

摘要

车牌号码作为车辆的身份证，交通量采集系统作为智能交通系统和车辆信息管理的重要组成部分，因此车牌的检测与识别是智慧交通系统中不可或缺的环节，对于改善交通环境以及管理车辆信息至关重要。

本文利用深度学习技术，采用检测加识别的思路，通过网上下载数据集 CCPD2019 来获取。使用 OpenCV 的双线性插值方法调整图像尺寸。剔除重复及模糊图像，避免模型出现过拟合现象。数据集按照 9: 1 进行划分，使用 LabelImg 对数据集进行标注，最终构建稳定有效的数据集。

将目标检测模型 YOLOv5 作为检测器对车牌进行定位，利用 LPRnet 算法对强光、弱光或光照不均匀的复杂环境下的车牌进行识别，以此提高车牌识别的准确率。本文基于 YOLOv5 + LPRnet 实现了车牌定位及识别，该系统对提升交通道路安全有重要的现实意义。

关键词：目标检测，识别，YOLOv5，LPRnet，

目录

第一章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 传统目标算法.....	2
1.2.2 基于深度学习的目标检测.....	4
1.3 目前存在的问题.....	5
1.4 论文主要研究内容.....	6
第二章 车牌检测技术原理.....	8
2.1 卷积网络.....	8
2.2 目标测试方法研究.....	8
2.3 YOLOv5 算法结构.....	9
2.3.1 输入端.....	10
2.3.2 主干网络.....	10
2.3.3 Neck 网络.....	12
2.3.4 输出端.....	13
2.4 目标检测算法.....	14
2.4.1 目标检测评价指标.....	15
2.4.2 目标检测算法选择.....	16
2.5 文章小节.....	21
第三章 车牌识别技术算法.....	22
3.1 LPRNet 模型.....	22
3.1.1 相关研究.....	22
3.1.2 LPRNet 结构.....	23
3.1.3 LPRNet 的骨干网络.....	23
3.2 训练步骤:.....	25
3.3 实验结果与分析.....	26
3.3.1 数据集获取与处理.....	26
3.3.2 系统性能与评价指标.....	26
3.3.3 性能评估.....	27
3.4 本章小节.....	29
第四章 车牌识别系统设计与实现.....	30
4.1 系统可行性分析.....	30
4.1.1 需求分析.....	30
4.1.2 系统可行性分析.....	30
4.1.3 系统流程分析.....	31
4.1.4 系统架构设计.....	32
4.2 模块设计.....	33
4.2.1 各模块功能.....	33
4.2.2 模块功能实现.....	34
4.3 系统测试与效果分析.....	34
4.3.1 系统功能测试.....	34
4.3.2 系统检测和定位效果分析.....	36

4.4 本章小结.....	37
第五章 总结与展望.....	38
5.1 总结.....	38
5.2 不足与展望.....	38
引用文献.....	40
致谢.....	45

第一章 绪论

1.1 研究背景及意义

随着城市化进程的加快和车辆数量的激增，智能交通系统（ITS）变得越来越重要。车牌识别是 ITS 中的一项关键技术，用于车辆管理、交通流量监控、违章记录等，通过计算机视觉和深度学习技术，车牌检测与识别实现了对车辆牌照信息的自动识别，不仅提高了交通管理的效率和准确性，减少了人力成本，而且促进了交通安全和智慧城市的发展，对于缓解交通拥堵、提升城市管理水平起到了至关重要的作用。可是传统的车牌识别方法依赖于手工特征提取和传统的图像处理技术，这些方法在复杂环境下（如光线变化、车牌污损、角度变化等）表现不佳。近年来，深度学习特别是卷积神经网络（CNN）在图像识别领域取得了突破性的进展。神经网络能够自动学习特征，适应复杂的环境变化，提高了识别率和鲁棒性。对此，计算能力的提升和大数据的可用性使得训练复杂的神经网络模型成为可能，这为车牌识别技术的发展提供了坚实的技术基础。

