学号：202071553　　密级：

长江大学

硕士研究生学位论文

基于树莓派的吸烟手势检测研究

专业领域：　计算机科学与技术

研究方向：　机器学习与人工智能

研　究　生：　徐元聪

指导教师：　崔艳荣　教授

论文起止日期：2022 年 4 月至 2023 年 5 月

学号：202071553　　密级：

长江大学

硕士研究生学位论文

基于树莓派的吸烟手势检测研究

专　　业：计算机科学与技术

研究方向：机器学习与人工智能

研 究 生：徐元聪

指导教师：崔艳荣 教授

论文起止日期：2022 年 4 月至 2023 年 5 月

A Research on Smoking Gesture Detection Based on Raspberry Pi

Major：Computer Science and Technology

Direction of Study：Machine Learning and Artificial Intelligence

Graduate Student： Xu Yuancong

Supervisor： Prof. Cui Yanrong

School of Computer Science

Yangtze University

April,2022to May,2023

摘 要

吸烟能导致各种危害身体健康的疾病，全世界每年有大量的人因为吸烟而死亡。同时，不当的吸烟行为也会造成许多安全事故，因废弃烟头导致的火灾等层出不穷，造成许多经济损失和人员伤亡，因此实现对吸烟行为的快速、准确识别是一个具有重大意义的课题。与传统的人工监督与烟雾报警器效率低且花费高相比，基于深度学习图像分析的吸烟行为检测具有监控范围广、成本低且效率高等优点。但现有的吸烟手势检测模型存在一些不足：（1）为了保障检测精度，模型往往设计的十分复杂，导致模型参数量大、计算量大，使得模型难以部署和推理速度慢，无法满足快速准确的吸烟行为检测需求。（2）检测速度快的轻量级的模型往往在精度上有所欠缺，不能准确识别出吸烟行为。

针对上述问题，本文主要工作如下：（1）在轻量级的单目标检测算法 YOLOv7-tiny 的基础上进一步轻量化改进。在网络结构部分引入 Ghost 模块和 Ghost 瓶颈结构，并对特征融合部分结构进行裁剪，使得整个模型更轻量，实现吸烟手势快速检测。（2）为了保障轻量化改进后的 YOLOv7-tiny 的检测精度，在网络结构部分引入 CARAFE 上采样算子和 ECA 通道注意力模块，并在聚类算法和损失函数上选用 K-medoids 聚类算法和 Alpha-IoU 损失函数，在不影响模型轻量化的同时提升了模型检测精度，改进后得到的新算法命名为 YOLOv7-tl(YOLOv7-tiny-lite)。

实验结果显示，改进后的 YOLOv7-tl 算法对比 YOLOv7-tiny，mAP 等各项指标均得到提升。除此之外，对比 YOLOv5s、YOLOX-tiny 等算法，YOLOv7-tl 在 mAP、参数量、计算量、模型大小和 FPS 等五项指标上均远远超出。实验结果表明了本文所提出的 YOLOv7-tl 算法的先进性，并实现了精度和速度上的平衡。此外，本文将改进后得到的 YOLOv7-tl 算法应用在树莓派上，实现了一个吸烟手势检测系统。

本文创新之处如下：（1）针对当前吸烟手势检测算法的不足，提出了基于 YOLOv7-tiny 改进的 YOLOv7-tl 算法，实现了精度和速度上的平衡。（2）将改进后的YOLOv7-tl算法模型与边缘设备树莓派结合，构建了一个吸烟手势检测系统，具有识别准确率高、速度快、成本低等优势，具有一定实际意义。

关键词： 吸烟手势检测，深度学习，卷积神经网络，YOLOv7-tl，树莓派

Abstract

Smoking can lead to various diseases that harm physical health, and a large number of people die every year worldwide due to smoking. In addition, improper smoking behavior can also cause many safety accidents, such as fires caused by discarded cigarette butts, resulting in economic losses and casualties. Therefore, it is of great significance to achieve rapid and accurate identification of smoking behavior. Compared with traditional manual supervision and smoke alarms, smoking behavior detection based on Deep Learning image analysis has the advantages of a wide monitoring range, low cost, and high efficiency. However, there are some shortcomings in existing smoking gesture detection models: (1) the models are often designed to be very complex to ensure detection accuracy, which leads to a large number of model parameters and computations, making it difficult to deploy the model and slow down its inference speed, and cannot meet the requirements of fast and accurate smoking behavior detection. (2) Lightweight models with fast detection speeds often lack accuracy and cannot accurately recognize smoking behavior.

To address the above issues, this paper's main work is as follows: (1) further lightweight improvements are used based on the lightweight single-object detection algorithm YOLOv7-tiny.Ghost module and Ghost bottleneck structure are introduced into the network structure, and the feature fusion part of the structure is cut, making the whole model more lightweight, to realize the rapid detection of smoking gestures. (2) To ensure the detection accuracy of the lightweight optimized YOLOv7-tiny, the CARAFE upsampling operator and ECA channel attention module are introduced in the network structure, and the K-medoids clustering algorithm and Alpha-IoU loss function are selected for clustering algorithm and loss function. These improvements improve model detection accuracy without affecting model lightweight. The new algorithm is named YOLOv7-tl (YOLOv7-tiny-lite).

Experimental results show that compared with YOLOv7-tiny, mAP and other indexes of the improved YOLOv7-tl algorithm are improved. In addition, compared with other algorithms such as YOLOv5s and YOLOX-tiny, YOLOv7-tl far exceeds five indicators such as mAP, Param, GFLOPs, Size, and FPS. Experimental results demonstrate the advancedness of the YOLOv7-tl algorithm proposed in this paper and YOLOv7-tl achieves a balance between accuracy and speed. In addition, the improved YOLOv7-tl algorithm is applied to Raspberry Pi to realize a smoking gesture detection

system.

The innovations of this article are as follows: (1) In response to the shortcomings of current smoking gesture detection algorithms, an improved YOLOv7-tl algorithm based on YOLOv7-tiny has been proposed, achieving a balance between accuracy and speed. (2) By combining the improved YOLOv7-tl algorithm model with the edge device Raspberry Pi, a smoking gesture detection system has been constructed with advantages such as high recognition accuracy, fast speed, and low cost and has certain practical significance.

Key words: Smoking Gesture Detection, Deep Learning, Convolutional Neural Network, YOLOv7-tl, Raspberry Pi

目　录

第 1 章 绪论

1.1 研究背景及意义

吸烟能导致严重的呼吸道疾病、肺病及心血管疾病，还能诱发其它的疾病。调查显示，每年约有 1700 万人死于心脏病，烟草及二手烟是主要病因[1]。据世界卫生组织报告，目前，在全球估计有 10 亿吸烟者，约 80%生活在低收入和中等收入国家(LMICs)，烟草每年造成 800 万人死亡，其中包括 100 万人死于二手烟[2]。在美国，吸烟每年造成的死亡人数约占全年死亡人数的 20%，同时也增加了许多严重疾病的发病几率。美国每年与吸烟有关的疾病花费 3000 亿美元，其中包括直接医疗和因吸烟引起的疾病导致的生产力损失[3,4]。在欧洲，目前大约有四分之一的成年人吸烟，其中包括三分之一的男性和五分之一的女性[5]。在中国，吸烟人数超过 3 亿，15 岁以上人群吸烟率为 26.6%，其中男性吸烟率为 50.5%。每年 100 多万人因烟草失去生命，预计到 2030 年将增至每年 200 万人，到 2050 年将增至每年 300 万人[6]。除此之外，吸烟产生的烟头也会导致环境问题和安全隐患，因废弃烟头导致的火灾层出不穷，造成许多经济损失和人员伤亡。

随着对吸烟危害的逐步认识，世界各国都采取了相应的措施来控制这种局面，例如规定在一些公共及特殊场合禁止吸烟、提升烟草价格、禁止未成年人购买烟草等。中国对此也采取了许多措施来禁止吸烟，《中华人民共和国烟草专卖法》第二章第五条规定：国家和社会加强吸烟危害健康的宣传教育，禁止或者限制在公共交通工具和公共场所吸烟，禁止中小学生吸烟；《公共场所卫生管理条例实施细则》第二章第十八条规定：室内公共场所禁止吸烟，公共场所经营者应当设置醒目的禁止吸烟警语和标志，并配备专（兼）职人员对吸烟者进行劝阻等[7]。这些措施收获了一定成效，但仍然有人罔顾规定在禁止吸烟场合吸烟。

传统的吸烟监控方法依赖于人工，通过人力监控，对吸烟者进行劝阻或是处罚；或是通过烟雾报警器等边缘端设备，对吸烟产生的烟雾粒子进行检测。但人力监控会消耗大量的人力物力，而且可能达不到很好的效果[8]。而现有烟雾报警器等边缘端设备多针对火灾，吸烟产生的少量烟雾不易被检测[9]，同时还会受到环境大小和密封性影响，并会受到灰尘粒子、气体粒子等干扰，不能及时检测出吸烟行为[10]。由于吸烟导致的诸多危害和当前的检测现状，研究实现对吸烟的快速准确检测具有重要意义，而随着人工智能深度学习的蓬勃发展，给解决这个难题提供了新的思路。

目前基于深度学习的吸烟行为检测一般分为三种：香烟识别、香烟烟雾识别和吸烟手势识别[11]。但对香烟进行识别则往往受限于香烟本身体积小，且容易被其他

相似物品所干扰，导致识别出错，不能很好的识别出吸烟行为[12]。同样，对比传统的烟雾报警器，通过深度学习来识别香烟烟雾更准确，能更好的检测出吸烟行为，但吸烟产生的烟雾存在浓度低、发散快的特点，且烟雾颜色容易被采集图像的背景所干扰，导致检测不稳定[13]。现有的深度学习检测模型为了保障检测精度，往往设计的十分复杂，导致模型参数量大、计算量大，而需要进行吸烟行为检测的地方复杂多样，搭载模型的检测设备多处于网络的边缘，存在计算、延迟、带宽和能耗等方面的问题，过于复杂的模型会导致模型难以部署和推理速度慢，无法满足快速的吸烟行为检测需求。而检测速度快的轻量级的模型往往在精度上有所欠缺，不能准确识别出吸烟行为。

本文针对上述的问题展开研究，采用深度学习的方式，来解决传统方式的成本高、精度低的不足，同时选用吸烟手势作为检测目标，吸烟手势对比香烟和香烟烟雾，具有体积大、特征明显等优势。并选用轻量级深度学习卷积神经网络模型来进行吸烟手势检测，并对模型进行研究改进，使之更轻量和更准确，来满足边缘检测设备的需求。并使用树莓派充当边缘设备作为载体进行模型部署，最后实现一个快速、准确的吸烟手势检测系统。

1.2 国内外研究现状

吸烟能造成极大的健康危害和安全隐患，随着近些年社会各界对吸烟危害的逐渐认识，针对吸烟行为的快速、准确检测成为了十分迫切的需求。相较于香烟和香烟烟雾，吸烟手势以其特征明显的优势受到国内外的广泛研究，伴随着深度学习的快速发展，深度学习领域相关的吸烟手势研究也层出不穷，大致可以分为两个类型，一种是结合可穿戴式设备进行研究，判断是否出现了吸烟手势和动作。另一种是对采集的图像进行分析，检测是否存在吸烟手势。本小节将针对以上两个研究类型进行概述。

1.2.1 穿戴式设备检测方法

结合可穿戴式设备的深度学习吸烟手势检测研究的主要原理为通过可穿戴式设备采集吸烟手势信息，并用采集到的数据训练神经网络模型，然后通过训练好的模型来判断是否出现了吸烟手势，产生了吸烟行为。

Cole 等[14]将智能手表与人工神经网络(Artificial Neural Network, ANN)结合，用智能手表采集到的多维数据提训练人工神经网络，生成了能有效识别吸烟手势的模型。并在训练模型时发现，使用采集到的三维数据训练的神经网络模型表现优于一维和二维数据训练的模型。其中智能手表加速度计产生的 x 维信息能有效识别吸烟姿势，而 y 维和 z 维信息能有效消除吃、喝和抓挠鼻子等其他姿势产生的误

判，最终对吸烟行为的识别准确率超过了 90%。

Maguire 等[15]在智能手表的基础上加装手指弯曲传感器，以识别拔出香烟的动作，减少吃饭、喝水等相同手到嘴动作带来的误判。并集成了一个可穿戴的空气质量传感设备，能够将吸烟行为背景化，减少虚假吸烟通知。最后在定制测试数据集上训练的神经网络分类器分析收集的数据，并通过智能手机应用程序通知是否检测到吸烟或吸烟前活动，并制定了测未来吸烟行为的系统，分类准确率达到了 80.6%。

Añazco 等[16]使用一个包含单个惯性单元传感器的腕带来测量日常活动信息，包含吸烟手势、喝酒、喝水等动作，并将收集到的信息制作为数据集用以训练深度学习模型。采用循环神经网络(Recurrent Neural Network, RNN)作为分类算法，并采用长短期记忆网络(Long Short-Term Memory, LSTM)来避免梯度消失和爆炸问题，通过时间反向传播更新权重，随机梯度下降和 NesterOV 动量的组合来优化权重，最终对吸烟活动的识别准确率达到了 91.38%。

Alharbi 等[17]等首次将卷积神经网络(Convolutional Neural Networks, CNN)应用于可穿戴式设备的训练及手到嘴的吸烟手势及动作识别，将智能手表和智能手机的加速计和陀螺仪的数据作为模型的输入来训练模型，并设计了一种适合该数据的 CNN 架构，模型最终取得了 96%左右的 F1 分数。

Odhiambo[18]等将使用智能手表传感器收集到的吸烟手势样本重新表述为转态转换模型，由手对嘴唇开始，手对嘴唇的持续时间和手离嘴唇三个小手势组成，有效的降低了检测的复杂性。同时采用网络匹配的标注的目标值和预测值来计算损失，并使用均方误差损失函数来量化人工神经网络的预测期望输出，最终将所得到的模型准确率提高到了 99%，能准确识别吸烟手势。

以上研究表明穿戴式设备结合深度学习对吸烟手势的检测准确率能达到较高水准，但实际应用中穿戴式设备成本过高，同时还受限电量、信号等因素的制约。除此之外，穿戴式设备还存在被攻击的安全风险，因此该种吸烟行为检测方式只适用于个体或特殊情况下使用，无法真正大规模推广，用于多种复杂场景下的吸烟行为检测。

1.2.2 图像分析检测方法

基于图像分析的深度学习检测方法的原理为通过吸烟手势图像数据训练出相应的深度学习模型，然后使用训练出的模型来分析图像采集设备采集到的图像，判断是否出现了吸烟手势，进行了吸烟行为。

Zhang 等[19]提出了一个名为 SmokingNet 的吸烟检测模型，该模型基于 GoogLeNet[20]进行改进，来对图像中的吸烟手势进行检测。先增强了利用非平方卷

积核对目标图像进行特征提取的能力，同时调整网络结构和使用特殊的卷积层，更好地提取吸烟图像的特征并提高模型对局部目标检测能力，提高了模型检测精度。最后在基于超大数据集进行预训练后对模型进行调整，以提高网络性能，最后模型准确率达到 90%，检测速度达到 80FPS，保证能够实时检测吸烟图像。

Wei 等[21]提出了一种改进的 YOLOv3 (You Only Look Once v3)[22]算法来检测吸烟手势。首先引入马赛克数据来增强丰富图像的背景，增加数据集的大小，避免过度拟合并降低对 GPU 的要求。其次用 Mish 激活函数、DIoU(Distance IoU)损失函数等对 YOLOv3 进行改进，Mish 激活函数具有良好的泛化能力和有效的结果优化能力，DIoU 损失函数在保留了传统 IoU 损失函数优点的同时，还大大加快了模型收敛速度。改进后的算法对比原算法准确率提升超过了 20%，检测精度达到 88.73%。

Zhang 等[23]结合动作识别和吸烟手势识别来检测吸烟行为。先使用 YOLO[24]目标检测算法结合卡尔曼滤波器跟踪人体，然后使用 Alphapose 的人体姿势估计来获取人体的关键点，再将关键点输入时空图卷积网络，以初步识别吸烟行为。同时为了排除饮酒、抓挠等行为的误判，在动作识别的基础上，第二次增加了 YOLO 目标检测算法，基于面部手势图案来提取吸烟的图像纹理，实现了对吸烟行为的联合确定，以提高吸烟行为识别的检测精度和鲁棒性，最后的模型在置信度阈值 0.8 下 F1 分数达到了 74.44%。

Hameed 等[25]使用深度学习来检测油田或加油站等危险区域中的吸烟行为。采用 Xethru X4M03 UWB RADAR 雷达传感器收集数据，并以光谱图的形式进行表示，最后使用 InceptionV3[26]、VGG19[27]和 VGG16 来提取光谱图中的时空信息，并进行分类是否属于吸烟手势和姿态。最后结果表明，InceptionV3 能达到 90%的最大精度。

王彦生等[28]使用改进后的 YOLOv5s 算法来检测吸烟手势，以此保障发电厂厂区内的生产安全。在 YOLOv5s 的基础上，使用多头自注意力层替换 C3 模块中的 3x3 卷积，来提高算法的学习能力。在网络中添加 ECA(Efficient Channel Attention)[29]注意力模块，来加强网络中特征图的通道特征;同时 SloU (Scylla Intersection over Union)[30]作为回归损失函数，并采用加权双向特征金字塔网络 (BiFPN, Bidirectional Feature Pyramid Network)[31]快速进行多尺度特征融合，进一步提高算法的检测精度。改进后算法吸烟行为的检测精度为 89.3%，能有效识别吸烟手势，检测出吸烟行为。

以上研究现状表明基于图像分析的深度学习吸烟手势检测方法虽然准确率略低于穿戴式设备，但使用该方法检测吸烟行为成本低、鲁棒性高，并能适应多种场景，能更好的用于部署各种检测系统。此外，上述研究还表明，深度学习中卷积神

经网络能更好的学习到图像的各种特征，实现较高的识别精度，同时还能根据需求，实现轻量级的网络，更方便的部署在边缘端的检测设备上。因此，本文采用图像分析的深度学习检测方法，并选用轻量级的卷积神经网络来进行吸烟手势的检测研究。

1.3 论文主要工作

本文的研究目的是在树莓派上实现吸烟手势检测系统，选取 YOLOv7-tiny 算法进行改进，使算法更轻量化，准确度更高，速度更快，使其能够在树莓派上高效运行，最终实现快速准确检测吸烟手势，识别出吸烟行为，以达到各种场合吸烟提醒警示作用。本文主要工作如下：

1.研究了深度学习目标检测算法的两种类型，使用速度快的单阶段检测算法作为研究对象，并选用当前最优秀的轻量级目标检测算法之一的 YOLOv7-tiny 算法来实现吸烟手势检测。

2.制作了吸烟手势数据集，使用 Python 爬虫和实际拍摄收集图像，并对得到的图像进行筛选、清洗和标注，最后共得到 2300 张图像的数据集。

3.对 YOLOv7-tiny 算法进行了改进，在保障精度的同时，使模型参数量、计算量和模型尺寸更小，提高了检测速度，并更易于在树莓派上部署。

4.改进得到的 YOLOv7-tl(YOLOv7-tiny-lite)算法，在树莓派上实现了对吸烟手势的快速、准确检测，最后介绍了吸烟手势检测系统的开发环境和检测流程，并对实现的系统效果进行了展示。

1.4 论文组织结构

本文由六个章节组成，其组织结构如下：

第一章介绍了本文的研究背景及意义，对国内外当前基于深度学习的吸烟手势检测的两种研究现状进行了说明，分别为穿戴式设备检测和图像分析检测两种方式，同时说明了两种方式的优缺点及本文选择的研究方式。最后对本文的主要工作和组织结构进行了概述。

