放垃圾到本产品中进行测试。由于实验条件有限，因此本次实际测试仅使用以下材料进行测试。



图 5.7 实际测试材料

一次在垃圾托盘中只放入一个垃圾，然后 Jetson Nano 对垃圾进行识别，识别完之后在屏幕上打印结果，并且将结果通过串口发送给 STM32。而 STM32 在接收到信号之后，就转动垃圾桶，从而使垃圾落入对应的桶体中。

以下是上面测试材料的概率分布，每个测试材料的类别都识别正确，只是识别的概率不一致。

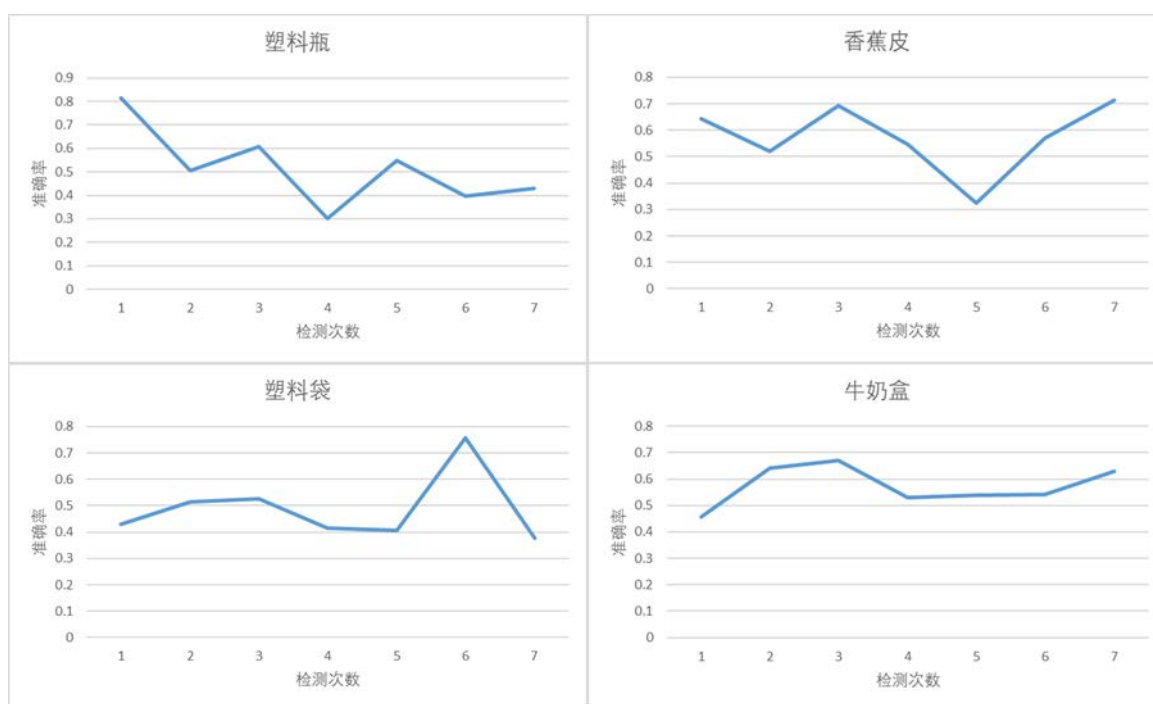


图 5.8 实际测试材料的识别效果概率分布

5.4 本章小结

本章主要是对智能分类垃圾桶进行总体测试。首先是先把分类识别的深度学习模型训练好，然后解释说明了控制模块和识别模块之间的通信规则，最后是对垃圾桶进行实际测试。直接往桶里面投放垃圾，看看垃圾识别的概率分布已经有没有正确的把垃圾投放到对应的桶中。实验证明，本垃圾桶在分类上基本可以做到准确分类。

第六章 总结与展望

目前实现的主要功能是在本地进行垃圾分类识别，再将垃圾投放到对应的桶中，在 Jetson Nano 板上识别垃圾仅需 0.4S，这个速度对于使用远程网络识别的方式来说快了很多倍。