通过自动识别车牌，可以实现车辆的快速识别和记录，减少人工干预，提高交通管理的效率。车牌识别技术可以用于电子警察系统，帮助快速识别和追踪违章车辆，提高道路安全。而在停车场中，车牌识别可以自动记录车辆进出信息，减少人工收费和管理的工作量。作为智能城市的一部分，车牌识别技术可以与其他系统（如环境监测、交通信号控制等）集成，提供更全面的城市管理解决方案。在商业应用上，高速公路收费系统、停车场管理、车辆租赁等商业领域都可以通过车牌识别技术提高服务质量和效率。不仅如此，研究车牌识别技术推动了计算机视觉、模式识别、机器学习等领域的技术进步，促进了相关技术的跨领域应用。

总体来说，基于神经网络的车牌检测与识别技术不仅在实际应用中具有广泛的意义，而且在推动人工智能和计算机视觉技术的发展方面也起到了重要作用。

1.2 国内外研究现状

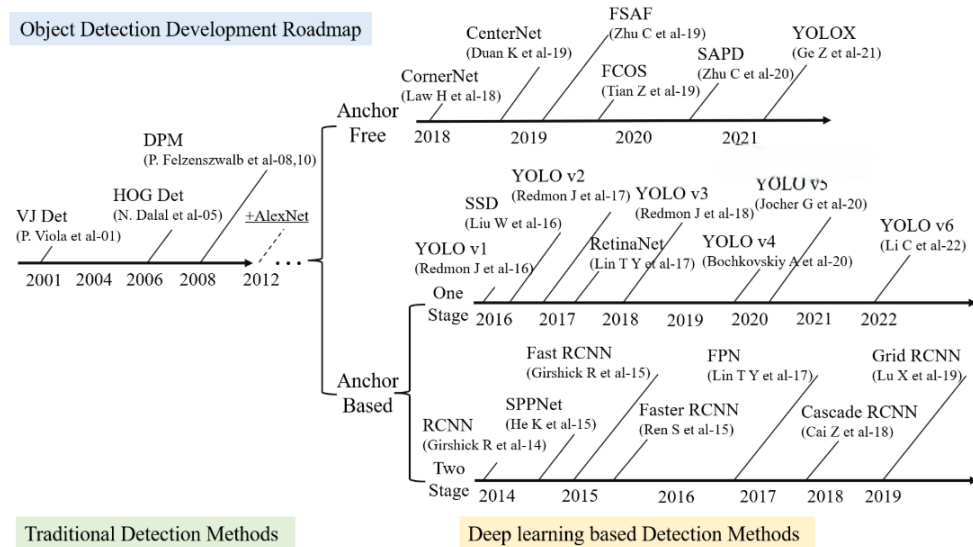


图 1.1 目标检测算法近 20 年来发展路线图

目标检测国内外研究现状分为两个部分，如图 1.1 所示。2012 年以前以传统目标检测为主，主要的代表算法有 VJ 检测器^[1]、HOG 检测器^[2]、DPM 检测器^[3]、Haar 特征^[4]、LBP 特征^[5]等。2012 年之后基于深度学习的目标检测算法成为研究焦点，其代表算法有 C RNN^[6]、SPPNet^[7]、YOLO^[8]、FestRCNN^[9]、SSD^[10]、Retina-Net^[11]等，传统目标检测由于缺乏图像特征表示方法且建立在人工整理的基础上，目标检测模型较为简单。研究者通过设计复杂的特征表示和加速技巧来弥补当时的资源不足问题。

1.2.1 传统目标算法

传统目标检测算法框架主要分为五个步骤，如图 1.2 所示。输入图像对目标物体进行特征提取，目标物体的特征经过分类器得到各个类别的置信度分数，根据置信度分数高低确定目标定位结果^[12]：