第二章说明了卷积神经网络的发展历程，对卷积神经网络的组成进行了介绍，包括：卷积层、池化层、激活函数和激活函数。接下来对深度学习在目标检测领域中发展中产生的两大类型算法（单阶段和两阶段）进行了原理说明和优劣势对比，并对两阶段和单阶段检测算法的代表分别进行了概述，提出使用轻量级单阶段检测算法 YOLOv7-tiny 进行吸烟手势的检测，并详细介绍了其网络结构、组成模块、聚类算法和损失函数，最后对迁移学习的思想和概念进行了阐述。

第三章详细说明了选用的 YOLOv7-tiny 算法的改进和优化，为了使模型更加

轻量和精度更高，对该算法模型主要从网络结构、先验框聚类算法和回归损失函数等三个方面进行了改进，并阐释了每种改进所选用的原因以及带来的结果，改进后的模型命名为 YOLOv7-tl(YOLOv7-tiny-lite)。

第四章主要对本文的数据集和实验结论进行了分析。对进行实验的环境和评估指标进行了说明，采用准确率、召回率、平均准确率、平均准确率均值、参数量、计算量、模型大小和每秒帧速来衡量不同模型效果。并对实验所用数据集进行了说明，包括图片采集、标注和增强。同时对本文改进后得到的 YOLOv7-tl 算法进行了详细的实验，并对实验结果进行了分析说明。

第五章对实现手势检测系统进行了设计与实现。介绍了系统的硬件和软件开发环境，同时对系统的整体流程进行了详细介绍，并对系统整体的实现效果进行了测试。

第六章对本文的研究成果进行了总结和分析，并指出还存在哪些不足之处，最后对下一步的研究和改进的方向进行了展望。

第 2 章 相关技术介绍

2.1 卷积神经网络

2.1.1 卷积神经网络概述

神经网络这一概念最早由生物界提出，1962 年，Hubel 和 Wiesel 记录了猫脑中各个神经元的电活动，由 Hubel 早期发明的特殊记录电极实现，他们通过一些实验系统地创建了视觉皮层的 map。1980 年，日本科学家福岛邦彦从猫的视觉系统实验中得到启发，提出了 Neocognitron[32]，在这项工作中，首次使用卷积神经网络实现了模式识别，因此被认为是真正的卷积神经网络发明者。到 1989 年左右，LeCun 将反向传播应用到了类似 Neocoginitro 的网络上来做有监督学习，CNN 开始逐渐走向各个应用领域[33]。1998 年，LeCun 提出了 LeNet-5[34]，其网络结构对比现在的一般卷积神经网络基本没有区别，只是层数较浅，成为卷积神经网络发展的一个里程碑。直到 2012 年，AlexNet[35]被提出，在 ImageNet2012 的图片分类任务上，AlexNet 以 15.3%的错误率登顶，而且以高出第二名十几个百分点的差距击败所有其他参与者，它的出现标志着神经网络的复苏和深度学习的崛起。在这之后，卷积神经网络飞速发展，人工智能迈入新时代。

2.1.2 卷积神经网络组成

卷积神经网络由输入层、卷积层、激活函数、池化层、全连接层组成[36]。输入层为整个卷积神经的输入，一般将图像转换为像素矩阵，然后使用卷积层来提取相应特征，激活函数则对卷积操作提取的特征进行非线性映射。池化层则负责对感受域内的特征进行筛选，提取区域内最具代表性的特征，能够有效地降低输出特征尺度，进而减少模型所需要的参数量。全连接层则负责对卷积神经网络学习提取到的特征进行汇总，将多维的特征输入映射为二维的特征输出。

2.1.2.1 卷积层

卷积层是构成卷积神经网络结构的基础，一般由线性和非线性的运算构成，即卷积操作和激活函数[37]。在卷积神经网络模型中，由输入层将输入图像转换为矩阵后，就可以对图像进行卷积操作。卷积操作是利用不同的卷积核来对输入图像进行特征提取，卷积核又被称为滤波器，一般为行、列数相同且为奇数的矩阵，因此利用卷积核对图像进行特征提取的过程，本质上是两个矩阵的交互。不同的卷积核对输入图像的处理，能获取到不同的特征，如角、边、线等，同时产生图像锐化、模糊等效果。

卷积过程如图 2-1 所示。卷积核对输入矩阵进行从左至右、从上到下的互相关

运算，即对应位置先相乘再整体相加，卷积后生成的图像称为特征图(feature map)，生成的矩阵称为 feature map 矩阵。除此之外，特征图的大小还与卷积的填充(Padding)和步长(Stride)两个超参数有关。

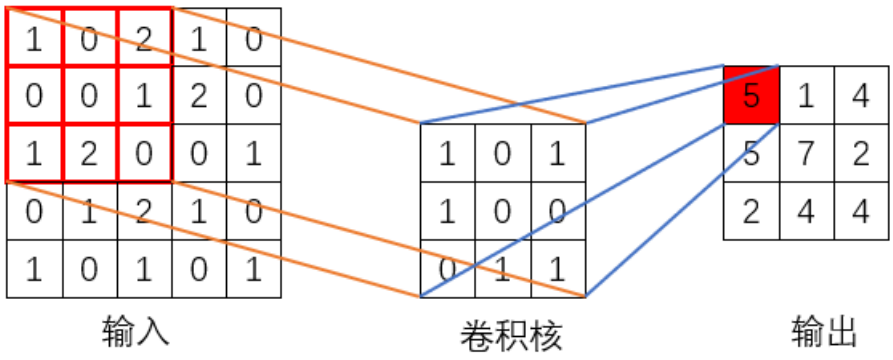

图 2-1 卷积过程

Figure 2-1 Convolution process

由图 2-1 可以看出，输入图像与卷积核进行卷积后损失了一部分，同时图像的中间部分经过多次卷积，可以提取到有效的特征，而边缘的特征则丢失掉了，这是卷积的互相关运算过程所导致的必然结果。并且有时还需要控制输入和输出的大小一致，便于进行后续操作。为了解决这些问题，可以在进行卷积操作时，对原矩阵进行填充，在高和宽的两侧填充元素，通常为 0，其原理如图 2-2 所示。

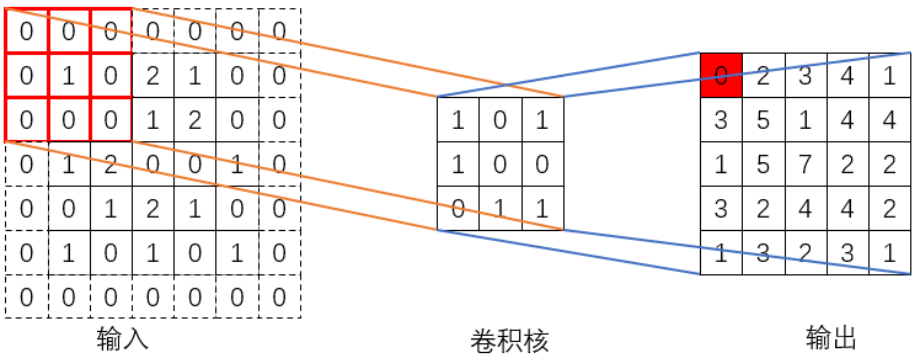

图 2-2 卷积过程（填充）

Figure 2-2 Convolution process(padding)

除填充外，步长也能控制特征图的大小，步长越小，能提取到的特征就越多，生成的特征图就越大。图 2-1 的卷积过程中，步长为 1，当步长变为 2 时，其生成的特征图也会发生变化，如图 2-3 所示。

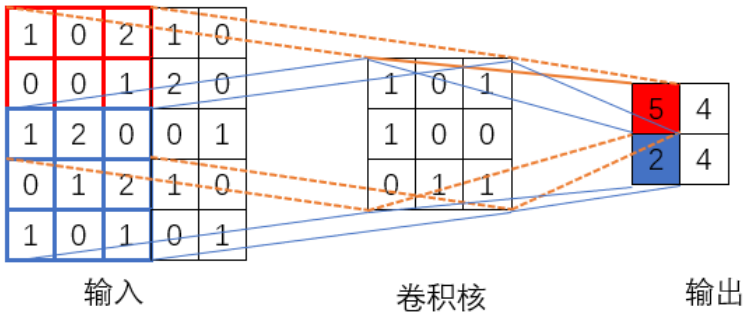

图 2-3 卷积过程（步长为 2）

Figure 2-3 Convolution process(padding is 2)

由此可知，当输入图像尺寸为 $k\times k$，卷积核大小为 $h\times h$，步长为s，填充为 p，特征图尺寸为 o 时，卷积计算公式如公式 2-1 所示：

$$o = \left\lfloor \frac{k + 2p - h}{s} \right\rfloor + 1 \qquad (2\text{-}1)$$

其中$\lfloor\ \rfloor$为向下取整，当结果不是整数时进行向下取整。

上述为单通道卷积过程，而实际上大多数输入图像均为多通道，多通道卷积与单通道卷积原理一致，此时卷积核通道数与输入图像通道数相同，卷积时对应的通道进行互相关运算，得到的结果相加，最终形成单通道输出，其原理如图 2-4 所示。

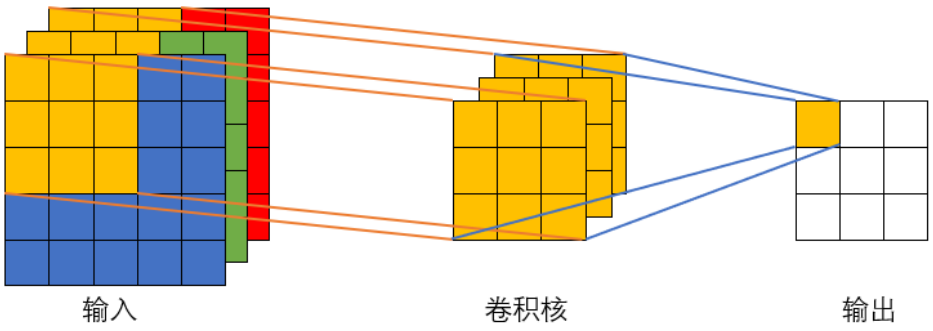

图 2-4 多通道卷积过程

Figure 2-4 Multi-channel convolution process

2.1.2.2 激活函数

激活函数(Activation Function)，指在人工神经网络的神经元上运行的函数，一般为非线性函数，其对输入信息进行非线性变换，然后将变换后的输出信息作为输入信息传给下一层神经元。在卷积神经网络中，常与卷积层、池化层、残差结构等连用，引入非线性因素，使得神经网络可以更好地解决较为复杂的问题。常用的激

活函数有：Sigmoid、Tanh、ReLU、LeakyReLU 等。

Sigmoid 函数，广泛使用的激活函数之一，表达式为$f(x)$ =1/(1+e^{-x})，函数图像如图 2-5 所示，其计算方式为指数运算，将输入映射到(0,1)区间，可以用于输入的归一化，防止输出值跳跃。缺点为在神经网络反向传播时，可能出现梯度消失或梯度爆炸的情况，并且计算代价大。

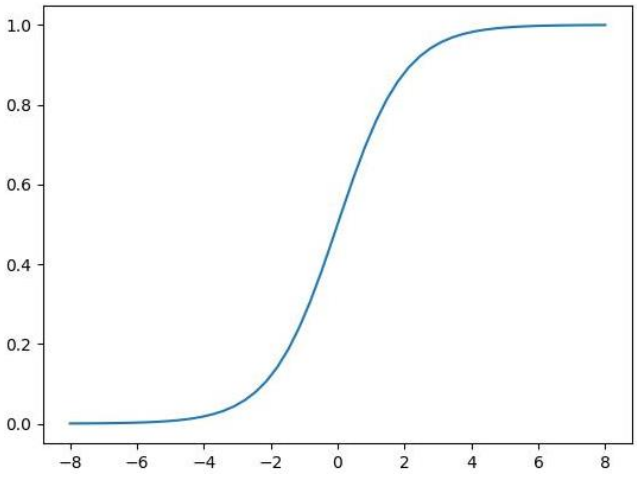

图 2-5 Sigmoid 激活函数

Figure 2-5 Sigmoid activation function

Tanh 函数，表达式为$f(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$，函数图像如图 2-6 所示，为双曲正切函数，优点与 Sigmoid 函数类似，输出映射区间变化为(-1,1)，缺点也为会出现梯度消失或梯度爆炸现象，计算代价大。

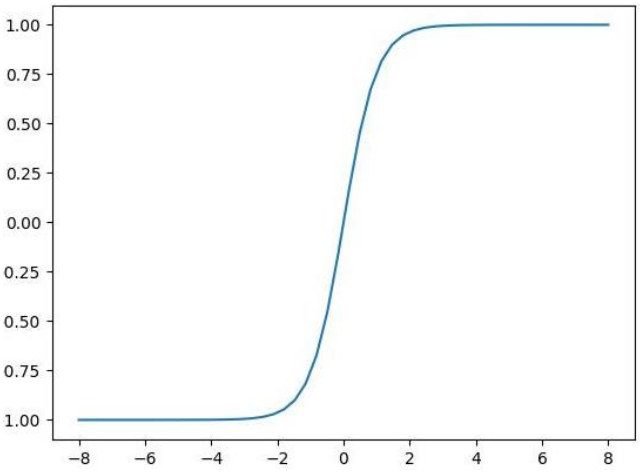

图 2-6 Tanh 激活函数

Figure 2-6 Tanh activation function

ReLU 函数，是目前使用最多的激活函数，表达式为$f(x) = x, x > 0$，函数图像如图 2-7 所示，其计算方式简单、效率高，能够使网络快速收敛，同时在 x>0 时可以保持梯度不衰减，从而缓解了梯度消失问题，但是当输入接近零或为负时，函数的梯度变为零，网络无法进行反向传播，也无法学习。

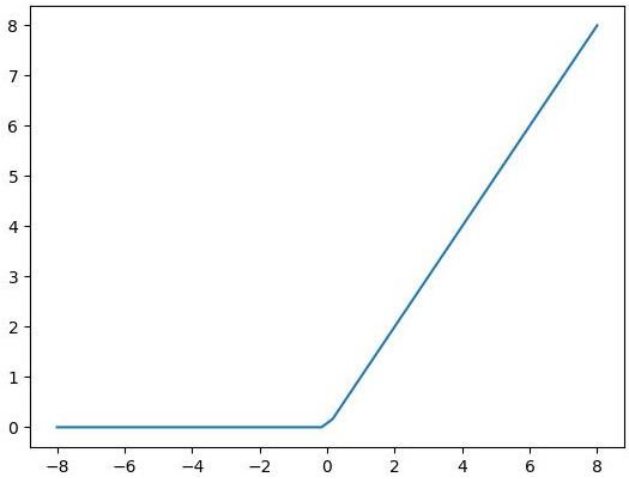

图 2-7 ReLU 激活函数

Figure 2-7 ReLU activation function

LeakyReLU 函数，是 ReLU 函数的一种变形，为了解决 ReLU 函数在 $x \leqslant 0$ 时的梯度为零的问题。当 x>0 时，表达式不变，当 $x \leqslant 0$ 时，引入一个超参数，其表达式为$f(x) = \alpha x$，一般设置为 0.01，如图 2-8 所示。在反向传播过程中，对输入小于零的情况，也可以计算得到一个梯度，避免了梯度消失的问题。

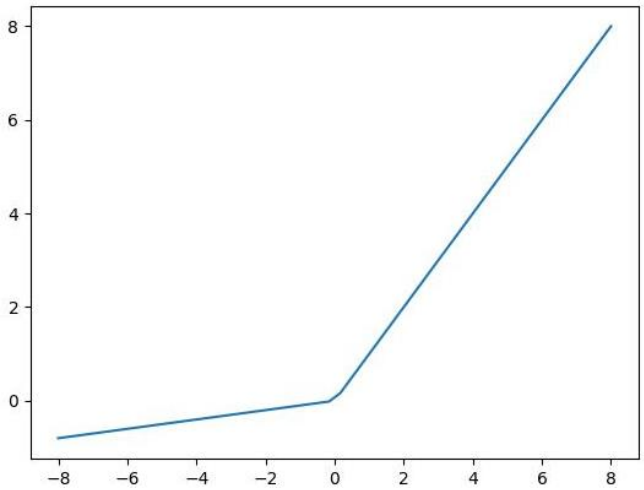

图 2-8 LeakyReLU 激活函数

Figure 2-8 LeakyReLU activation function

除上述激活函数外，还有许多其他的激活函数，如：ELU、SeLU、Swish 等，它们各自具有不同的优缺点，能够满足不同用途的需求。

2.1.2.3 池化层

池化层一般位于卷积层之间，池化是一种降采样(down-sampling)技术，目的是为了缩小特征图尺寸，减少网络模型参数，并提取特征图主要特征。对于输入的特征图，将其进行分割，然后对于分割的子窗口，采取某种策略取一个特征值。池化会不断地减小数据的空间大小，因此参数的数量和计算量也会下降，这在一定程度上也控制了模型过拟合。

池化操作根据策略不同可分为许多不同种类，应用较多的为最大池化和平均池化。最大池化取卷积核覆盖区域的最大值作为特征值，如图 2-9 所示。同理，平均池化取卷积核覆盖区域的平均值作为特征值，其他与最大池化保持一致。在进行特征提取的过程中，平均池化可以减少邻域大小受限造成的估计值方差，能更多保留图像背景信息；而最大值池化能减少卷积层参数误差造成估计均值误差的偏移，能更多的保留纹理信息。

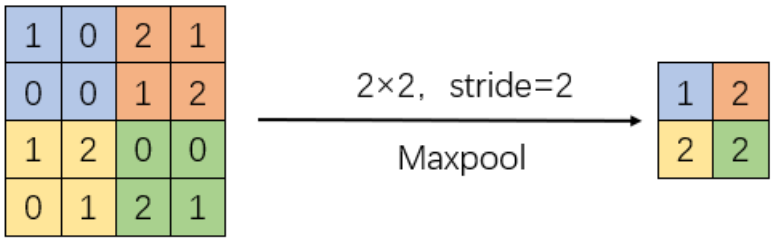

图 2-9 最大池化

Figure 2-9 Maxpool

2.1.2.4 全连接层

在卷积神经网络中，经过多个卷积或池化层之后，通常连接一个或一个以上的全连接层，全连接层中的每个神经元与其前一层的所有神经元进行全连接[38]。全连接层会将卷积层提取和池化层采样得到的特征进行整合并映射到网络的最终输出端，一般为具有类别区分性的局部信息，最终的全连接层输出点个数一般与样本类数相同[39]。

2.2 目标检测算法

传统的目标检测算法需要先对图像进行区域划分，再从图像中提取特征，特征还需手工进行设计，然后使用训练好的分类器进行分类并使用非极大值抑制进行筛选得到最终结果，整个过程需要耗费大量的时间和资源。随着近年来深度学习的

快速发展，其也被应用到目标检测领域中，使用卷积神经网络代替传统方式进行特征提取和分类，解决了传统检测方式的弊端。而随着深度学习目标检测算法被应用到各种检测任务中，各式各样的检测算法被提出，算法的性能也越来越好。总体上，这些目标检测算法大致可分为两类：两阶段(two-stage)检测算法和单阶段(one-stage)检测算法。

两阶段含义为先从输入图像中提取候选区域，再进行目标分类和坐标位置回归，代表算法如：R-CNN[40] 、SPP-Net (Spatial Pyramid Pooling)[41]、Fast R-CNN[42]、Faster R-CNN[43]、R-FCN[44]等。单阶段含义为直接从输入图像回归目标的坐标位置与类别分类信息，不再划分候选区域。代表算法如：YOLO、YOLO9000[45]、YOLOv3、SSD (Single Shot MultiBox Detector)[46]、Retina-Net[47]等。检测过程的不同，也导致它们的优缺点不同。相较而言，两阶段检测算法的精确度高，但检测速度慢。单阶段检测算法检测速度较快，但精确度上略逊于两阶段检测模型。

2.2.1 两阶段检测算法概述

区域卷积神经网络(R-CNN)是两阶段检测算法的起始之作，也是第一个将深度学习用于目标检测的算法，在它的基础上，衍生出了一系列优秀的两阶段目标检测算法，本小节选取经典的两阶段算法 R-CNN、SPP-Net、Fast R-CNN、Faster R-CNN 进行概述。