但是，该项目存在以下问题：

- a) 项目主要是本地进行部署，所以后续模型升级困难
- b) 使用的 Jetson Nano 开发板价格较贵，成本较高
- c) 桶的机械结构不够完善
- d) 模型的准确率不够高，还是会存在一些误判行为

因此接下来的工作是：

- a) 结合物联网技术，对本地运行的产品，实现联网可自动更新深度学习模型，提升设备的使用体验
- b) 目前的科技一直在发展，等待更好更便宜的硬件，整体降低制作成本
- c) 改进桶的机械结构，使它投放垃圾的速度更快更准。
- d) 继续优化学习模型

参考文献

- [1] 中华人民共和国生态环境部.2019 年全国大、中城市固体废物污染环境防治年报, [EB/OL],
<http://www.mee.gov.cn/ywgz/gtfwyhxpj/gtfw/201912/P020191231360445518365.pdf>,2019-12
- [2] 刘建伟,何岩.餐厨垃圾两相厌氧发酵技术研究和应用进展[J].科学技术与工程,2017,17(06):188-196.
- [3] 卢锋长,欧阳玲惠,王晶,李海艳,曾凡.浅析我国垃圾分类工作目前存在的问题及解决方案[J].市场周刊,2019(09):152-154.
- [4] 卢焱.发达国家和地区生活垃圾分类管理模式、历程与机制[J].科学发展,2019(03):87-97.
- [5] 黄国维. 基于深度学习的城市垃圾桶智能分类研究[D].安徽理工大学,2019.
- [6] WarrenS McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 1943, 5(4):115 — 133.
- [7] D E Rumelhart , G E Hinton , R J Williams. Learning internal representations by error propagation. Nature , 1986, 323(99):533 — 536
- [8] Hubel, D. and Wiesel, T. (1968).Receptive fields and functional architecture of monkey striate cortex.Journal of Physiology (London), 195, 215–243.
- [9] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks [J]. Advances in Neural Information Processing Systems, 2012, 25(2):2012
- [10] Simonyan, Karen and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” CoRR abs/1409.1556 (2014): n. pag.
- [11] Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed,D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich.Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,pages 1–9, 2015.
- [12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed,D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich.Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,pages 1–9, 2015.
- [13] He, Kaiming et al. “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 770-778.
- [14] Howard, Andrew G. et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.” ArXiv abs/1704.04861 (2017): n. pag.
- [15] Sandler, Mark et al. “Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation.” ArXiv abs/1801.04381 (2018): n. pag.
- [16] 张涛. 步进电机快速定位方法研究[D].北方工业大学,2012.

- [17] 李玲娟. 多细分二相混合式步进电机驱动器的研制[D].西北工业大学,2007
- [18] 杨真真,匡楠,范露,康彬.基于卷积神经网络的图像分类算法综述[J].信号处理,2018,34(12):1474-1489.
- [19] 沈则瑾. 上海“史上最严”垃圾分类已两月,咋样了,[EB/OL],http://www.xinhuanet.com/tech/2019-09/11/c_1124984284.htm,2019-09.
- [20] 钟梅,马宇龙,杨谢泽华.集中分拣回收,解决垃圾围城[J].北方环境,2013,25(01):62-64.

致谢

时光匆匆，转眼间大学本科四年的学习生涯即将结束。回首这四年，努力过，也迷茫过。庆幸的是，一路走来，得到了老师、同学和亲人们的关心和帮助。借论文完成之际，向你们表示衷心的感谢和祝福。

首先要感谢的是杨斌老师，不仅为我的论文的撰写提供了很多宝贵的意见，而且在写论文期间，心情低潮时，也给予鼓励。在我迷茫时，也多次与我谈心，帮助我坚定目标，鼓励我继续努力奋斗。

其次要感谢的是艾广燊老师，在大学四年中，教会了我许多的不仅仅是专业方面的知识，还有宝贵的学习和生活经验，并且在迷茫的时候给予鼓励，是一位很好的老师。

再者要感谢的是电子系的全体老师，谢谢你们这四年的悉心栽培和关心爱护，让我在学到知识的同时，也学会了许多的人生道理。

最后要感谢是广东东软学院，感谢你们四年以来的栽培。

2020 年 4 月 20 日