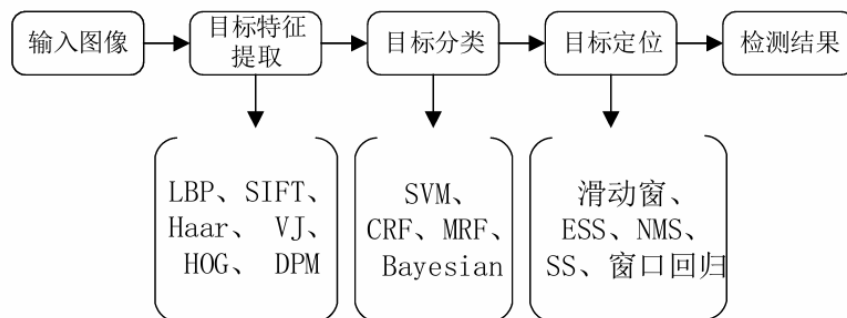


图 1.2 传统目标检测算法框架

特征提取是把图像中像素级数据转变为目标物体的特征描述子。LBP(局部二值模式, Local Binary Pattern)、SIFT^[13] (尺度不变特征变换, Scale-Invariant Feature Transform)、Haar 特征等都是局部领域特征描述子的代表。LBP 描述子是将其定义在一个 3*3 的区域内, 中心像素的灰度值是阈值, 剩余区域的灰度值分别与中心区域的阈值作比较, 若周边区域像素灰度值大于阈值, 则标记为 1, 否则标记为 0, 顺时针记录八位二进制数字串, 将二进制数字串转换为十进制所得结果便是中心像素灰度值^[5]。LBP 算法思想, 如图 1.3 所示。其特点是具有旋转不变性和灰度不变性, 人脸识别中就有不少是采用 LBP 描述子完成的。SIFT 描述子每一个关键点都拥有位置、尺度和方向三个信息, 用向量描述关键点, 使关键点不随光照和视角的改变而改变且具有唯一性^[13]。Haar 描述子是将所有黑色像素值的总和减去所有白色像素值的总和, 再引入积分图计算。

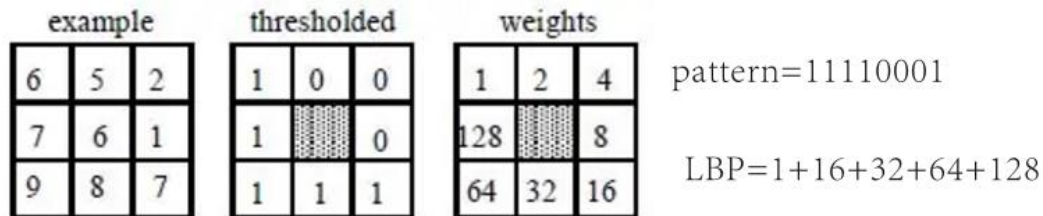


图 1.3 LBP 算法思想

目标分类是将目标物体的特征处理后得到类别置信度分数。分类器基本可以分为两种: 概率模型和非概率模型。基于概率模型的代表有 Bayesian 框架、条件随机场 (Conditional Random Field, CRF)、马尔可夫随机场 (Markov Random Field, MRF) 等, 该方法的求解过程相对较慢且需要大量的数据进行训练^[14]。非概率模型方法可以通过数据训练直接得到输入输出的关系, 最常使用的方法是支持向量机 (Support Vector Machine, SVM)。

目标定位是在输入图像中寻找与模型最佳的匹配区域。早期是通过滑动窗口尽可能列举出图像上检测目标的锚框, 通过滑动窗口去遍历目标物体的各个位置, 效率较为低下。目前高效子窗口搜索 (Efficient Subwindow Search, ESS)、选择性搜索 (Selective Search, SS)、非极大抑制 (Non-Maximum Suppression, NMS) 和窗口回归 (Bounding Box Regression) 在检测速度上都有明显的提升^[15]。

Paul Viola 和 Michael Jones 在 2001 年发表了一种目标检测方法^[6]。该算法使用了 Haar 特征进行目标物体的特征提取, 分为三个关键步骤, 第一步是引入“积分图”, 可以快速获得目标所需特征。第二步是基于 AdaBoost 的学习算法, 它能从大量的视觉特征中选取少量关键特征, 从而得到较高的分类效果^[16]。第三步是一种“级联”结构的方法, 该方法能快速丢弃背景的同时加速目标检测。该方法丢弃了不感兴趣的背景对象, 却提供了统计保证, 能够快速处理图像且获得较高的检测率^[6]。

HOG^[11](Histogram of Oriented Gradients)检测器于 2005 年提出, 通过统计图像特定边缘的有无和不同边缘特征的组合检测目标, 边缘特征通过像素的梯度信息来描述, 包括梯