2.2.1.1 R-CNN 算法

R-CNN 算法由 Ross Girshick 等人于 2014 年发表，在此之前，传统的目标检测方法通常在图像识别的基础上进行物体检测，先使用穷举法选出所有物体可能出现的区域框，然后进行特征提取和图像识别分类，然后采用非极大值抑制(Non-maximum Suppression,NMS)对分类成功的区域进行筛选，得到最终输出。

R-CNN 在传统的方法的思路上，进行了改进。检测思路仍然使用候选区域选取，提取候选区域特征、分类与回归和非极大值抑制的方法发生了改变。R-CNN 使用选择性搜索算法来选取候选区域，先将所有分割区域的框列入到候选区域列表中，然后基于相似度合并区域，直到得到最终的候选区域，最后每个图像得到 2000 个候选区域。然后对得到的候选区域进行尺寸调整并使用卷积神经网络进行特征提取，特征映射被卷积和汇聚并输出。接下来使用 SVM 分类器对特征输出进行分类并进行边界框回归，最后使用非极大值抑制对多余的框进行去除，得到最后的检测结果，如图 2-10 所示。相较于传统检测算法，R-CNN 的检测精度大大提升，但仍存在许多问题，包括：效率低、时间长、占用空间大等，但它的出现为后续的算法奠定了基础。

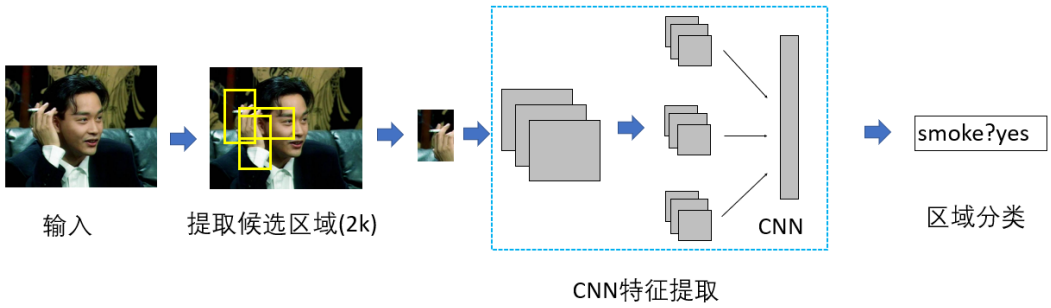

图 2-10 R-CNN 算法检测步骤

Figure 2-10 R-CNN algorithm detection steps

2.2.1.2 SPP-Net 算法

空间池化金字塔网络(SPP-Net)算法由何恺明等人于 2015 年发表。SPP-Net 的出现是为了解决 R-CNN 算法速度慢的问题，其基本原理与 R-CNN 类似，但在特征提取时，不再对得到的候选区域进行特征提取，改为直接在原图上提取特征，大大加快了特征提取的速度。此外，由于神经网络中全连接层的神经元个数是固定的，导致神经网络的输入尺寸也受到了限制，对输入的不同尺寸的图片进行调整时，无论是裁剪还是缩放，都会导致一定程度的信息丢失或变形，进而对模型的精度产生一定影响。

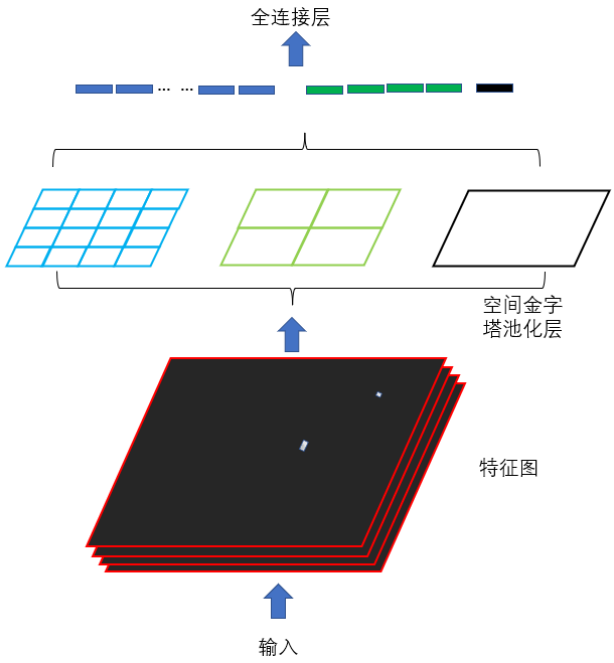

图 2-11 空间金字塔池化过程

Figure 2-11 Spatial Pyramid Pooling steps

空间池化金字塔(SPP)结构则有效解决了上述问题，在最后的卷积层之后、全连接层之前加入 SPP 层，SPP 层可以接受不同尺寸的图像，并输出固定尺寸的特征向量，再将特征向量输入至全连接层，这样就让全连接层也适应了不同尺寸的输入，模型最终的精度也得到了提升。空间金字塔池化过程如图 2-11 所示。

尽管 SPP-Net 在 R-CNN 的基础上解决了部分问题，提升了一定的速度和精度，但其思想和检测原理仍采用 R-CNN 的方式，使用选择性搜索算法选取候选区域及采用 SVM 算法进行分类等，因此仍然存在效率低、消耗内存大等问题。

2.2.1.3 Fast R-CNN 算法

Fast R-CNN 由 Girshick 等人于 2015 年发表。Fast R-CNN 在 R-CNN 的基础上进行了一些改进，使算法速度更快、精度更高。Fast R-CNN 继续使用选择性搜索算法筛选出 2000 个候选区域，但为了提高效率，减少重复运算，只对输入图像整体进行卷积运算提取特征。然后，Fast R-CNN 使用感兴趣区域(Region Of Interest，ROI)池化层统一特征图输出尺度，再输入到全连接层。其原理为对每个候选框，均映射到最后卷积得到的特征图上，获取相应的感兴趣区域，然后对每个感兴趣区域划分固定尺寸的均匀网格，然后对每个单元格使用最大值池化，原特征图就被映射成为了固定尺寸的新特征图。此外，Fast R-CNN 使用深度神经网络来进行分类和回归操作，使用 softmax 代替 SVM 分类器并将 bounding box 线性回归器放入网络同时训练，有效的提高了模型训练速度同时节省了训练 SVM 分类器和回归器所需的存储空间。Fast RCNN 算法结构如图 2-12 所示。

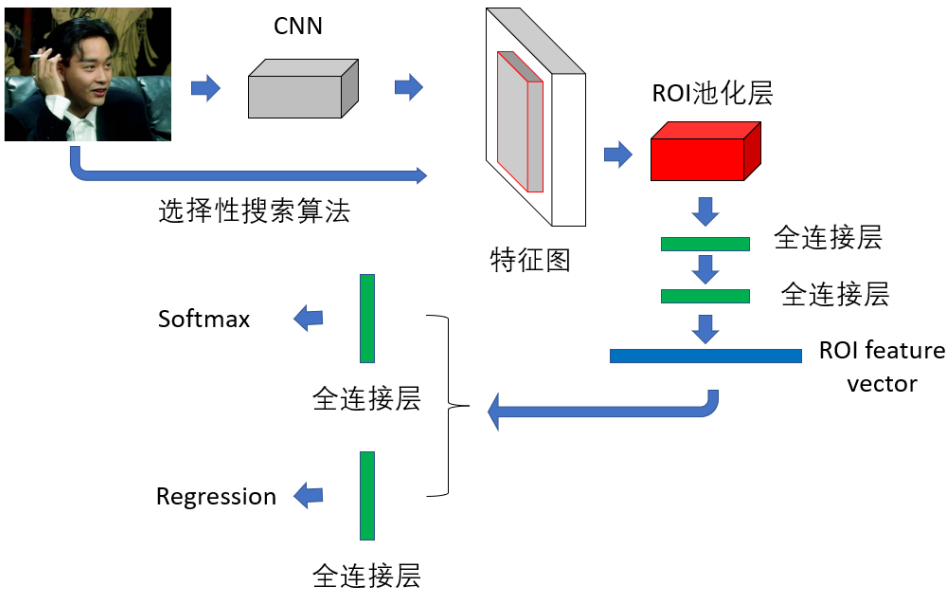

图 2-12 Fast R-CNN 算法结构图

Figure 2-12 The structure of Fast R-CNN algorithm

2.2.1.4 Faster R-CNN 算法

Faster R-CNN 仍然由 Girshick 等人于 2015 年发表。Faster R-CNN 在 Fast R-CNN 的基础上更进一步，由四个部分组成：特征提取网络、区域候选网络(Region Proposal Network,RPN)、ROI 池化和分类与回归。

Faster R-CNN 首先将输入图像调整到固定大小，然后输入到特征网络中。特征提取网络由 13 个卷积层、13 个 ReLU 层和 4 个池化层组成，对输入图像的特征进行提取，并将特征图输入到后续的 RPN 网络中。RPN 网络则代替了选择性搜索算法，用于生成候选框。RPN 采用了先验框(anchor)机制，对输入的特征图每一个点都生成 9 个 anchor 作为初始的检测框，包含 3 种尺度和 3 种宽高比，基本能覆盖各个位置上不同大小的目标。然后使用 softmax 分类器提取出 positive anchors，即包含物体位置的 anchor，并使用 bounding box 回归器对 positive anchors 进行回归，生成最终的候选区域，输入到 ROI 池化层中。ROI 池化层则与 Fast R-CNN 中的功能保持一致，对输入的特征图进行尺寸统一，再输入到全连接层中。最后使用深度神经网络进行分类与回归实现预测，其结构如图 2-13 所示。

Faster R-CNN 算法创造性的使用 RPN 代替了选择性搜索算法生成候选框，将算法所有步骤全部使用神经网络来完成，极大的提高了网络性能，在精度和速度上都取得了较大提升，成为了两阶段检测算法的典型代表。

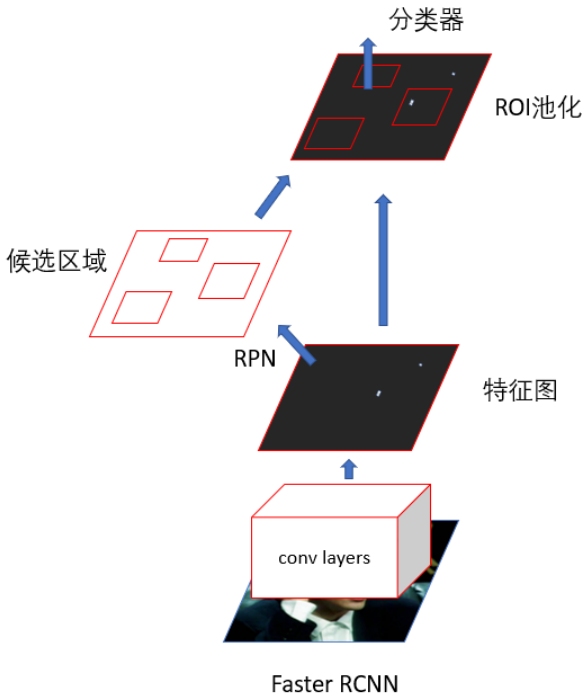

图 2-13 Faster R-CNN 算法结构图

Figure 2-13 The structure of Faster R-CNN algorithm

2.2.2 单阶段检测算法概述

作为单阶段检测算法的典型代表，YOLO 算法系列一直是各个时期最具有竞争力的目标检测算法，其中 2016 年发表的 YOLOv1 算法更是单阶段检测算法的开山之作。经过长时间的发展，YOLO 系列衍生出了各种各样的版本，本小节选取经典的单阶段检测算法 YOLO 系列的部分版本进行概述。

2.2.2.1 YOLOv1 算法

YOLOv1 算法由 Joseph Redmon 等人在 2016 年发表，创新型的提出了通过直接回归的方式来获取目标的具体信息和类别分类信息，不再使用提取候选区域这一步，打破了当时 Faster R-CNN 等两阶段目标检测算法为主流的局面。虽然单阶段检测方式造成了一定的检测精度损失，但极大的降低了模型计算量，并显著的提升了检测速度，被广泛的应用在实际工程中。

YOLOv1 的网络结构使用 CNN 神经网络来设计，包含 24 个卷积层和 2 个全连接层，卷积层用于提取图像的特征，全连接层用于预测输出的坐标和概率，YOLOv1 网络结构如图 2-14 所示。

类型	通道数	Size/Stride	输出特征图大小	
Convolutional	64	7×7/2	224×224	
Maxpool		2×2/2	112×112	
Convolutional	192	3×3	112×112	
Maxpool		2×2/2	56×56	
Convolutional	128	1×1	56×56	
Convolutional	256	3×3	56×56	
Convolutional	256	1×1	56×56	
Convolutional	512	3×3	56×56	
Maxpool		2×2/2	28×28	
Convolutional	256	1×1	28×28	×4
Convolutional	512	3×3	28×28	
Convolutional	512	1×1	28×28	
Convolutional	1024	3×3	28×28	
Maxpool		2×2/2	14×14	
Convolutional	512	1×1	14×14	
Convolutional	1024	3×3	14×14	
Convolutional	512	1×1	14×14	
Convolutional	1024	3×3	14×14	
Convolutional	1024	3×3/2	7×7	
Convolutional	1024	3×3	7×7	
Convolutional	1024	3×3	7×7	
Connected				
Connected			7×7	

图 2-14 YOLOv1 算法结构图

Figure 2-14 The structure of YOLOv1 algorithm

其检测原理为首先将输入图像调整为 448×448×3，然后将图像划分为 7×7 的网格，检测目标的中心点所落的网格负责检测该物体，每个网格又预测 2 个边界框(bounding box)，每个边界框包含五个元素：边界框中心点坐标的坐标、边界框长宽及边界框置信度，同时每个网格还预测 c 个条件类别概率，为检测到的目标属于特定类别的概率，网络最终输出一个 7×7×30 的张量。

2.2.2.2 YOLOv2 算法

YOLOv1 虽然检测速度快，但在定位方面表现不佳，并且召回率较低，对小物体检测效果差。因此 Joseph Redmon 等人在 2017 年发表了 YOLOv2 算法。相对于 YOLOv1 算法，在保持其检测速度的基础上，进行了大量改进，挺高了模型的精确度、召回率和定位能力。

在网络结构上，YOLOv2 算法骨干网络为 Darknet19，其结构如图 2-15 所示，主要采用 3×3 卷积层和 2×2 最大池化层构建。并采用多尺度训练策略，在训练时每隔几次迭代就会调整网络的输入尺寸，由于 Darknet19 下采样总步长为 32，因此输入图片尺寸一般为 32 的倍数。多尺度训练策略可以更灵活的满足不同需求，当输入图片分辨率低时，速度更快，精度更低，输入图片分辨率更高时，则相反。

类型	通道数	Size/Stride	输出特征图大小
Convolutional	32	3×3	416×416
Maxpool		2×2/2	112×112
Convolutional	64	3×3	112×112
Maxpool		2×2/2	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		2×2/2	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		2×2/2	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		2×2/2	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

图 2-15 Darknet19 结构图

Figure 2-15 The structure of Darknet19

此外，在 YOLOv1 基础上，YOLOv2 算法在每个卷积层后添加了批归一化层(Batch Normalization，BN),去掉了 dropout 层，能有效提升模型的收敛速度，同时能起到一定的正则化效果，防止模型过拟合。YOLOv2 还引入了先验框来替代全连接层进行边界框的预测，避免了全连接层预测边界框会导致丢失空间信息、定位不准的问题，并且使用 K-means 聚类算法替代手动设定来生成先验框，适合尺寸的先验框使得网络学习特征更容易，收敛更快，同时有效的提高了模型的召回率。为了更好的检测小目标，YOLOv2 还提出了一个 passthrough 层将高分辨率的特征图与低分辨率的特征图连接到一起，使模型具有了细粒度特征，能适应不同尺寸大小的目标。相较于 YOLOv1，YOLOv2 在各方面都取得了长足进步。

2.2.2.3 YOLOv3 算法

2018 年，Joseph Redmon 等人发表了 YOLOv3 算法，在 YOLOv2 的基础上，融合了当时许多优秀算法的思想。

YOLOv3 首先引入了残差网络模块[48]来构建新的骨干网络，残差结构可以缓解训练时梯度消失的问题，并使得模型更容易收敛。新的骨干网络 Darknet53 结构如图 2-16 所示。使用多个残差块进行堆叠，加深了网络深度，能更好的提取到图像特征。

类型	通道数	Size/Stride	输出特征图大小	
Convolutional	32	3×3	256×256	
Convolutional	64	3×3/2	128×128	
Convolutional	32	1×1		
Convolutional	64	3×3		×1
Residual			128×128	
Convolutional	128	3×3/2	64×64	
Convolutional	64	1×1		
Convolutional	128	3×3		×2
Residual			64×64	
Convolutional	256	3×3/2	32×32	
Convolutional	128	1×1		
Convolutional	256	3×3		×8
Residual			32×32	
Convolutional	512	3×3/2	16× 16	
Convolutional	256	1×1		
Convolutional	512	3×3		×8
Residual			16×16	
Convolutional	1024	3×3/2	8× 8	
Convolutional	512	1×1		
Convolutional	1024	3×3		×4
Residual			8×8	
Avgpool		Global		
Connected		1000		
Softmax				

图 2-16 Darknet53 结构图

Figure 2-16 The structure of Darknet53

此外，YOLOv3 算法的还借鉴了特征金字塔网络的思想，融合了网络中浅层和深层语义信息，在底、中、高三个层次上分别预测目标框，最后输出三个尺度(52×52、26×26、13×13)的特征图信息[49]。52×52 尺度的特征图负责检测小型物体， 26×26 尺度的特征图负责检测中型物体，13×13 尺度的特征图负责检测大型物体。先验框的生成方式仍然使用 K-means 聚类算法，但数量增加至 9 个，划分为三组，分别匹配三种不同尺度的输出。此外，YOLOv3 取消了池化层，通过改变卷积核的步长来调整特征图的尺寸，最后使用多个 logistic 分类器替代 Softmax 来对每个框进行分类，在保证准确率的同时支持多标签分类。

对比 YOLOv2 算法，YOLOv3 基本解决了小目标的检测问题，在精度和速度方面都进行了提升，并实现了一个较好的平衡。

2.2.2.4 YOLOv4 算法

YOLOv4[50]算法由 Alexey Bochkovskiy 等人于 2020 年 4 月发表，在 YOLOv3 的基础上，进行了适当的改进，并采用了当时许多优秀算法和网络的结构，从而在速度和精度上都得到了一定的提升。

YOLOv4 的骨干网络采用的是 CSPDarknet53，借鉴了 CSPNet (Cross Stage Partial Network)[51]的思想，将 CSP 结构融入到网络中，其原理为将输入的特征图按照维度拆分为两部分，一份经过残差结构继续提取特征，另一份使用拼接操作(concat)到最后的输出，这种结构可以在保持性能不变的情况下，减少模型的参数量和计算量，使网络能够更加高效的学习到特征信息。同时，YOLOv4 算法将原 DarkNet53 的卷积层的激活函数 LeakyReLU 全部替换为了 Mish 激活函数，其表达式为$f(x) = xtanh(\ln(1 + e^x))$，使信息能更好的深入网络，得到更好的准确性和泛化，使网络能提取到更好的特征。此外，YOLOv4 采用 SPP(Spatial Pyramid Pooling)结构来增加感受野，其流程为分别使用 1×1、5×5、9×9、13×13 大小的池化层来处理输入的特征图，最后进行拼接，大的感受野具有更多的信息，使网络能够学习到更多的特征。CSP 结构如图 2-17 左边所示，SPP 模块结构如图 2-17 右边所示。