度的大小和方向^[17]。在尺度特征不变性(Scale Invariant Feature Transform)和形状上下文(Shape Contexts)的改进中,注重平衡特征的不变性和非线性特性,通过在均匀间隔的密集网格上计算重叠的局部对比度归一化,以实现更精确的检测准确性^[18]。该算法以本地像素块为基础进行特征直方图提取,在目标局部变形和受光照影响下仍能表现出卓越的稳定性,为计算机视觉领域带来了新的可能性^[11]。

DPM(Deformable Parts Model)属于目标检测传统算法,在 2008 年提出。与传统的 VJ 检测器和 HOG 检测器相比,DPM 进行了延伸与改进。该模型由主过滤器(Root filter)和多个辅过滤器(Part-filters)构成,为了提高检测精度,DPM 采用了硬负挖掘(Hard negative mining),边框回归(Bounding box regression)和上下文启动(Context priming)等策略^[18]。其优势在于成为传统目标检测算法的 SOTA (State Of The Art)、运算速度快、且能够适应物体形变。DPM 的局限性在于无法适应大幅度的旋转,导致在处理此类情况下的稳定性较差^[8]。

由于传统目标检测算法一方面需要手工提取特征,另一方面硬件跟不上,导致传统目标检测算法的识别准确率不高。还需要遍历各个包含物体的区域,导致计算量较大,计算速度较慢,甚至会产生多个正确的识别效果。

1.2.2 基于深度学习的目标检测

在 2012 年的 ImageNet 竞赛中,AlexNet 特征提取算法取得了重大成就^[19]。迅速将 CNN 引入到目标检测领域,但这种方法准确率不尽人意。R-CNN 是 2013 年提出的,其基于区域两阶段的框架,但这种两阶段检测算法存在两个大问题,首先 Selective Search 并不是卷积,因为它不是端到端可训练的。其次是速度问题,在每个 Region Proposal 上分别运行会使其更慢。R-CNN 的一个快速后续是减少对多个 Region Proposals 的重复卷积,通过对整个图像运行一次 CNN 并在多个 Region Proposals 之间共享计算,来改进目标检测^[20]。不同的 Region Proposals 具有不同的大小,这可能导致在计算特征时出现不一致的尺度问题。

2015 年, Fast R-CNN 算法被提出,此种算法速度非常快,这使它成为工业界的热门选择^[14]。2015 年 Faster R-CNN 提出,创建 RPN 网络^[20],标志着目标检测深度学习方法的一个里程碑,如图 1.3 所示。随后提出了 Fast R-CNN、Fester R-CNN、Mask R-CNN^[21]和 SPPNet,都是基于 R-CNN 的两阶段目标检测算法。在一定程度上提升了运行速度、检测速度和检测精度,为计算机视觉领域带来了重要突破。He 等人在 2014 年提出了 SPP-Net, SPP-Net 是在 R-CNN 的基础上引入了一种直接对全图进行特征提取的策略^[22]。虽然 R-CNN 在目标检测上的准确率不错,但检测方法较为麻烦。为了更好地解决这个问题,YOLO 系列算法被提出,相对于 Two-stage 目标检测算法而言,一阶段目标检测算法牺牲了一些准确率,但是检测速度明显提升。

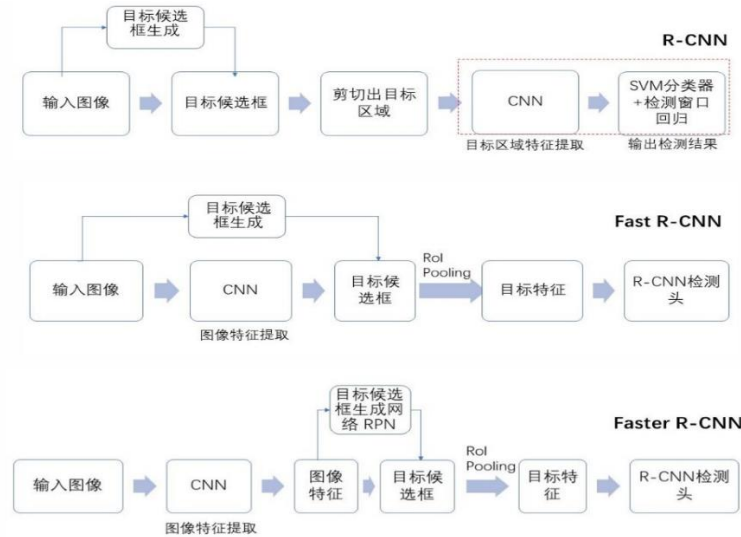


图 1.4 基于深度网络的两阶段目标检测算法流程

YOLO 系列是一阶段目标算法，使用多尺度特征图进行目标预测。YOLOv1^[13]是由 Redmon 等人在 2016 年提出，第二年，他们提出了 YOLOv2^[23]算法 2018 年发布了 YOLOv3^[24]算法，随着算法的不断改进，检测的性能也不可同日而语。Yolo 算法采用一个单独的 CNN 模型实现 end-to-end 的目标检测，将整个图像输入网络，并在最后回归物体的位置、检测框和类别。将图像分为 $S \times S$ 的网格，每个网格负责预测中心落在该网格的物体，主要预测 B 个边界框、网格所属类别 C 与对应置信度。每个网格预测的类别信息和置信度相乘，表示网格所属类别与置信度。最后通过设定的阈值过滤掉得分较低的检测框，最后用非极大值抑制算法（non maximum suppression, NMS）得到最终的检测结果

Liu 等人提出单阶段目标检测算法 SSD^[15]，与 YOLO 不同，它直接使用卷积神经网络输出的特征图进行目标检测，而 YOLO 是在全连接层之后检测。此外，SSD 在不同尺度的特征图上直接进行目标检测，利用富含细节特征的大尺度特征图来检测小物体，使用包含更多全局语义信息的小尺度特征图来检测大物体。这种多尺度的设计使得 SSD 对不同大小的目标都具有良好的检测能力。Lin 等在 2017 年提出 RetinaNet^[16]，通过分析候选区域包含潜在目标的概率，来校正最终的训练损失，有效地解决了潜在的类别不平衡问题。

1.3 目前存在的问题

(1)目标准确大：模型的推理速度无法满足实时要求。这可能会导致在处理视频流或其他需要实时响应的场景中出现延迟。

(2)自适应性低，通常目标检测算法是在特定场景下训练，才对类似场景下的图像检测效果较好，因此自适应较差，当出现天气变化、光照变化、尺度变化、遮挡重叠时，造成识别难度较大，漏检误检概率较高。

1.4 论文主要研究内容

本文主要对车牌进行检测定位及识别，考虑到传统目标检测算法模型参数量庞大，影响模型的实时性效果。本文将目标检测算法模型进行轻量化处理，确保目标检测算法模型在检测性能不降低且保证实时性的前提下提高模型的精确度，具体技术路线如图 1.5 所示：