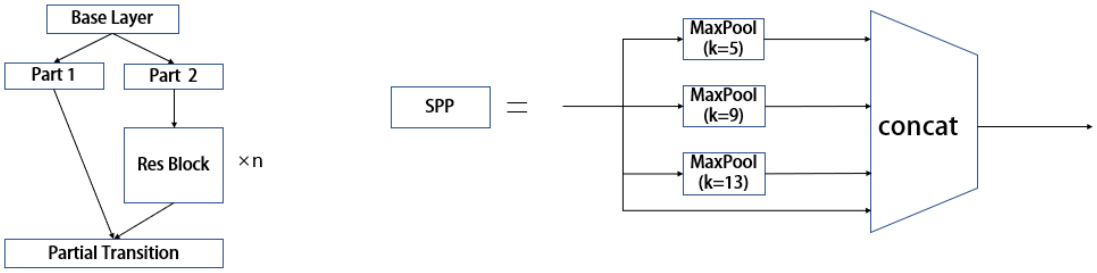

图 2-17 CSP 和 SPP 模块结构图

Figure 2-17 The structure of CSP and SPP module

在网络的特征融合部分，YOLOv4 采用路径聚合网络(Path Aggregation Network,PANet)结构对特征进一步融合，在网络中将特征在深层和浅层融合了两次，达到更强的特征聚合效果，显著的提升了模型的性能。YOLOv4 网络结构如图 2-18 所示。

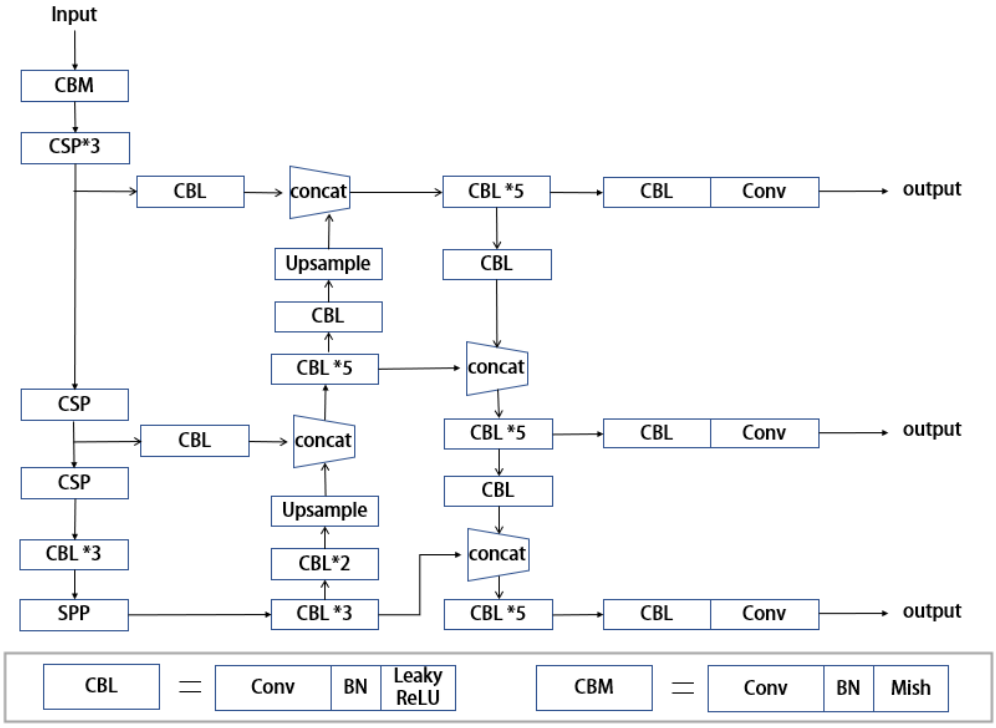

图 2-18 YOLOv4 算法结构图

Figure 2-18 The structure of YOLOv4 algorithm

2.2.2.5 YOLOv7 算法

YOLOv7[52]在 2022 年 7 月由 Alexey Bochkovskiy 等人发表。YOLOv7 整体框架与前作相近，但同样推出了更快更强的网络结构，并融合了许多当前优秀的框架和算法思想，是当前目标检测领域检测速度最快、精度最高的物体检测器之一。

YOLOv7 主要使用 CBS 模块、扩展高效聚合网络 (Extended Efficient Layer Aggregation Networks,E-ELAN)模块和 DownC 模块来构建网络的骨干部分。CBS 模块由卷积 (Conv)、批归一化和激活函数 SiLU 组成，用来提取图像的特征。E-ELAN 是一个高效的网络结构，由七个 CBS 模块和一个 concat 模块组成构成，该结构对输入基数(Cardinality)做了扩展(Expand)、乱序(Shuffle)、合并(Merge)，能够在不破坏原始梯度路径的情况下，提升网络的学习能力，从而提高网络的准确率，根据分支的不同，将所形成的的两种模块命名为 E-ELAN 和 E-ELAN-H。结构如图 2-19 所示。

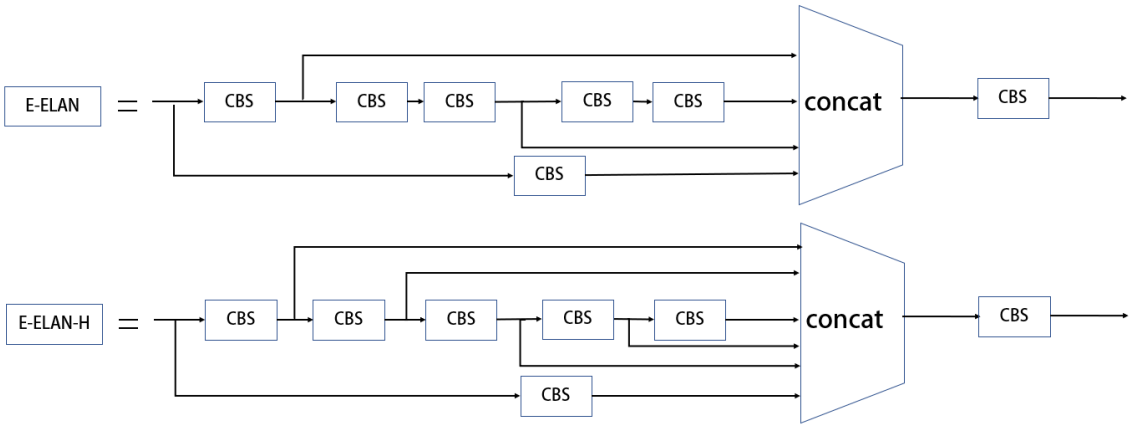

图 2-19 E-ELAN 模块和 E-ELAN-H 模块结构图

Figure 2-19 The structure of E-ELAN and E-ELAN-H module

YOLOv7 构建了一个 SPPCSPC 模块，该模块由 CSP 结构和 SPP 结构组合而成，共有 7 个 CBS 模块、3 个最大池化模块和 2 个 concat 模块。CSP 结构部分将输入特征图分为两个部分，一个部分进行常规卷积，另一个部分使用 SPP 结构进行处理，最后将得到的结果进行拼接，最后形成的 SPPCSPC 模块既能够扩大感受野，学习到更多的特征，提升模型精度，又减少了计算量，提升了速度。DownC 模块则包含两个分支，第一个分支使用最大池化层使得特征图大小减半并使用 1×1 卷积改变通道数，第二个分支先使用 1×1 卷积改变通道数，再使用步长为 2 的 3×3 卷积减半特征图大小，最后将两个分支的输出进行拼接。得到的最终输出通道数不变且特征图大小减半。DownC 模块和 SPPCSPC 模块结构如图 2-20 所示。

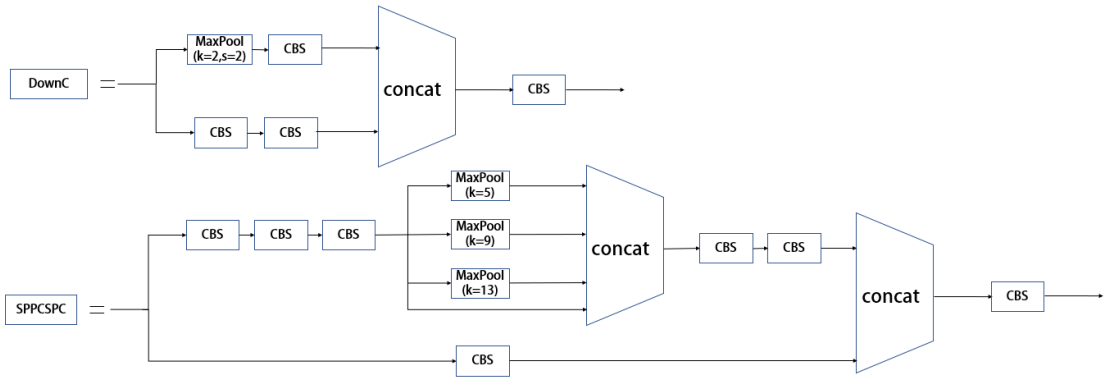

图 2-20 DownC 模块和 SPPCSPC 模块结构图

Figure 2-20 The structure of DownC and SPPCSPC module

最后将 RepVGG 的卷积重参数化的思想引入到网络架构中，即网络中的 Rep 模块，结构如图 2-21 所示。其思想为训练时采用多分支结构，可以更好的学习图

像特征，在进行推理时多分支结构可以等效融合为单个卷积的单分支结构，实现更低的推理延迟。

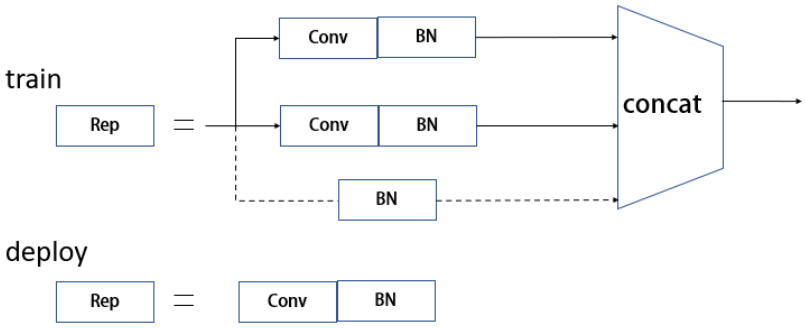

图 2-21 Rep 模块结构图

Figure 2-21 The structure of Rep module

YOLOv7 整体网络结构如图 2-22 所示。此外，YOLOv7 仍然使用 Anchor-based 的检测器，使用 K-means 聚类算法来生成先验框。

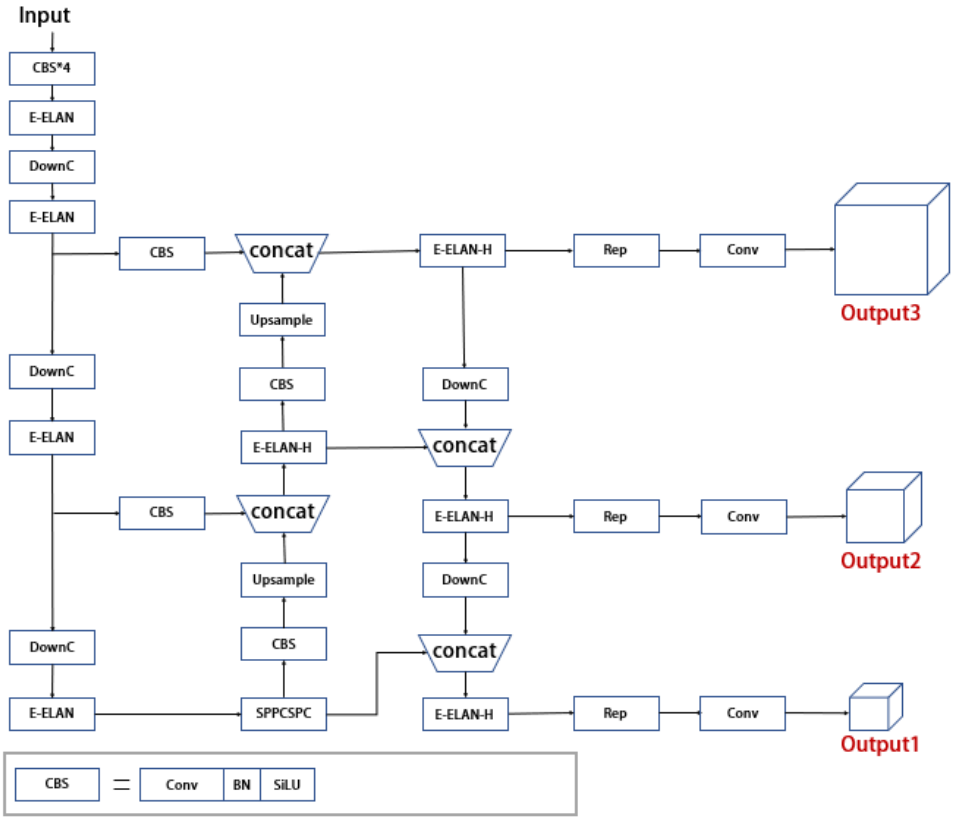

图 2-22 YOLOv7 算法结构图

Figure 2-22 The structure of YOLOv7 algorithm

2.3 YOLOv7-tiny 算法介绍

YOLOv7-tiny 是在 YOLOv7 的基础上简化而来。对比 YOLOv7，YOLOv7-tiny 缩减了网络层数，网络减少为 77 层，使网络变得更简洁，减少参数量和计算量。还对 E-ELAN、SPPCSPC 等模块进行了简化，简化后的模块命名为 ME-ELAN(Mini-E-ELAN)和 MSPPCSPC(Mini-SPPCSPC)，并使用 MP 模块代替 DownC 模块控制特征图大小减半，同时在最终输出之前采用三个 1×1 卷积代替 RepConv，在整个网络中使用 LeakyReLU 作为激活函数，使得模型更加轻量化，最终输出则保持三尺度不变。

2.3.1 YOLOv7-tiny 网络结构

YOLOv7-tiny 网络首先对输入的图像进行处理，调整至统一大小，然后将图像输入骨干网络。骨干网络由若干 CBL 模块、ME-ELAN 模块和 MP 模块交替组成。CBL 模块由卷积(Conv)、批归一化和激活函数 LeakyReLU 组成，用来提取图像的特征。ME-ELAN 模块在 YOLOv7 中扩展高效聚合网络(E-ELAN)模块的基础上减少了组成模块，改为由五个 CBL 模块和一个连接(concat)模块组成，在保持扩展、乱序、合并等基本操作不变的同时尽可能的减少计算量和参数量，使模块更轻量化。MP 模块为一个大小和步长都为 2 的最大池化模块，来使输入的特征图大小减半。

网络头部由 MSPPCSPC 模块、若干 CBL 模块、上采样(Upsample)模块、concat 模块和三个卷积模块组成。MSPPCSPC 模块在 YOLOv7 中 SPPCSPC 模块的基础上进行简化，由 4 个 CBL 模块，三个大小(kernel=5,9,13)不同的 Maxpool 模块和两个 concat 模块组成，MSPPCSPC 模块在保持特征图大小不变的同时使通道数减半，并获得多尺度特征信息。网络最后输出三个尺度的特征图。YOLOv7-tiny 各模块结构及网络结构如图 2-23、2-24 所示。

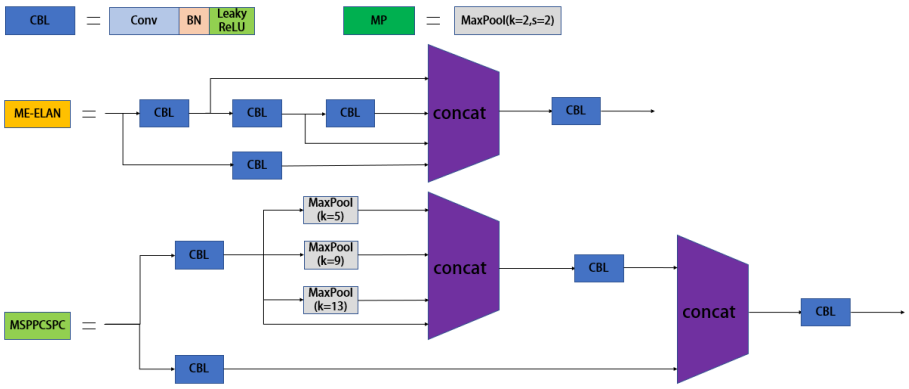

图 2-23 CBL、MP、ME-ELAN、MSPPCSPC 模块结构图

Figure 2-23 The structure of CBL, MP, ME-ELAN and MSPPCSPC module

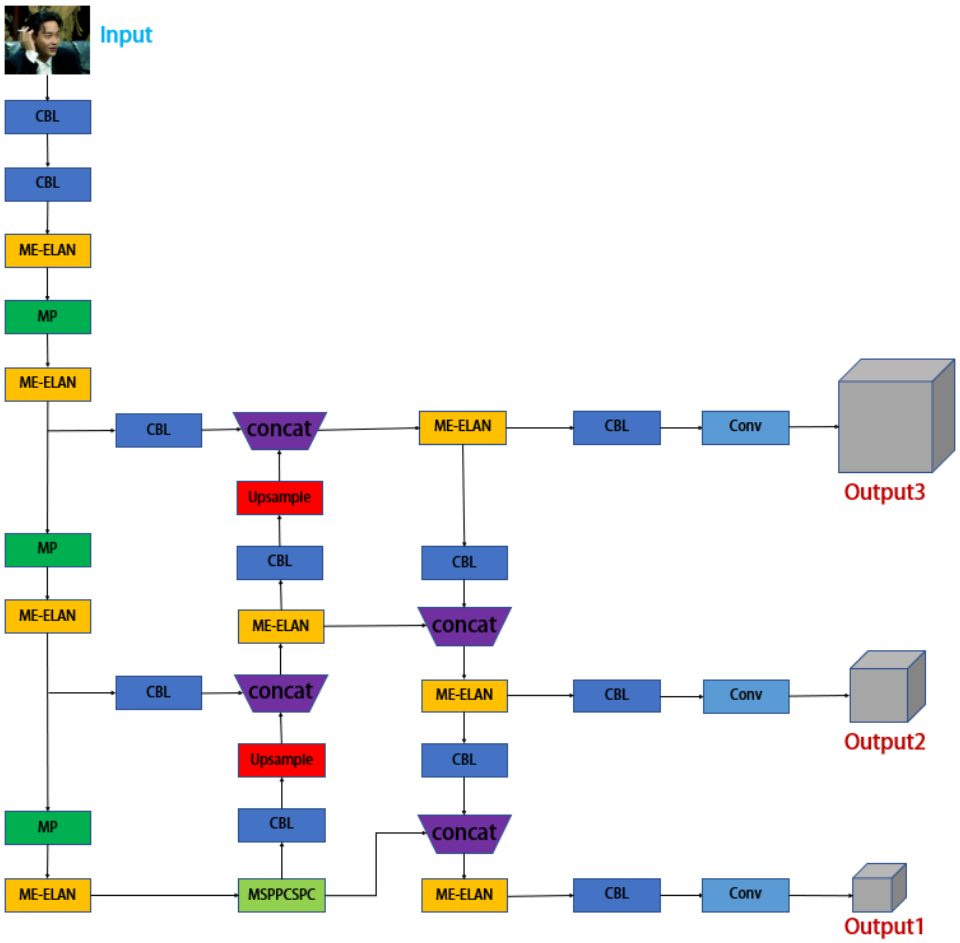

图 2-24 YOLOv7-tiny 网络结构图

Figure 2-24 The structure of YOLOv7-tiny algorithm

2.3.2 YOLOv7-tiny 先验框聚类算法

在目标检测领域中，部分算法在训练时会使用先验框，先验框的设置能有利于网络更好地学习输入图像的特征。YOLOv7-tiny 作为 YOLOv7 的轻量级版本，在模型训练时同样使用先验框，其流程为先初始化一定数量的先验框，并在后续的训练过程中不断的调整先验框，使之拟合真实框，因此不同先验框的选取会对预测结果产生一定的影响。YOLOv7-tiny 仍然使用 K-means 聚类算法对数据集的标注框进行聚类，得到先验框，算法原理为：