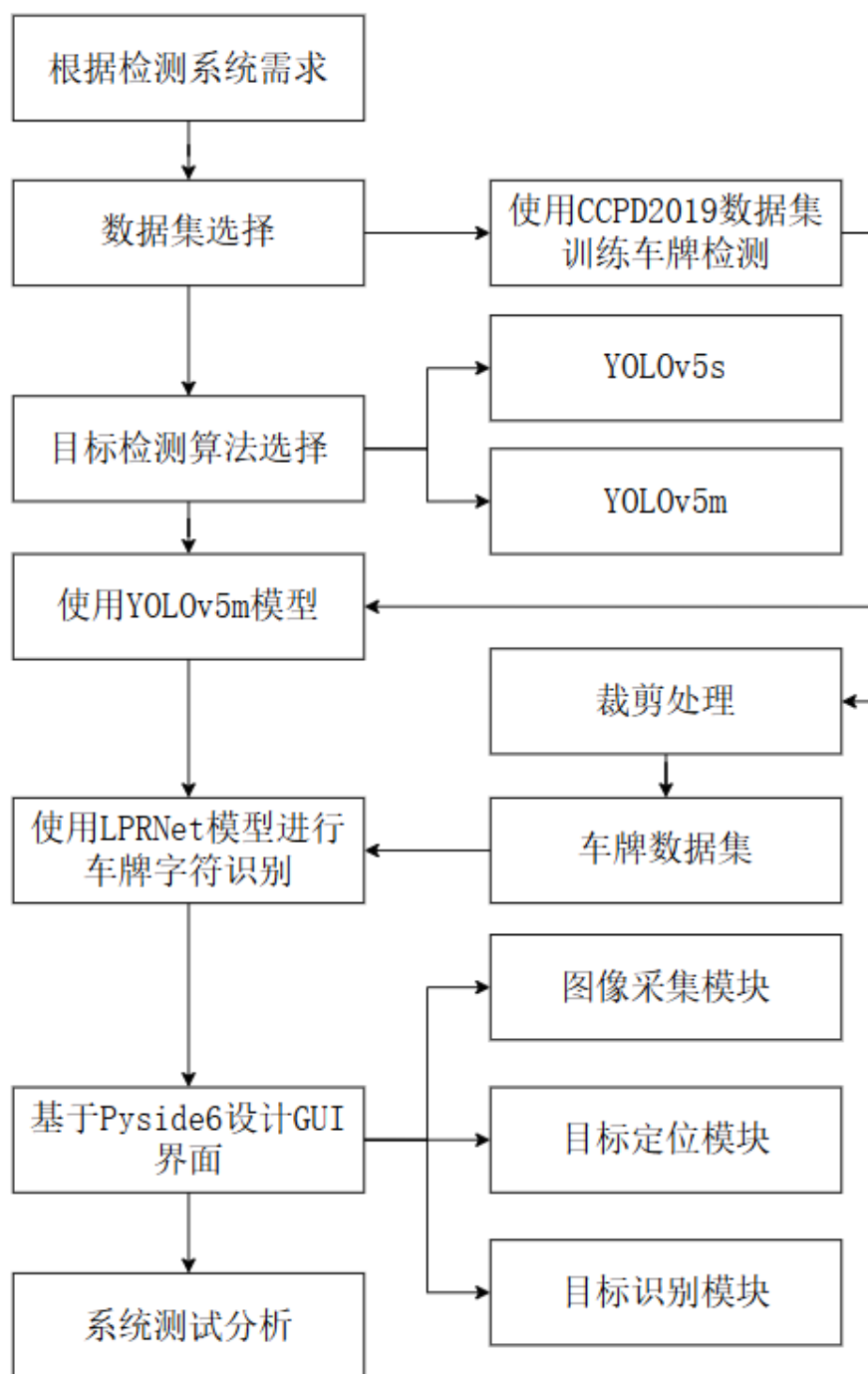


图 1.5 车牌检测与识别系统流程图

(1) 训练数据集:

针对数据集选择的问题, 本文使用了 CCPD2019 数据集来进行对 YOLO 的训练, 并且对其中部分数据进行裁剪车牌来对 LPRnet 算法进行训练, 两部分的数据集以 9: 1 划分为训练集与验证集, 使用 LabelImg 和基于深度学习的半自动标注相结合的方式进行数据集标注, 提高标注效率。

(2) 搭建车牌检测算法模型:

针对车牌检测问题, 由于设备性能条件和实时性以及检测效果的需求, 本文首先使用 YOLOv5s 和 YOLOv5m 算法模型, 并利用预处理后的 CCPD2019 数据集进行训练, 通过基本指标对比选用 YOLOv5m 作为目标检测算法。为满足本文对车牌的检测研究。

(3) 搭建车牌字符识别算法模型:

YOLOv5m 算法模型具有高效的特征提取和目标定位能力, 在车牌区域检测任务中, 该模型可以准确地定位并框选出车牌区域。为实现车牌区域的字符识别, 接着引入 LPRnet 模型, LPRNet 由轻量级的卷积神经网络组成, 它可以采用端到端的方法来进行训练。针对车牌倾斜和模型参数量大的问题, 对 LPRnet 模型进行了空间变换网络和门控循环单元优化, 对比 C-RNN 在识别效率上有所提高。

(4) 搭建车牌识别检测系统

基于 YOLOv5m 的车牌识别检测系统设计: 从系统总体设计, 模块设计, 系统测试与效果分析三大部分将前几章算法进行整合, 并使用 Pyside6 设计 GUI 软件界面, 并在不同场景下进行识别准确率的测试和效果分析。实验证实该系统具有较好的识别效果, 能够准确地检测与识别车牌字符。