1.在样本空间中，随机选择 k 个初始化的聚类中心（与先验框个数保持一致）。

2.计算每一个样本到聚类中心的欧式距离，计算方式如公式 2-2 所示，并将样本点归到欧式距离最小，即相似度最高的簇中。

$$d(X, C_j) = \sqrt{\sum_{i=1}^{n} (X_i - C_{ij})^2} \quad (2-2)$$

X 表示样本中的数值点，C_j 表示第 j 个聚类中心，n 表示样本维度。

3.所有样本点分配后，计算每个簇的平均值，将所得的点为新的簇中心，重复 1、2 步，直至收敛。

当目标框尺寸较大时，采用标准 K-means 聚类时会产生较大的误差，因此引入了 IoU 值替换欧式距离，来避免这个问题，如公式 2-3 所示：

$$d(a, b) = 1 - IoU(a, b) \quad (2-3)$$

a 表示目标框，b 表示聚类产生的先验框，IoU(a,b)表示它们两者的交并比，使用 K-means 聚类算法最终得到的先验框的值如表 2-1 所示。

表 2-1 K-means 算法聚类得到的先验框的值

Table 2-1 Values of anchors obtained by K-means clustering algorithm

算法	先验框		
	大型特征图	中型特征图	小型特征图
K-means (k=9)	(29,27)	(56,77)	(107,141)
	(36,57)	(72,103)	(138,123)
	(48,72)	(87,100)	(211,227)

算法得到九个先验框的值，将它们划分为三组，小尺寸先验框对应于最终输出的大型特征图，负责检测小目标。中尺寸先验框对应中型特征图，负责检测中型目标。大尺寸先验框对应小型特征图，负责检测图像中的大目标。

2.3.3 YOLOv7-tiny 损失函数

YOLOv7-tiny 的损失函数由三部分组成：回归损失函数(L_{box}),置信度损失函数(L_{obj})和分类损失函数(L_{cls})[53]。YOLOv7-tiny 总损失函数公式如式 2-4 所示：

$$LOSS = L_{box} + L_{obj} + L_{cls} \quad (2-4)$$

其中，置信度损失函数和分类损失函数使用二元交叉熵损失，回归损失使用 CIoU 损失函数，CIoU 在 GIoU(Generalized IoU)和 DIoU 的基础上，对预测框与真实框的长宽比也进行了考虑，能够更好更快的进行预测框回归，其计算公式如下：

$$L_{box} = 1 - IoU + \frac{\rho^2(p, q)}{c^2} + \beta\gamma \quad (2-5)$$

$$IoU = \frac{A \cup B}{A \cap B} \tag{2-6}$$

$$\gamma = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{2-7}$$

$$\beta = \frac{\gamma}{(1 - IoU) + \gamma} \tag{2-8}$$

A 为预测框，B 为真实框，IoU 表示预测框和真实框的的交集区域面积与并集区域面积的比值。p, q为预测框和真实框的中心点，ρ代表计算两个中心点之间的距离，c 代表的是能够同时覆盖预测框和真实框的最小矩形的对角线距离，β是用于平衡比例的参数，γ用来衡量预测框和真实框之间的比例一致性，β和γ能够控制预测框的长宽尽快地与真实框的长宽接近。w^{gt} 和 h^{gt} 为真实框的宽和高，w 和 h 为预测框的宽和高。

2.4 迁移学习

在深度学习的实际应用中，当没有对应目标领域的公开数据集时，个体收集与制作数据集往往成本高昂，难以形成大规模的数据集合。当可以利用现有的其他领域的类似具有大量标记的数据作为源领域的知识，对目标领域的模型进行构建时，可以有效增强目标领域模型的学习效果，最终达到举一反三的效果，这就是迁移学习的思想[54]。迁移学习是针对在训练样本量有限的情况下一种有效的学习方法，可以缓解由于训练样本量不足时所导致的卷积神经网络过拟合现象[55]。

迁移学习包含两个基本概念，领域(D)和任务(T)，领域由特征空间和边缘概率分布组成，当两者有一个不同时，则认为是不同的领域。领域则又分为源领域(D_s)和目标领域(D_t)，源领域指具备一定数据的领域，目标领域则表示待解决的领域。同理，任务也分为源任务(T_s)和目标任务(T_t)，其由标记空间和预测函数组成，两者之一不同则表示任务不同。因此，迁移学习可以被定义为：在给定源域和源任务的情况下，将从中学习到的知识迁移到目标域，帮助提高预测函数的性能，其中 $D_s \neq D_t$，$T_s \neq T_t$。

此外，迁移学习可以根据分类方式的不同划分为多种不同的类型[56]。根据知识的不同类型可以将迁移学习划分为：基于实例的迁移学习、基于特征的迁移学习、基于参数的迁移学习和基于关系的迁移学习。根据源领域和目标领域的相似度可以将迁移学习划分为：归纳式迁移学习、无监督迁移学习和直推式迁移学习。根据源领域和目标领域的样本空间与标记空间的一致性，可以将迁移学习分为同构迁移学习和异构迁移学习[57]。

2.5 本章小结

本章首先对卷积神经网络的发展历程进行了概述，并对卷积神经网络的组成：输入层、卷积层、池化层、全连接层、和激活函数进行了详细的说明，包括原理与常用类型等。

其次描述了传统目标检测算法的流程及存在的弊端。并引入了深度学习在目标检测领域的发展，其算法主要可以分为两种类型：两阶段检测算法和单阶段检测算法，并分别对这两类的检测原理和优缺点进行了大致的说明，然后对两阶段检测算法的经典算法 R-CNN、SPP-Net、Fast R-CNN、Faster R-CNN 进行了概述，对这些算法的检测原理与流程进行了阐释，并说明了算法做出了哪些创新型的改进及仍存在的一些问题。并对单阶段目标检测算法典型代表 YOLO 系列算法的部分版本进行了概述，重点说明了它们的检测原理与网络结构，以及在算法的发展过程中融合的优秀思想。对比两阶段目标检测算法，YOLO 系列算法参数量小、计算少、模型简单，更适合部署在边缘设备上。

接下来详细介绍了 YOLOv7-tiny 算法，重点对其网络结构、组成模块、聚类算法以及损失函数进行了具体的阐释。本文也是选择该算法作为基础模型进行改进，以满足最终需求。

本章最后对迁移学习的思想及基本概念进行了说明，并对其分类方式及类别进行了描述，是一种能在训练样本有限的情况下提升模型性能的解决方案。

第 3 章 改进的 YOLOv7-tiny 吸烟手势检测算法

本文的最终目的是实现一个轻量且准确的吸烟手势检测算法，并在树莓派上实现一个快速、准确检测吸烟手势的系统，由于树莓派的资源有限，因此选用当前性能优异的目标检测算法 YOLOv7 的轻量级版本 YOLOv7-tiny 进行研究和改进，使之更轻量和更准确，来满足边缘设备和最终检测系统的需求。

YOLOv7-tiny 已经在常见的日常检测任务中取得良好成绩，但在树莓派上实行目标检测时仍有不足，因此需要在各方面上进一步提升。为了使模型更加轻量化和精度更高，在网络结构上，使用 Ghost 模块[58]和 Ghost 模块构建的 Ghost 瓶颈结构对部分模块和骨干网络进行重构，使模型轻量化。同时采用 CARAFE(Contenet-Aware Reassembly of Features)[59]算子实现上采样并引入 ECA[60]注意力模块，来提升模型性能。最后对 PANet 结构进行裁剪，以此降低网络的参数量和提高推理速度。在先验框设置上，采用聚类算法 K-medoids 来代替默认的 K-means 算法生成先验框。使用 Alpha-IoU[61]来代替 CIoU (Compelete IoU)[62]损失函数，加快模型收敛并提升模型精度。改进后的算法命名为 YOLOv7-tl(YOLOv7-tiny-lite)。

3.1 网络结构改进方法介绍

3.1.1 Ghost 模块

传统的卷积操作在提取特征时，会生成大量的冗余特征图，来保证模型在训练时学习到各种各样的特征[63]。如图 3-1 所示，其展示了 ResNet50 中第一个标准卷积输出的特征图的可视化结果，这些特征图展示了吸烟手势场景中不同的细节特征信息。

在图 3-1 的特征图可视化结果中，使用相同的颜色对特征相似的特征图进行标注，并将标注出的“特征图对”中的其中一张视为本征特征图，其余则视为与本征特征图相似的冗余特征图，图中可以看出存在许多的冗余特征图，这些冗余特征图包含了大量的特征信息，保障了模型的精度。但处理大量的冗余特征图往往会导致消耗大量的计算资源，并降低模型的检测速度[64]。基于此，GhostNet 提出并非所有的冗余特征图都要用卷积操作来生成，“特征图对”中的特征图都可以视为另一个特征图的幻影(Ghost)，该幻影特征图可以通过另一个特征图的简单线性变换得到，并设计出了一种全新的神经网络基本单元 Ghost 模块，解决了传统卷积操作生成冗余特征图计算量大的问题，同时也兼顾了模型精度。

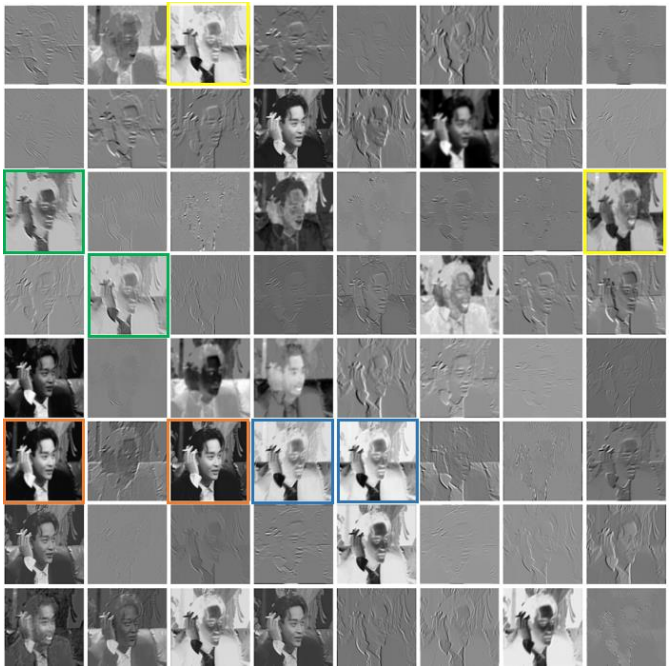

图 3-1 ResNet50 第一个卷积输出特征图

Figure 3-1 The output feature map of ResNet50 first convolution

传统卷积方式生成特征图的过程如图 3-2 所示，假设输入特征图尺寸为 $c\times w\times h$，c 表示输入通道数，w 表示输入的宽，h 表示输入的高，卷积核大小为 $k\times k$，个数为 n，输出的特征图的尺寸为 $n\times w'\times h'$，则传统卷积方式的计算量(FLOPs)表示如下：

$$FLOPs = n \times w' \times h' \times c \times k \times k \tag{3-1}$$

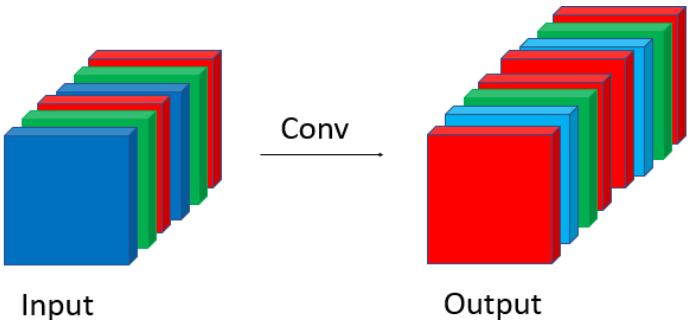

图 3-2 传统卷积

Figure 3-2 Traditional convolution

Ghost 模块如图 3-3 所示，与传统卷积方式不同，该模块由三个部分组成，首先使用传统卷积降维得到少量基础特征图 Y，再对得到的基础特征图的各个通道对

应的特征图分别进行线性变换 ϕ，得到与之对应的 Ghost 特征图 Y'，最后将基础特征图 Y 进行恒等映射并与得到的 Ghost 特征图进行拼接，得到最终输出。

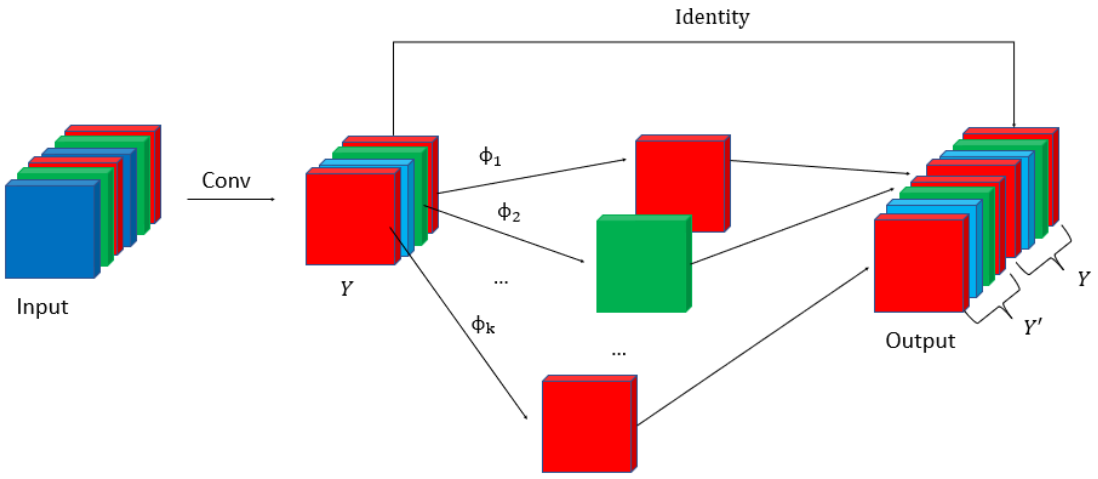

图 3-3 Ghost 模块

Figure 3-3 Ghost module

同样，假设输入特征图形状为 $c\times w\times h$，第一步传统卷积卷积核大小为 $k\times k$，个数为 $m(m\leqslant n)$，得到的基础特征图 Y 的形状为 $m\times w'\times h'$。对基础特征图 Y 的每个通道对应的特征图进行线性变换，得到s个特征图[65]，对于 m、s、n，三者之间的关系为：

$$n = m \times s \tag{3-2}$$

Ghost 模块的计算量(FLOPs-G)由传统卷积的计算量和去掉恒等映射后的线性变换生成 Ghost 特征图产生的计算量两部分组成，表示如下：

$$\begin{aligned} FLOPs-G &= m\times w'\times h'\times c\times k\times k+(s-1)\times m\times w'\times h'\times d\times d \\ &= \frac{n}{s}\times w'\times h'\times c\times k\times k+(s-1)\times\frac{n}{s}\times w'\times h'\times d\times d \end{aligned} \tag{3-3}$$

公式 3-3 中 s-1 代表基础特征图每个通道去掉恒等映射后线性变换生成的特征图，$d\times d$ 表示 Ghost 模块中线性变换的内核平均大小，大小与 $k\times k$ 近似。

Ghost 模块与传统卷积模块的加速比为：

$$\begin{aligned} r_s &= \frac{n\times w'\times h'\times c\times k\times k}{\frac{n}{s}\times w'\times h'\times c\times k\times k+(s-1)\times\frac{n}{s}\times w'\times h'\times d\times d} \\ &= \frac{c\times k\times k}{\frac{1}{s}\times c\times k\times k+\frac{(s-1)}{s}\times d\times d}\approx\frac{s\times c}{s+c-1}\approx s \end{aligned} \tag{3-4}$$

同样，Ghost 模块与传统卷积模块的参数压缩比为：

$$r_c = \frac{n \times c \times k \times k}{\frac{n}{s} \times c \times k \times k + (s-1) \times \frac{n}{s} \times d \times d} \approx \frac{s \times c}{s + c - 1} \approx s \qquad (3-5)$$

从式 3-4 和 3-5 可以看出，参数压缩比大约等于加速比，Ghost 模块通过简单的线性变换得到了与传统卷积模块相同数量的特征图，在保证模型能够学到足够的特征的同时将计算开销压缩了s倍。

此外，Ghost 模块的灵活性使得它可以在网络的任何地方进行插入使用。使用 Ghost 模块来对原网络中(非骨干网络)的 ME-ELAN 和 MSPPCSPC 模块进行改造，形成的新模块如图 3-4 所示，分别命名为 GME (Ghost-ME-ELAN)和 GMS(Ghost-MSPPCSPC)。

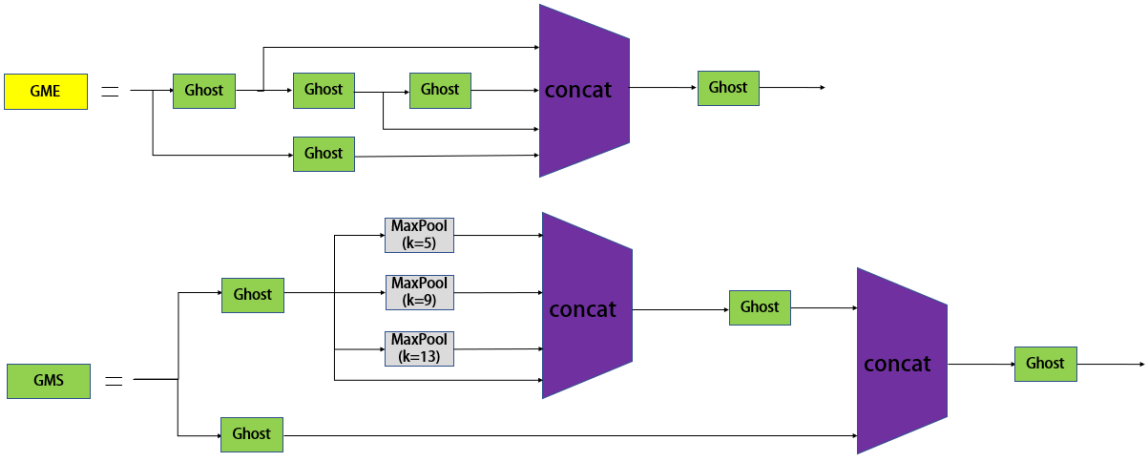

图 3-4 GME 模块和 GMS 模块结构图

Figure 3-4 The structure of GME and GMS module

3.1.2 Ghost 瓶颈结构

利用 Ghost 模块可以构建 Ghost 瓶颈结构，Ghost 瓶颈结构采用先升维后降维的形式，能够有效避免信息在不同维度之间转换时出现损失的问题。Ghost 瓶颈结构共有两种结构，以卷积步长来进行区分。

当步长为 1 时，Ghost 瓶颈结构由两个 Ghost 模块组成，第一个 Ghost 模块用于扩展输入通道数，第二个用于减少通道数来匹配 shortcut 连接[66]。第一个模块后使用 BN 操作和 ReLU 激活函数，第二个模块后只使用 BN 操作，最后将输入特征图和第二个模块的 BN 输出使用 shortcut 相加，得到最终输出。当步长为 2 时，在步长为 1 的 Ghost 瓶颈结构的基础上，在两个 Ghost 模块之间添加了一个步长为 2 的深度卷积，并在添加的深度卷积模块后使用批归一化操作。步长为 2 的 Ghost 瓶

颈结构模块命名为 GBS(Ghost Bottleneck Stucture) ，其结构如图 3-5 所示。

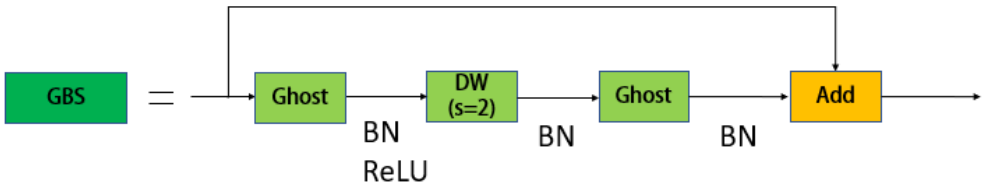

图 3-5 GBS 模块结构图

Figure 3-5 The structure of GBS module

步长为 1 的 Ghost 瓶颈结构不改变输入特征图的大小，步长为 2 的 Ghost 瓶颈结构使得输入特征图的大小减半，同时 Ghost 瓶颈结构具有和 Ghost 模块相同的即插即用特性，因此可根据网络中不同尺寸的特征图输出需求来使用 Ghost 瓶颈结构。在本文中，主要使用步长为 2 的 Ghost 瓶颈结构来对 YOLOv7-tiny 的骨干网络进行重构，使得新形成的网络结构更加轻量并且保持足够的特征提取能力，新形成的骨干网络命名为 GBS-4，其结构如图 3-6 所示。

图 3-6 骨干网络 GBS-4 结构图

Figure 3-6 The structure of backbone GBS-4

3.1.3 CARAFE 上采样算子

特征上采样是许多卷积神经网络架构的关键操作，其设计能对目标检测、语义分割等领域产生一定的影响，其操作可以表示为每个位置的上采样核和输入特征图中对应领域的像素做点积，称为特征重组。YOLOv7-tiny 中使用最近邻插值来进行特征图上采样，但最近邻插值只通过像素点的空间位置来决定上采样核，没有利用到特征图的语义信息，并且感受野通常比较小。

因此引入 CARAFE 上采样算子代替最近邻插值，其结构如图 3-7 所示。CARAFE 上采样过程分为两个部分：上采样核预测和特征重组。算法整体流程为：

1.假定输入特征图形状为 $h\times w\times c$，上采样倍率为 μ，则输出的特征图尺寸为 $\mu h\times \mu w\times c$。首先使用 1×1 卷积压缩输入特征图通道数到 c'，减少后续步骤计算量。

2.假设上采样核大小为 $k\times k$，则需要预测的上采样核形状为 $\mu h\times \mu w\times k\times k$。利用一个 $k'\times k'$的卷积层来预测上采样核，输入特征图形状为 $h\times w\times c'$，输出特征图形状

为 $h\times w\times k^2\times \mu^2$，然后将特征图的通道维在空间维展开，得到形状为 $\mu h\times \mu w\times k^2$ 的上采样核。

3.将利用 softmax 函数对上采用核进行归一化处理，使得卷积核权重之和为 1。

4.将输入特征图与得到上采样核进行卷积运算，得到最终的输出。

CARAFE 上采样算子的上采样核利用输入特征图预测生成，包含更多的语义信息，同时具有更大的感受野，能更好的利用像素点周围的信息。并且 CARAFE 上采样算子只引入了极少量的参数和计算量，实现了精度的提升，是一个高效的轻量级算子。其参数量 P 如公式 3-6 所示[67]：

$$P = cc' + \left(c'k'^2\mu^2 + 1\right)\mu^2\mathrm{k}^2 + \mu^2\mathrm{k}^2c \qquad (3-6)$$

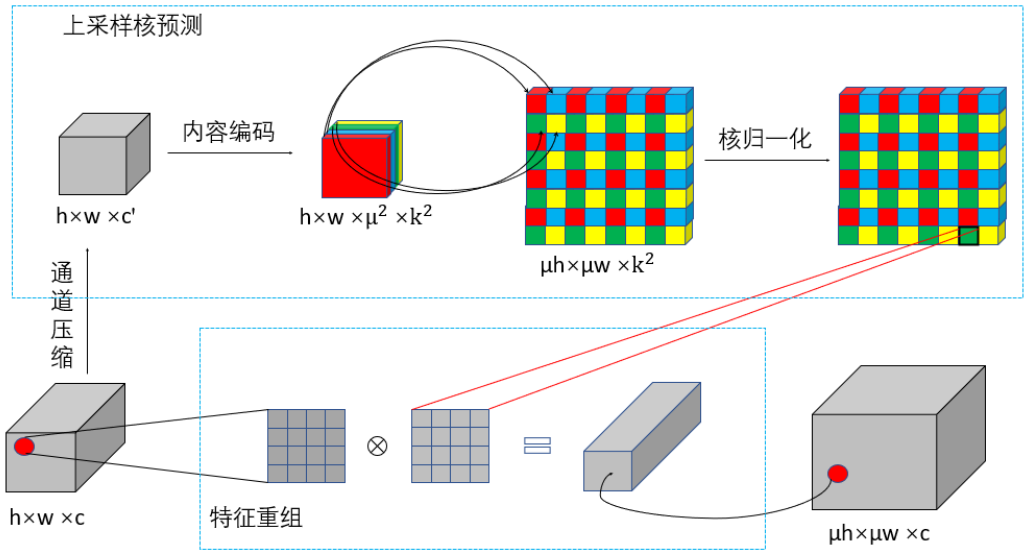

图 3-7 CARAFE 上采样算子结构图

Figure 3-7 The structure of CARAFE up-sampling operator

3.1.4 ECA 通道注意力模块

近年来，通道注意力机制在改善和提高卷积神经网络的性能上表现十分出色，各式各样的注意力机制层出不穷，但为了获得更好的性能，大多数注意力机制设计的越来越复杂，不可避免的带来了更多的参数量和计算量，使得模型更加复杂[68]。而 ECA 却克服了这一困境，只引入了少量参数，却能带来明显的性能提升，实现了性能和复杂性之间的平衡。本文的吸烟手势数据集中吸烟手势情况复杂且背景多样，ECA 模块能有效提高吸烟手势特征对网络模型的影响，从而加强对吸烟手势的定位。

ECA 模块主要在 SE(Squeeze-and-Excitation)注意力模块的基础上进行改进，取

消了 SE 模块中全连接层的降维操作，转变为一维卷积并实现局部跨通道交互，有效避免全连接层带来的信息丢失和大计算量，ECA 模块结构如图 3-8 所示。

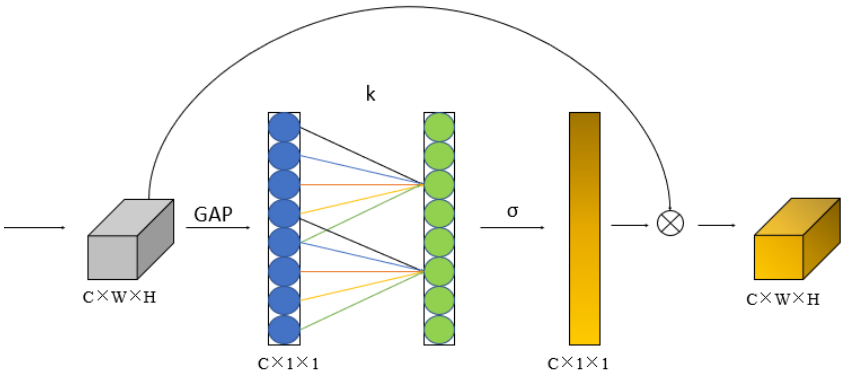

图 3-8 ECA 模块结构图

Figure 3-8 The structure of ECA module

ECA 模块流程由三个部分组成，假定输入图像尺寸为 $C\times W\times H$，第一步先采用全局平均池化(Global Average Pooling,GAP)对输入图像进行处理，将输入图像沿着空间维度进行特征压缩，得到 C×1×1 大小的特征图。接下来进行卷积核大小为 k 的一维卷积操作，并经过 Sigmoid 激活函数确定各个通道的权重 ω。k 不仅为卷积核的大小，也表示跨通道局部交互的覆盖范围[69]，同时，k 的大小根据通道数 C 自适应确定，如公式 3-7 所示。权重 ω 的计算则与 k 相关联，其计算公式如 3-8 所示。最后将获取的权重 ω 与原始输入特征图的相应元素相乘，得到最终的加权特征图。

$$k = |\frac{log_2 C + 1}{2}|_{odd} \tag{3-7}$$

$$\omega = \sigma(C1D_k(y)) \tag{3-8}$$

式 3-7 中，k 表示卷积核大小，C 表示总通道数，$|x|_{odd}$ 表示离 x 最近的奇数。同样，式 3-8 中，k 表示卷积核大小，C1D 表示一维卷积，σ 表示 Sigmoid 激活函数。

本文在改进后的 YOLOv7-tl 算法的骨干网络尾部及特征融合部分添加 ECA 注意力模块，计算并分配特征图通道上的权重信息，使得网络能学习到更多有关吸烟手势的特征，从而提高模型性能。

3.1.5 特征融合结构改进

YOLOv7-tiny 网络结构采用 PANet 结构进行特征融合并进行多尺度预测，PANet 结构在特征金字塔网络(Feature Pyramid Network,FPN)结构的基础上增加了

一个自下到上的路径结构。FPN 从上到下对网络不同层次的特征进行提取与融合，将语义信息传给浅层特征，自下向上的结构则将底层特征融合到高层特征中，增加了高层特征的定位特征。

尽管 PANet 结构带来了模型性能上的提升，但过多的特征融合和分支结构导致大量的参数量和计算量，显著的增加了边缘设备的计算成本。因此对 PANet 的自下向上结构进行部分裁剪，舍弃小尺度的输出分支，形成的新的特征融合结构命名为 SimPAN (Simple-PANet)。小尺度的输出分支主要负责检测大型目标，但本文所使用的吸烟手势数据集大多数是小型和中型目标，所以与原始结构相比，新形成的特征融合结构可能会导致一定程度的性能下降，但影响不显著。一方面保留下的 FPN 结构确保深层语义信息被传递到浅层特征中，为小目标检测提供支持。另一方面，去掉的部分结构会使得模型参数量和计算量降低，提高模型推理速度，更适合边缘设备进行部署。最后形成的 YOLOv7-tl 完整的网络结构如图 3-9 所示。

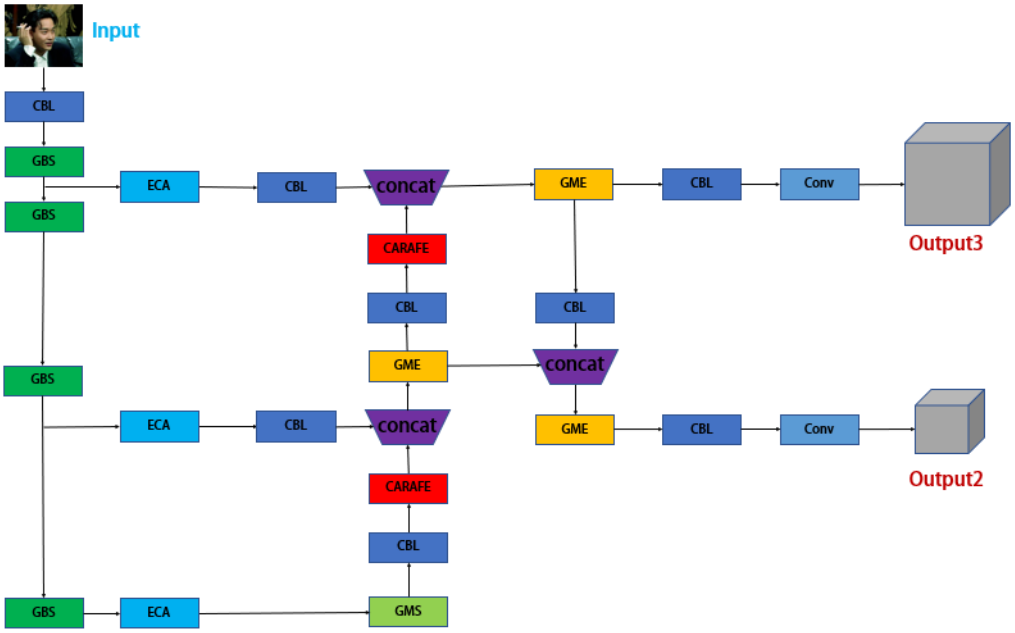

图 3-9 YOLOv7-tl 算法结构图

Figure 3-9 The structure of the YOLOv7-tl algorithm

3.2 聚类算法改进方法介绍

为了保证数据集的多样性，数据集往往包含不同场景下的图像，所标注的框大小差异很大，出现离群噪音点不可避免。然而，K-means 算法基于平均值的聚类方式会被离群噪音点所影响，导致聚类结果出现偏差。因此，改进后的 YOLOv7-tl 算

法使用 K-medoids 算法来解决这一问题，K-medoids 算法使用簇中位置最中心的点来聚类，削弱了离群噪音点的影响，使得聚类结果更加准确。其算法流程如下：

1.在样本空间中，随机选取 k 个数据集中的样本框作为初始聚类中心。

2.计算其他样本框与所有聚类中心之间的距离，并将其划分到距离最小的聚类中心，形成 k 个簇，距离计算公式如下所示。

$$d(b,c) = 1 - IoU(b,c) \tag{3-9}$$

b 表示普通样本框，c 表示簇中心。IoU 表示两个框的交并比。

3.根据分类好的簇，重新计算簇中心:

①在每个簇中，将所有样本框作为一次簇中心，并计算其它样本框到簇中心的距离之和。计算公式如式 3-10 所示。

$$S = \sum_{b_i \in c_i} d(b_i, c_i) \tag{3-10}$$

S 表示距离之和，c_i 表示簇中心，b_i 表示簇中其它样本点。

②选取距离之和最小的样本框作为簇中心。

4.重复步骤 2 和步骤 3，直到簇中心不再发生变化，得到最终先验框的值。

使用 K-medoids 聚类得到的先验框的值如表 3-1 所示。

表 3-1 K-medoids 算法聚类得到的先验框的值

Table 3-1 Values of anchors obtained by K-medoids clustering algorithm

算法	先验框		
	大型特征图	中型特征图	小型特征图
K-medoids (k=9)	(33,29)	(58,46)	(101,177)
	(36,61)	(73,124)	(150,114)
	(51,92)	(88,74)	(203,224)
K-medoids (k=6)	(39,44)	(94,164)	
	(58,103)	(147,113)	
	(84,68)	(203,224)	

为了便于对比实验分析，分别设置 k 为 9 和 6，K-medoids 算法聚类得到两组不同的先验框。

3.3 损失函数改进方法介绍

在基于先验框的目标检测器中，通常将 bounding box 回归和目标分类分为两

个并行任务进行学习[70]。在 bounding box 回归任务中，早期使用 l_n-norm 作为损失函数，但后续研究表明，l_n-norm 对 bounding box 的尺度变化十分敏感，因此后续通常使用 IoU 损失及其衍生出的变体来进行回归[71]。当使用 IoU 来度量 bounding box 回归时，当预测框与真实框没有交集时，IoU 为 0，在训练网络时，无法计算梯度，进行优化。同时，存在多个预测框同时与真实框 IoU 相同的情况，然而不同位置的预测框对定位的效果也会造成不同影响。基于此，学者们先后提出了不同的损失函数，如 GIoU、DIoU、CIoU 等来解决这一问题。

在 GIoU 中，引入了预测框与真实框的最小外接矩形，以反映两者的重合程度，解决了预测框与真实框 IoU 为 0 时导致的梯度消失问题。在 DIoU 中，还考虑了预测框与真实框之间的中心点距离、重叠率和尺度，使得 bounding box 回归比 GIoU 更稳定，达到了更快的收敛速度。而在 DIoU 的基础上，CIoU 不仅考虑了重叠面积、中心点距离，还考虑了纵横比，进一步提升了模型收敛速度和模型精度。

Alpha-IoU 损失函数则包含了上述所有的损失函数的特性，是在现有的 IoU 损失函数上进行幂变换得到的一个新的 IoU 族。Alpha-IoU 除了具备了上述损失函数优点外，还能够自适应地重新加权高 IoU 目标的损失和梯度，有效提高了 bounding box 回归和目标检测精度。对 IoU loss 进行幂变换后得到的公式如下所示：

$$L_{box} = \frac{1 - IoU^{\alpha}}{\alpha} \tag{3-11}$$

α表示幂变换，当α=1 时，则表示为传统的 IoU 损失函数。

通过对上述幂变换思想进行推广，可得到 GIoU、DIoU、CIoU 幂变换后的损失函数，其中对 CIoU 进行幂变换后得到公式如下所示：

$$L_{box} = 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}(p,q)}{c^{2\alpha}} + (\beta\gamma)^{2\alpha} \tag{3-12}$$

本文所提出的 YOLOv7-tl 算法使用公式 3-12 所得到的 Alpha-IoU 代替 YOLOv7-tiny 算法使用的 CIoU 作为回归损失函数。在后续的实验中表明，使用 Alpha-IoU 作为损失函数对模型进行训练，能有效提升模型检测精度。

3.4 本章小结

本章对 YOLOv7-tiny 算法进行了改进，并对所采用的改进策略进行了详细的原理阐释与使用说明。考虑到吸烟手势目标检测的实际部署环境与需求，需要在提升精度的同时使得模型进一步轻量化，提升速度，在两方面保持平衡，达到在检测时快速且准确的效果。

为了使模型进一步的轻量化，引入 Ghost 模块对 YOLOv7-tiny 中的一些模块进行改进，使用 Ghost 模块替换掉传统的卷积模块，能有效保证模型学到足够多的

的特征的同时大幅度减少了模型计算量和参数量。此外，使用 Ghost 模块构建的 Ghost 瓶颈结构对原网络的骨干部分进行了重构，在保持足够特征提取能力的同时，使模型进一步轻量化。同时，为了提升网络性能，引进 CARAFE 上采样算子和 ECA 通道注意力模块，CARAFE 上采样算子代替最近邻插值实现上采样操作，能使特征图融合更多的语义信息，实现模型精度提升。ECA 通道注意力模块能使得网络能学习到更多有关吸烟手势的特征并加强对吸烟手势定位，从而提高模型性能。此二者均只引入少量参数量和计算量，不会导致模型轻量化程度降低。

YOLOv7-tiny 采用 PANet 结构进行特征融合，但过多的特征融合和分支结构带来了大量的参数量和计算量。因此在对吸烟手势数据集进行分析后，对自下向上分支进行了裁剪，并舍弃了最终的小尺寸特征图输出，只保留对小型目标和中型目标检测的输出，损失少量性能的同时有效提升了模型推理速度。

最后使用 K-medoid 聚类算法和 Alpha-IoU 损失函数代替原本的方法作为先验框聚类算法和模型回归函数，K-medoid 聚类算法生成的先验框更适配吸烟手势数据集，Alpha-IoU 损失函数则能够自适应地重新加权高 IoU 目标的损失和梯度，提高 bounding box 回归和目标检测精度，二者都有效的提升了模型性能。

第 4 章 实验与分析

4.1 数据集介绍

目前深度学习检测吸烟行为方法众多，包括：吸烟手势、香烟烟雾、香烟本身等，各种检测目标所使用的数据集也有所差异，因此目前在吸烟手势检测领域没有通用的、权威的公开数据集。本文的吸烟手势数据集主要通过在互联网上用 Python 爬虫和实际拍摄收集而来，对收集到的样本图片进行筛选、清洗，选出不同场景、不同姿势和不同方位的高质量样本图片，包含电梯、户外、宿舍等多种场景，以及正面、侧面和背面等多种姿势，最后共得到 2300 张图片。收集到的图片采用 LabelImg 工具进行标注，为了满足对比实验其他模型的训练需求，先将图片标注为 PASCAL VOC 格式得到 xml 文件，然后将 xml 文件转化为 YOLO 模型所需要的 txt 文件，如图 4-1 所示。

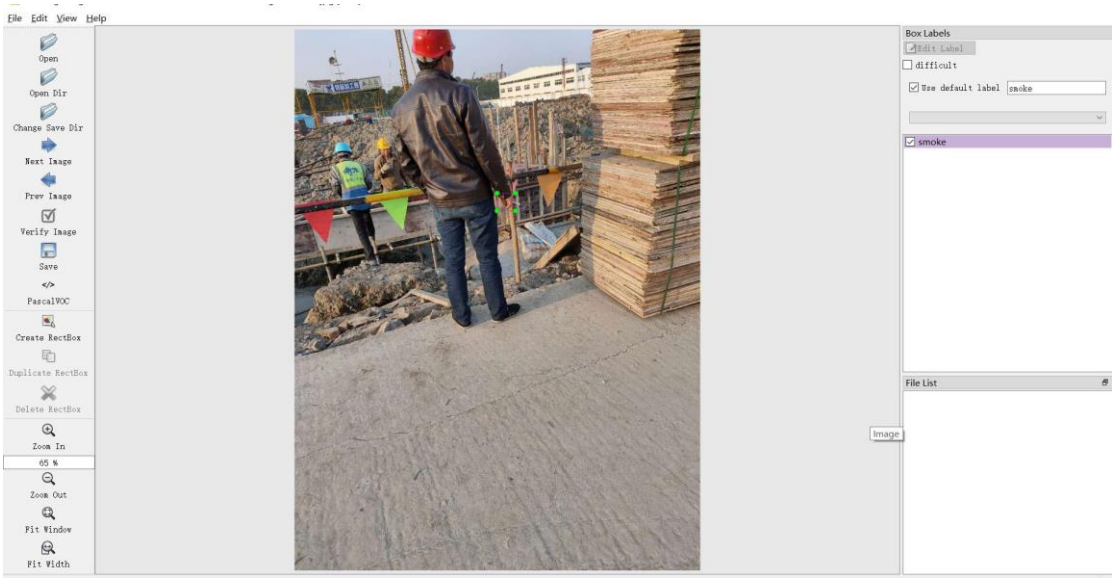

图 4-1 LabelImg 标注图像过程

Figure 4-1 Process of LabelImg image annotation

将处理好的数据集划分为训练集、验证集和测试集(7:2:1)，如表 4-1 所示。同时在模型训练时使用数据增强，数据增强可以有效扩充数据集的样本量，并提高模型在不同场景下的泛化性和鲁棒性，包括：旋转、翻转、对比度调节等。除此之外，使用 Mosaic 数据增强方法，原理为随机选取数据集中四张图片进行裁剪并拼接为一张图片，并将其传入到网络中进行学习，有效的丰富了检测物体背景。

表 4-1 吸烟手势数据集

Table 4-1 Smoking gestures dataset

总数据集	训练集	验证集	测试集
2300	1610	460	230

4.2 实验环境

本文实验环境：深度学习框架为 Pytorch 1.10，系统为 Ubuntu 18.04，GPU 型号为 NVIDIA GeForce RTX 3090(24GB)，CPU 型号为 AMD EPYC 7773X 64-CORE Processor，内存大小为 30GB ，Python 版本为 3.8.13，OpenCV 版本为 4.5.3，采用 CUDA11.1 和 cuDNN8.2.1 加速模型训练。其他训练超参数设置如表 4-2 所示。

表 4-2 训练超参数设置

Table 4-2 Setting of training hyperparameters

参数	参数值
图像大小(image size)	416×416
训练批次(epoch)	135
初始学习率(lr0)	0.001
循环学习率(lrf)	0.1
动量(momentum)	0.937
权重(decay)	0.0005
批尺寸(batch_size)	32
线程数(worker)	8
优化器(optimizer)	Adam

4.3 模型评估指标

为了对本文所提模型及各类对比模型进行性能评估，选取平均准确率均值(mean Average Precision,mAP)，准确率 P(Precision)、召回率 R(Recall)、F1 分数、参数量 Param (Parameter)、计算量(GFLOPs)和模型大小(Size)等作为评估指标。另外，最终所得到模型会部署在树莓派上进行推理，所以每秒传输帧数(Frames Per Second, FPS)也是一个重要指标。

①准确率：为所有预测值为正样本中真实值为正样本所占的比例，反映了模型的错检程度。

②召回率：为所有真实值为正样本中预测值为正样本所占的比例，反映了模型的漏检程度。

③平均准确率均值：为计算每一类的平均准确率 (Average Precision,AP)，然后求平均值得到，反映了模型的整体性能。

④参数量：表示模型网络结构的参数总和，反映模型的空间复杂度。

⑤计算量：表示计算整个网络模型中乘法/加法的运行次数，反映模型计算复杂度。

⑥模型大小：表示模型实际大小，为衡量模型空间复杂度的另一个参数。

⑦每秒传输帧数：表示每秒可以处理多少张图片，用来衡量模型检测的实时性。

⑧F1 分数：定义为准确率和召回率的调和平均数，是两者的综合，能反映模型检测的准和全的能力。

部分指标计算公式如下：

$$Precision = \frac{TP}{TP + FP} \tag{4-1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4-2}$$

$$AP = \int_0^1 p(r)dr \tag{4-3}$$

$$mAP = \frac{1}{c}\sum_{i=1}^{c} AP_i \tag{4-4}$$

$$F1 = \frac{2PR}{P + R} \tag{4-5}$$

TP、TN、FP、FN 定义如图 4-2 所示。其中，TP 表示将正样本预测为正样本的数目，TN 表示将负样本预测为负样本的数目，FP 表示将负样本预测为正样本的数目，FN 表示将正样本预测为负样本的数目，c 为检测的类别数。每个类别根据其准确率和召回率绘制的 P-R 曲线所包围的区域面积大小即为 AP。

混淆矩阵		预测值	
		正样本	负正样
真实值	正样本	TP	FN
	负样本	FP	TN

图 4-2 TP、TN、FP、FN 定义

Figure 4-2 Definitions of TP, TN, FP, and FN

4.4 实验对比与分析

4.4.1 迁移学习实验

由于本文所采用的数据集图片数量偏少，尽管训练时采用了数据增强的方式扩充数据集，但仍存在模型过拟合的风险。采用迁移学习的方式预训练可以缓解由于训练样本量不足时所导致的卷积神经网络过拟合现象，同时迁移学习还可以有效提升模型的泛化能力，并能提升模型在目标领域的性能。

本文迁移学习所使用的源领域的数据集为 Pascal VOC 2007 数据集，该数据集划分为训练集（5011 张图片）和测试集（4952 张图片），共计 9963 张图片，包含人、自行车、汽车等 20 个类，这些图片包含丰富的特征信息，是图像识别与分类的标准化图像数据集之一。为了验证迁移学习的有效性，在吸烟手势数据集上进行了实验，实验结果如表 4-3 所示。

表 4-3 迁移学习实验

Table 4-3 Experiment of transfer learning

算法	迁移学习	P(%)	R(%)	F1(%)	mAP(%)
YOLOv7-tiny		71.4	68.9	70.1	71.5
	√	88.2	86.6	87.4	92

实验表明，使用了迁移学习后，模型的性能明显提升，准确率提升了 16.8%，召回率提升了 17.7%，F1 分数提升了 17.3%，mAP 提升了 20.5%，证明了迁移学习的有效性，后续实验都在此基础上进行，使用该预训练权重进行初始化。

4.4.2 Alpha-IoU 实验

为了验证本文所采用的 Alpha-IoU 回归损失函数的有效性并确认最佳的幂变换的α值，在数据集上进行了不同α取值(1-10)时的 YOLOv7-tiny 性能对比实验，当α=1 时，Alpha-IoU=CIoU，模型为初始模型。实验结果如表 4-4 所示。

采用 Alpha-IoU 作为回归损失函数后，不同α的取值对模型施加了不同的影响，从 mAP 上看，只有当α值取 4、6 和 10 时，mAP 出现了下降。从准确率和召回率上看，各有上升和下降，因此用 F1 分数来衡量两者的综合水平，只有当α值取 2 和 3 时，F1 分数取得了提升。当α=3 时，模型 mAP 达到 93.5%，准确率达到 88.4%，召回率达到 88.1%，F1 分数达到 88.2%，mAP 达到最高，尽管准确率和召回率没有达到最高，但都位于前列，同时 F1 分数达到最高，且四个指标均高于α=1 时的

初始模型。实验表明，当α取值为 3 时，模型性能达到最好。因此在后续实验中，均使用α=3 时的 Alpha-IoU 回归损失函数进行实验。

表 4-4　不同α值下 YOLOv7-tiny 性能对比

Table 4-4 Performance comparison of YOLOv7-tiny with different α values

α	P(%)	R(%)	F1(%)	mAP(%)
1	88.2	86.6	87.4	92
2	88.7	87.3	88.0	93.2
3	88.4	88.1	88.2	93.5
4	82.9	87.3	85.0	91
5	85.1	88.5	86.8	93.4
6	83.0	86.0	84.5	90.8
7	85.6	87.3	86.4	92.4
8	92.3	82.4	87.1	93.1
9	86.6	88.1	87.3	93.1
10	86.6	79.9	83.1	89.6

4.4.3 消融实验

为了验证本文所采用的各项改进的有效性，在吸烟手势数据集上进行消融实验，以确保使用的改进的有效性。实验结果如表 4-5 所示。

无代表最原始的 YOLOv7-tiny 算法，全部代表改进后得到 YOLOv7-tl 算法，以此为基准，进行实验对比分析。从表 4-5 可以看出，施加不同改进后，准确率和召回率各有上升和下降，因此用 F1 分数来衡量两者的综合水平并对模型进行评估。

在网络中（非骨干网络）使用 GME 模块替换 ME-ELAN 模块后，模型 F1 分数保持不变，mAP 上升了 0.3%，参数量降低了 6.8%，计算量降低了 6.1%，模型大小减小了 6.8%。使用 GMS 模块替换 MSPPCSPC 模块后，模型 F1 分数提升了 0.2%，mAP 上升了 0.3%，参数量降低了 5.2%，计算量降低了 2.3%，模型大小减小了 5.1%，实验结果表明，使用 Ghost 模块替换传统卷积模块后，保证了网络学习到足够特征的同时降低了一部分参数量和计算量，减少了网络开销，验证了其有效性。

使用 GBS 模块对骨干网络进行重构后，新形成的 GBS-4 骨干网络使得模型 F1 分数下降了 4.8%，mAP 下降了 5.1%，参数量降低了 35.3%，计算量降低了 42.4%，

模型大小减小了 35%。使用简化后的特征融合结构 SimPAN 后，模型 F1 分数下降了 4.3%，mAP 下降了 1.8%，参数量降低了 33.9%，计算量降低了 12.9%，模型大小减小了 34.2%。实验结果表明，重构后的骨干网络和简化后的特征融合结构 SimPAN 大幅降低了网络参数量、计算量和模型大小，但也导致了模型一部分性能的损失，基本符合预期。

使用 CARAFE 上采样算子代替最近邻插值进行上采样后，模型 F1 分数提升了 0.5%，mAP 提升了 1.4%，参数量上升了 0.8%，计算量上升了 0.75%，模型大小增加了 0.85%。引入 ECA 通道注意力模块后，模型 F1 分数提升了 1.3%，mAP 提升了 0.8%，参数量、计算量和模型大小基本保持不变。实验结果表明，引入 CARAFE 上采样算子和 ECA 通道注意力模块后，有效的提升了网络性能并只增加了极少量的开销。

使用 K-medoids 聚类算法和 Alpha-IoU 损失函数后，模型的 F1 分数分别提升了 1.0%和 0.8%，mAP 分别提升了 1.7%和 1.5%，但没有影响模型的参数量、计算量和模型尺寸。表明 K-medoids 聚类得到的先验框和 Alpha-IoU 损失函数有效的提升了目标定位性能，提升了模型精度。

表 4-5 消融实验结果

Table 4-5 Results of ablation experiments

改进	P(%)	R(%)	F1(%)	mAP(%)	Param(10^6)	GFLOPs	Size(M)
GME	84.6	90.4	87.4	92.3	5.60	12.4	10.9
GMS	86.9	88.3	87.6	92.3	5.70	12.9	11.1
GBS-4	81.6	84	82.3	87.2	3.89	7.6	7.6
SimPAN	82.7	83.6	83.1	90.2	3.97	11.5	7.7
CARAFE	84	92.2	87.9	93.4	6.06	13.3	11.8
ECA	90.1	87.3	88.7	92.8	6.01	13.2	11.7
K-medoids	88.6	88.3	88.4	93.7	6.01	13.2	11.7
Alpha-IoU	88.4	88.1	88.2	93.5	6.01	13.2	11.7
无	88.2	86.6	87.4	92	6.01	13.2	11.7
全部	86.6	91.2	88.8	92.6	1.42	5.3	3.0

综合上述所有改进后得到的 YOLOv7-tl 算法，F1 分数达到 88.8%，mAP 达到 92.6%，参数量达到 1.42(10^6)，计算量(GFLOPS)达到 5.3，模型大小为 3.0M，对比原始 YOLOv7-tiny 算法，F1 分数提升了 1.4%，mAP 提升了 0.6%，参数量下降了

76.4%，计算量下降了 59.8%，模型大小降低了 74.4%，实验结果表明，改进后得到的 YOLOv7-tl 算法更轻量且更准确，并实现了两方面的平衡，更适合部署到树莓派上。

4.4.4 不同算法对比实验

为了更进一步的验证改进后的 YOLOv7-tl 算法的有效性，选取部分目前目标检测领域轻量或准确的算法进行对比，包括：YOLOv3-tiny、YOLOv4-tiny、YOLOv4、SSD、YOLOX-tiny、YOLOv7-tiny 以及 YOLOv5s。选取平均准确率均值(mAP)、参数量(Param)、计算量(GFLOPs)、模型大小(Size)和 FPS 作为评估指标，其中 FPS 数据为在树莓派上测量得到。实验结果如表 4-6 所示。

表 4-6 不同算法对比实验

Table 4-6 Comparison experiments of different algorithms

算法	F1(%)	mAP(%)	Param(10^6)	GFLOPs	Size(M)	FPS(帧/s)
YOLOv3-tiny	79.1	79.9	8.67	13.0	16.6	8.12
YOLOv4-tiny	78.8	82.6	3.06	6.4	5.9	16.3
YOLOv4	82.9	87.6	60.42	131.6	115.8	1.44
SSD	84.1	89.7	23.61	60.75	90.6	1.02
YOLOv5s	89.4	92.3	7.02	15.9	13.6	7.46
YOLOX-tiny	93.2	89.2	5.06	6.49	38.9	6.84
YOLOv7-tiny	87.4	92	6.01	13.2	11.7	13.94
YOLOv7-tl	88.8	92.6	1.42	5.3	3.0	25.6

数据表明，对比表中 YOLOv3-tiny、YOLOv4-tiny、YOLOv4、SSD、YOLOX-tiny、YOLOv7-tiny 以及 YOLOv5s 等算法，YOLOv7-tl 算法在 mAP、参数量、计算量、模型大小和 FPS 等五项指标上均远远超出。只在 F1 分数上略微逊色于 YOLOX-tiny 和 YOLOv5s。实验结果证明了本文所提出的 YOLOv7-tl 算法的先进性，实现了精度和速度上的平衡，能有效在树莓派上完成吸烟手势检测任务。

4.5 YOLOv7-tl 检测效果测试

为进一步验证 YOLOv7-tl 算法的效果，图 4-3 展示了不同情况下对吸烟手势的检测效果。

图 4-3(a)展示了在光线较暗下的检测情况，图 4-3(b)展示了在光线较亮下的检测情况，同时检测目标为小目标，图 4-3(c)展示了在人物背身的检测情况，检测目

标存在遮挡，图 4-3(d)展示了有多个目标的检测情况，同时存在有相似手势进行混淆。在四种不同的场景下，YOLOv7-tl 均成功检测出所有吸烟手势。结果表明：本文提出的 YOLOv7-tl 算法实际效果测试优异，能检测出不同尺寸吸烟手势，同时在复杂场景下：暗光、亮光、遮挡、混淆手势等表现良好，没出现漏检、误检等情况，具有一定的应用价值，适用于树莓派等边缘设备。

(a)暗光

(b)亮光

(c)背身

(d)多目标

图 4-3 YOLOv7-tl 不同场景下检测效果

Figure 4-3 Detection results of YOLOv7-tl in different scenarios

4.6 本章小结

为了验证本文所提出的 YOLOv7-tl 算法的性能，对算法进行了实验与分析。

本章首先对所用的吸烟手势数据集进行了说明，先对图片进行收集、清洗和制作吸烟手势标签，并将数据集划分为训练集、验证集和测试集进行实验，还在训练时采用数据增强的方式扩充数据集，提高模型鲁棒性。然后对实验所处的环境和对实验进行评估的指标进行了介绍。

接下来对所使用的改进进行了详细的测试和消融实验，先验证了迁移学习的有效性，然后对 Alpha-IoU 进行了测试，选取出了最优的α值使得模型性能提升。同时对其他所使用的改进都进行了实验，实验结果表明：以 YOLOv7-tiny 为基准，所使用的改进均取得了理想效果，达到了使模型更轻量化或性能更强的目的，最后得到的 YOLOv7-tl 在保持了精度的同时，模型的参数量、计算量和大小均大幅减少。除此之外，也将本文提出的 YOLOv7-tl 算法与其他经典的目标检测算法进行了各方面的对比，同时还包括在树莓派上的 FPS 对比。对比结果显示，本文提出的算法在各方面表现优异，其中 FPS 位列第一，表明能在树莓派上实现快速、准确检测。

最后对所提出的 YOLOv7-tl 算法进行了模型检测效果测试，结果表明所提出的 YOLOv7-tl 算法适用于各种复杂场景，能满足检测实际需求。

第 5 章 基于树莓派的吸烟手势检测系统设计与实现

5.1 系统硬件环境介绍

5.1.1 树莓派简介

树莓派(Raspberry Pi)是一款卡片式计算机，由英国一个非营利机构“树莓派基金会”开发,最初的目的是用于对少年儿童进行计算机普及教育。其优秀的扩展性和易于开发的特性，被用作多种用途。基于 ARM 的芯片可以满足部分负荷较小的计算机功能，例如音视频播放、文档处理、代码编写等，而且其拥有多个 GPIO 接口，使其能通过编写程序与外界各种设备进行交互[72]。由于树莓派的资源局限性及其体积小、功耗低、低成本的特点，经常作为边缘设备被用于边缘计算和轻量级神经网络模型部署和测试。

图 5-1 树莓派 4B

Figure 5-1 Raspberry Pi 4B

到目前为止，树莓派已经更新迭代了许多不同的型号，通过认真对比和仔细分析，本文选用树莓派 4B(4G)作为实验设备，如图 5-1 所示。4B 为目前最新型号，相较于其他型号，在各式配置上都有大幅提升。树莓派 4B(4G)的 CPU 为 64 位，所使用的芯片为 BCM2711，主频为 1.5GHz，采用 ARM 架构体系的 Cortex-A72 架构，GPU 为 VideoCore VI，频率从 400MHz 提升到 500MHz，性能对比 3B+提升约为三倍。此外，树莓派 4B 还配备各式接口，包括：USB2.0、USB3.0、Micro HDMI、CSI 等，这些接口使得树莓派 4B 具备了强大的扩展性。同时其还支持 Rapsbian、Ubuntu、Win10-IoT 等多种系统，能满足各种不同的需求。树莓派 4B 详细硬件配

置如表 5-1 所示：

表 5-1 树莓派 4B(4G)详细配置

Table 5-1 Detailed configurations of the Raspberry Pi 4B(4G)

名称	型号
CPU	Broadcom BCM2711
GPU	VideoCore VI(500MHz)
内存	4GB DDR4
蓝牙	5.0
网络	以太网/无线网(802.11ac)
USB 端口	2×USB2.0/2×USB3.0
输入功率	5V,3A

5.1.2 英特尔二代神经计算棒简介

英特尔二代神经计算棒(Neural Compute Stick 2，NCS2)，由英特尔公司在 2018 年的人工智能大会上发布。其尺寸为 72.5mm×27mm×14mm，配置 USB 3.0 Type-A 接口与外部设备连接，同时 NCS 2 内置了最新的 Intel Movidius Myriad X VPU 视觉处理器，集成 16 个 SHAVE 计算核心、专用深度神经网络硬件加速器，可以以极低的功耗执行高性能视觉和 AI 推理运算，支持 TensorFlow、Caffe 开发框架，还兼容 64 位的 Ubuntu、CentOS、Windows 10 等操作系统。NCS2 的性能比之前的 Movidius 计算棒有了极大的提升，其中图像分类性能高出约 5 倍，物体检测性能则高出约 4 倍，如图 5-2 所示。

图 5-2 英特尔二代神经计算棒

Figure 5-2 Intel Neural Compute Stick 2

NCS2 体积较小，价格低，专门用于图像计算，性能高于传统的嵌入式设备，能起到取长补短的作用，利用计算棒可以在网络边缘构建更智能的 AI 算法和计算

机视觉原型设备。因此其主要定位就是用于物联网的设备端，代替原有设备进行深度学习的推理，常用在图像分类，目标检测等应用场景。在应用某些较为复杂的模型时，可以使用多个 NCS2 进行协同工作，或者将 NCS2 作为核心推理设备的加速组件，来加速模型的推理，实现更好的应用。

5.1.3 其他硬件简介

本文所采用摄像头型号为 HBV-RPI1509B-28 V11，尺寸为 25mm×24mm×23mm，像素 500w，感光芯片为 OV 5647，焦距 2.8mm，传感器像素为 1080p，可以通过 CSI 排线与树莓派连接，支持每秒 30 帧 1080p 或 60 帧 720p 视频录像。

树莓派使用内存卡型号为 SAMSUNG MicroSD 64GB，最高读速可达 100MB/s。同时使用外置蓝牙音箱来输出系统警示音。最后的系统实物图如图 5-3 所示。

图 5-3 系统实物图

Figure 5-3 System physical drawing

5.2 系统软件环境介绍

5.2.1 OpenVINO 简介

随着大数据时代的到来，许多深度学习模型需要部署到边缘场景下使用。为了快速响应边缘设备的检测动作，以及提高模型在边缘设备的推理速度，英特尔在 2018 年推出了计算机视觉开发框架 OpenVINO(Open Visual Inferencing and Neural Network Optimization)，用于快速部署应用和解决方案，实现边缘端模型加速，并适配多种用途。该框架支持英特尔 CPU 和英特尔神经计算棒，并兼容多种主流深度学习框架如：Caffe、Torch、Darknet、Tensorflow 等。

OpenVINO 的核心组件是深度学习部署工具包(Deep Learning Deployment Toolkit, DLDT)，DLDT 主要包括模型优化器和推理引擎两个部分。模型优化器的主要功能为将其它框架训练好的神经网络模型进行转换，转换为统一的自定义格式，并在转换过程中进行模型的优化，包括去掉模型推理不必要的层(Dropout)、对卷积层、BN 层和激活函数进行的融合加速、内存优化等。推理引擎则负责接收经过转换后的统一格式的神经网络模型，并提供高性能的神经网络推理运算，同时还支持多线程推理和异步推理加速，达到更高的效率。

5.2.2 OpenCV 简介

OpenCV 全称为 Open Source Computer Vision Library，是一个跨平台的计算机视觉处理开源软件库，支持在大多数系统上运行。OpenCV 轻量且高效，其主要由 c++实现，同时还提供了 Python、Ruby、Go 等语言的 API，实现了图像处理和计算机视觉方面的很多通用算法。OpenCV 丰富的 API 和极佳的移植性使得常被用于开发实时的图像处理、计算机视觉以及模式识别程序，并广泛应用于增强现实、物体识别、运动跟踪、人机交互等领域。

5.2.3 Flask 简介

Flask 是一个基于 Python 的轻量级 web 开发框架，对比其他同类型的框架，Flask 更为灵活、轻便且易上手。此外，Flask 框架的核心构成十分简单，只有基本服务，没有附带任何插件，但具有很强的扩展性和兼容性，开发者可以根据自身的需求安装额外的扩展包或自行开发实现不同的功能。除此之外，Flask 还支持安全的 cookie、适配 RESTFUL 以及以及包含开发服务器和调试器，是中小型网站和 Web 服务的开发首选。

5.2.4 其他环境简介

树莓派使用系统为 RapsberryPi OS(32 bit)，Python 版本为 3.7.3，Python 开发环境为 RapsberryPi OS 自带的轻量级开发环境 Thonny，浏览器为基于 webkit 内核的 Chromium 开源浏览器。

5.3 系统整体构建

整个吸烟手势检测系统由六个模块组成：模型训练、模型转换、图像采集、图像处理、吸烟手势检测和结果输出，结果输出则分为图像与警示声两种。整个系统的组成及各个模块之间的联系如图 5-4 所示。

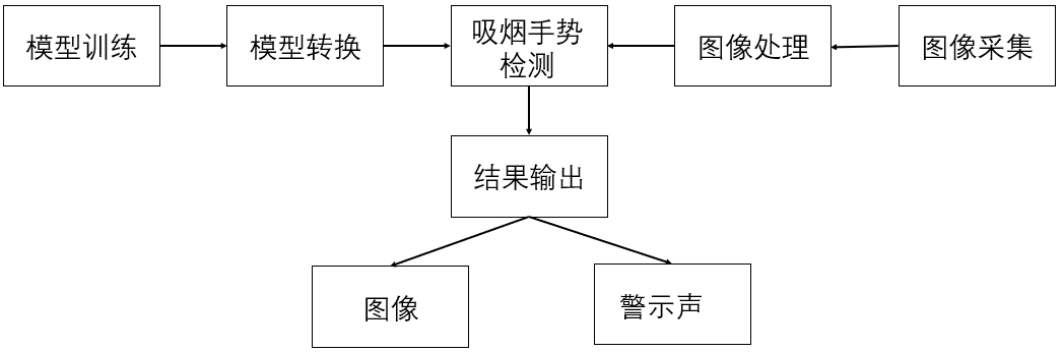

图 5-4 系统组成

Figure5-4 System composition

首先在服务器上完成吸烟手势识别算法 YOLOv7-tl 的模型训练。由于树莓派上的 OpenVINO 不支持模型转换功能，需要将训练得到的模型在 Windows 环境或 Linux 环境下转换成 OpenVINO 所需的自定义格式(IR)文件，并将转换后的文件传输到树莓派中。

将模型移植到树莓派后，其余的所有过程均在树莓派上实行，无需服务器端或 PC 端支持。吸烟手势检测系统工作流程如下：启动树莓派，通过 OpenCV 调用树莓派摄像头，实时采集图像数据，并对采集到的图像进行处理，包括将图像裁剪为统一尺寸 416×416，并将图像转换为像素值矩阵。接下来使用 OpenVINO 加载转换后的模型，驱动神经棒对图像进行推理，检测是否存在吸烟手势。定义置信度大于 0.5 时，即存在检测目标，因此当检测出吸烟手势的置信度大于 0.5 时，将目标所在位置标出，并显示 smoke 字样，同时输出此处禁止吸烟的警示音，并将最终检测结果实时展示到浏览器页面上。

5.4 系统效果测试

为了验证本文设计的吸烟手势检测系统的有效性，对设计的系统进行了实际效果测试。图 5-5 展示了系统调用摄像头进行检测的场景，测试了多个姿态同时包含小目标、遮挡等场景，并将摄像头的实时检测结果推送到 web 端进行实时监控，当检测到吸烟手势时，进行方框标记，并输出警示声。结果表明，设计的吸烟手势检测系统表现优异，能基本满足实际需求。

(a) (b)

(c) (d)

图 5-5 树莓派吸烟手势检测系统效果测试

Figure 5-5 Test of Raspberry Pi Smoking gesture detection system

5.5 本章小结

本章主要介绍了吸烟手势检测系统的设计与实验。本文不仅仅是在算法上进行研究，更结合实际需求，将吸烟手势检测算法部署到树莓派端。对系统搭建的硬件环境和软件环境进行了介绍，主要包括：树莓派、英特尔二代神经计算棒、OpenVINO、OpenCV 和 Flask 等。然后对系统的整体流程进行了说明，最后在树莓派上进行实验，测试了所构建的系统，实验结果显示，本文的吸烟手势检测系统效果优异，能快速、检测出吸烟手势，满足实际需求。

第 6 章 总结与展望

6.1 主要研究成果

本文利用人工智能深度学习领域的相关理论，对吸烟行为识别领域进行了深入研究，提出了基于深度学习的吸烟手势检测方法，改进了 YOLOv7-tiny 模型，解决了传统检测方式消耗资源多且检测不准确的问题，同时规避了检测香烟本身和香烟烟雾的弊端。最后研发了吸烟手势检测系统，部署在树莓派上，实现了吸烟手势的快速、准确检测，具有一定实际意义。主要研究成果如下：

（1）吸烟手势数据集的建立。目前相关领域没有一个权威的、公认的数据集，因此通过收集和清洗，制作了一个适用于本文算法模型训练和测试的吸烟手势数据集，弥补了这方面的不足。

（2）轻量级吸烟手势检测算法 YOLOv7-tl 的实现。本文研究是以工程实际价值为导向，需要将算法部署到树莓派上，算力资源受到限制。因此，在轻量级目标检测算法 YOLOv7-tiny 的基础上进行改进，通过使用 Ghost 模块和 Ghost 瓶颈结构对骨干网络和网络的部分模块进行改造，并对特征融合部分 PANet 结构进行裁剪，去掉负责检测大目标的分支，使得整个模型更轻量。并引入 CARAFE 上采样算子、ECA 通道注意力模块、K-medoids 聚类算法和 Alpha-IoU 损失函数，有效提升了模型学习特征和实现目标准确定位的能力，提高了模型精度。因此，改进后得到的 YOLOv7-tl 算法实现了精度和速度两方面的平衡，更适合部署在树莓派上。

（3）吸烟手势检测系统的实现。在树莓派上部署模型，结合实际应用需求，实现了一个吸烟手势检测系统，并采用英特尔二代神经计算棒和高性能网络计算框架 OpenVINO 进行加速，提高吸烟手势检测系统的推理效率。该系统支持主流图像和视频格式，同时构建了一个 web 页面，可以将摄像头检测结果实时展示到浏览器页面上，为问题的高效解决与处理提供了解决方案。

6.2 未来工作展望

本文提出基于深度学习的吸烟手势目标检测算法，并部署在树莓派上进行检测，取得了一定的效果，但仍存在一些不足：

（1）本文使用的数据集整体数据量偏少，尽管已使用数据增强和迁移学习来弥补一些不足，后续仍需继续收集或拍摄采集更多的、包含各种场景的图像来扩充数据集。

（2）本文设计的 YOLOv7-tl 算法尽管取得了不错的效果，实现了精度和速度上的平衡，但模型精度只达到了 92.6%，仍有较大的上升空间，需要在保持网络轻

量的同时进一步对网络进行研究改进，提高精度。

总体来说，研究方向与路线是正确的，未来需要在此方向上进一步研究。

参 考 文 献

[1] Thakur S S, Poddar P, Roy R B. Real-time prediction of smoking activity using machine learning based multi-class classification model[J]. Multimedia Tools and Applications, 2022, 81(10): 14529-14551.

[2] World Health Organization.Progress in the fight against tobacco epidemic.[EB/OL].(2021-07-27)[2023-03-22].https://www.who.int/news/item/27-07-2021-who-reports-progress-in-the-fight-against-tobacco-epidemic

[3] Xu X, Bishop E E, Kennedy S M, et al. Annual healthcare spending attributable to cigarette smoking: an update[J]. American journal of preventive medicine, 2015, 48(3): 326-333.

[4] US Department of Health and Human Services. The health consequences of smoking—50 years of progress: a report of the Surgeon General[J]. 2014.

[5] Gallus S, Lugo A, Liu X, et al. Who smokes in Europe? Data from 12 European countries in the TackSHS survey (2017–2018)[J]. Journal of epidemiology, 2021, 31(2): 145-151.

[6] 国家卫生健康委.中国吸烟危害健康公告 2020[EB/OL].(2021-05-28)[2023-03-22].http://www.nhc.gov.cn/guihuaxxs/s7788/202105/c1c6d17275d94de5a349e379bd755bf1.shtml

[7] 公共场所卫生管理条例实施细则[J].司法业务文选,2011(16):33-40.

[8] 万里波. 基于深度学习的吸烟行为检测系统研究[D].电子科技大学,2022.

[9] 孙召龙,徐昕,朱云龙,田枫.基于 YOLOv5 的油田作业现场吸烟检测方法[J].系统仿真技术,2021,17(02):89-93.

[10] 马晓菲. 基于人脸分析的公共场所吸烟行为检测系统研究[D].燕山大学, 2020.

[11] 张洋,姚登峰,江铭虎,李凡姝.基于 EfficientDet 网络的细粒度吸烟行为识别[J].计算机工程,2022,48(03):302-309+314.

[12] 姜晓凤,王保栋,夏英杰,李金屏.基于人体关键点和 YOLOv4 的吸烟行为检测[J].陕西师范大学学报(自然科学版),2022,50(03):96-103.

[13] 王鹏,尹勇,宋策.基于改进 RetinaFace 和 YOLOv4 的船舶驾驶员吸烟和打电话行为检测[J].上海海事大学学报,2022,43(04):44-50.

[14] Cole C A, Janos B, Anshari D, et al. Recognition of smoking gesture using smart watch technology[J]. arXiv preprint arXiv:2003.02735, 2020.

[15] Maguire G, Chen H, Schnall R, et al. Smoking Cessation System for Preemptive Smoking Detection[J]. IEEE Internet of Things Journal, 2021.

[16] Añazco E V, Lopez P R, Lee S, et al. Smoking activity recognition using a single wrist IMU and deep learning light[A]. In: Proceedings of the 2nd international conference on digital signal processing[C]. 2018:48-51.

[17] Alharbi F, Farrahi K. A convolutional neural network for smoking activity recognition[A]. In: 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)[C].IEEE, 2018:1-6.

[18] Odhiambo C O, Cole C A, Torkjazi A, et al. State transition modeling of the smoking behavior using lstm recurrent neural networks[A]. In: 2019 International Conference on Computational Science and Computational Intelligence (CSCI)[C].IEEE,2019:898-904.
[19] Zhang D, Jiao C, Wang S. Smoking Image Detection Based on Convolutional Neural Networks[A]. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC)[C]. IEEE,2018:1509-1515.
[20] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[A]. In: Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2015:1-9.
[21] Wei Z, Zhu Y X, Li Q X, et al. Improved Smoking Target Detection Algorithm Based On YOLOv3[A]. In: Journal of Physics: Conference Series[C]. IOP Publishing,2021,1883(1):012052.
[22] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
[23] Zhang X, Su X, Yu J, et al. Combine Object Detection with Skeleton-Based Action Recognition to Detect Smoking Behavior[A]. In: 2021 The 5th International Conference on Video and Image Processing[C]. 2021:111-116.
[24] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[A]. In: Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2016:779-788.
[25] Hameed H, Azam N, Usman M, et al. RF Sensing for Smoking Detection at Oil Fields[A]. In: 2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)[C].IEEE, 2022:944-945.
[26] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[A]. In: Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2016:2818-2826.
[27] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
[28] 王彦生,曹雪虹,焦良葆,孙宏伟,高阳.基于改进 YOLOv5 的电厂人员吸烟检测[J/OL].计算机测量与控制:1-13[2023-02-23].
[29] Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[A]. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition[C]. 2020:11534-11542.
[30] Gevorgyan Z. SIoU loss: More powerful learning for bounding box regression[J]. arXiv preprint arXiv:2205.12740, 2022.
[31] Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection[A]. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition[C]. 2020:10781-10790.
[32] Fukushima K. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position[J]. Biol, Cybern, 1980, 36: 193/202.

[33] LeCun Y, Boser B, Denker J, et al. Handwritten digit recognition with a back-propagation network[J]. Advances in neural information processing systems, 1989, 2.
[34] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324
[35] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.
[36] 张港. 基于深度学习轻量化卷积神经网络的遥感图像场景分类研究[D].南京邮电大学,2022.
[37] 刘雅兰.基于深度种子和局部图匹配的细胞追踪算法研究[D].湖南大学,2020.
[38] 李富豪. 基于卷积神经网络的鼻腔鼻窦肿瘤图像分割[D].青岛大学,2022.
[39] 牛黎莎.基于深度迁移学习的心电异常检测算法研究[D].齐鲁工业大学，2021.
[40] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[A]. In: Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2014:580-587.
[41] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(9): 1904-1916.
[42] Girshick R. Fast r-cnn[A]. In: Proceedings of the IEEE international conference on computer vision[C]. 2015:1440-1448.
[43] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28: 91-99.
[44] Dai J, Li Y, He K, et al. R-fcn: Object detection via region-based fully convolutional networks[J]. Advances in neural information processing systems, 2016, 29.
[45] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[A]. In: Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2017:7263-7271.
[46] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[A]. In: European conference on computer vision[C]. Springer,Cham,2016:21-37.
[47] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[A]. In: Proceedings of the IEEE international conference on computer vision[C]. 2017:2980-2988.
[48] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[A]. In: Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2016:770-778.
[49] 高锦风,陈玉,魏永明,李剑南.基于改进的 YOLOv3 和 Facenet 的无人机影像人脸识别[J].中国科学院大学学报,2023,40(01):93-100.
[50] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[51] Wang C Y, Liao H Y M, Wu Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[A]. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops[C]. 2020:390-391.
[52] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new st-ate-of-the-art for real-time object detectors[J]. arXiv preprint arXiv:2207.02696, 2022.
[53] Zheng J, Wu H, Zhang H, et al. Insulator-Defect Detection Algorithm Based on Improved YOLOv7[J]. Sensors, 2022, 22(22): 8801.
[54] 田佳敏. 基于半监督迁移学习的热轧钢带表面缺陷检测方法研究[D].西京学院,2022.
[55] 王紫腾. 基于深度迁移学习与多特征网络融合的高分辨率遥感图像分类[D].南京邮电大学,2022.
[56] 芦奕霏. 基于深度学习的轴承故障诊断方法研究[D].南京邮电大学,2022.
[57] Wu J, Wu Y, Niu N, et al. MHCPDP: multi-source heterogeneous cross-project defect prediction via multi-source transfer learning and autoencoder[J]. Software Quality Journal, 2021, 29(2): 405-430.
[58] S HAN K, WANG Y, TIAN Q, et al. Ghostnet: More features from cheap operations[A]. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition[C]. 2020:1580-1589.
[59] Wang J, Chen K, Xu R, et al. Carafe: Content-aware reassembly of features[A]. In: Proceedings of the IEEE/CVF international conference on computer vision[C]. 2019:3007-3016.
[60] Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[A]. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition[C]. 2020:11534-11542.
[61] HE J, ERFANI S, MA X, et al. Alpha -IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression[J]. Advances in Neural Information Processing Systems, 2021, 34: 20230-20242.
[62] ZHENG Z, WANG P, LIU W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[A]. In: Proceedings of the AAAI Conference on Artificial Intelligence[C]. 2020,34(07):12993-13000.
[63] 刘征.基于深度学习的机载红外场景分类算法研究[D].中国电子科技集团公司电子科学研究院,2022.
[64] 沈翔.基于 RetinaNet 的小目标检测提升方法研究[D].南京邮电大学,2022.
[65] 王军,冯孙铖,程勇.深度学习的轻量化神经网络结构研究综述[J].计算机工程,2021,47(08):1-13.
[66] 王立辉. 基于卷积神经网络的行人检测与跟踪算法研究[D].武汉科技大学,2021.
[67] 严继伟,苏娟,李义红.基于 Ghost 卷积与注意力机制的 SAR 图像建筑物检测算法[J].兵工学报,2022,43(07):1667-1675.
[68] 程曼茹.基于多特征时空图网络的智慧城市交通预测研究[D].南京邮电大学,2022.

[69] 韩兴,张红英,张媛媛.基于高效通道注意力网络的人脸表情识别[J].传感器与微系统,2021,40(01):118-121.
[70] Liu L, Liu Y, Yan J, et al. Object Detection in Large-Scale Remote Sensing Images With a Distributed Deep Learning Framework[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2022, 15: 8142-8154.
[71] 李鸿.基于轻量化网络的目标检测算法研究[D].中国科学院大学(中国科学院光电技术研究所),2022.
[72] 程世聪.基于树莓派图像处理的冷库除霜控制方法的实验研究[D].天津商业大学,2022.