第二章 车牌检测技术原理

2.1 卷积网络

机器学习利用计算机、概率论、统计学专业知识，在不断地计算中让计算机学习新的知识。作为机器学习的一个重要研究方向，深度学习学习的是数据的内在规律并将学习到的信息用于解释语言、图像、声音等数据。这些深度学习的应用形式对应着自然语言处理、计算机视觉、语音识别等深度学习领域。卷积神经网络是网络层次较深，主要依靠卷积计算的神经网络。在卷积神经网络之前，图像处理的主要研究都集中于特征的提取与分类上，研究人员提出了许多实用性强的分类器如：支持向量机(自适应增强(Adaptive Boosting)、随机森林(Support Vector Machine Random Forest))等，传统分类器往往是手动设计的，其存在泛用性差，分类准确率较低等缺点，在深度学习发展的过程中，卷积神经网络因其较高的图像处理准确率等优点收到广泛关注。卷积神经网络是一个层次模型，包括由输入层、卷积层、激活层、池化层构成的特征提取器和由全连接层、输出层构成的分类器。在计算机视觉领域，对图像进行识别的任务大量使用卷积神经网络，用于一阶段目标检测的YOLO模型就是经典的卷积神经网络模型。卷积神经网络可以自提取目标特征，发现样本集中规律。卷积神经网络(CNN)的计算主要有两种方式：前向传播与反向传播。前向传播中神经网络每一层的输出都作为下一层的输入并逐层计算，最终输出结果。反向传播作用于神经网络的输出，通过计算梯度由深到浅的更新网络参数。

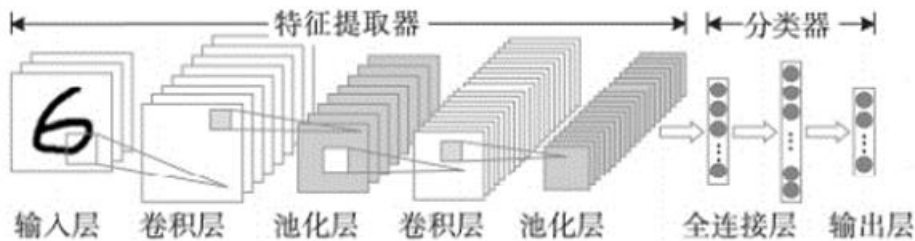


图 2.1 卷积神经网络示意图^[25]

前向传播中，假设给出某结点 w 存在 i,j,k 等上层结点，结点 w 的值由上层结点输出及对应权值进行加权运算求和最终加入偏置值，将所求结果通过一非线性激活函数进行函数值转化即为最终所求结点 w 的值。通过这一方法层层运算最终得到输出层结果。反向传播算法又被称为 BP 算法，为减小误差，通过梯度下降法对网络的权值与偏置值进行更新。在反向传播算法中，基于初始权重、偏置值等参数进行一次前向传播，并计算模型各结点的值，再通过反向传播过程由输出反向倒推，逐步更新每层的权重与偏置值，并随计算更新神经网络中各参数的值。

2.2 目标测试方法研究

目标检测定位是本文的主要研究内容之一，目前大多数采用人工神经网络。人工神经

网络是一种仿生学概念，模仿人的神经系统构造出的一种网络结构，推动着人工智能的发展。卷积神经网络（Convolutional Neural Networks, CNN）作为神经网络中一个重要分支，已经在很多领域大放异彩，取得一定程度的发展，被广泛应用于语音识别领域以及图像识别领域。近些年科技的进步是十分巨大的，这也伴随着一些改变，比如信息获取会更加方便，图像获得更加便捷，数据容量更大，目标检测的方法逐渐向深度学习方向偏移，与其他神经网络不同，CNN 的主要特征是进行卷积计算，并且擅长处理图像，基于深度学习的目标检测和目标分类在当前阶段均是由卷积计算实现，随着人工智能的不断发展，更多的目标检测算法被提出。其中 YOLO 系列算法更是近几年的后起之秀，在目标检测领域大放异彩，5 年的时间内，YOLO 算法已经发展到 YOLOv5。

2.3 YOLOv5 算法结构

YOLOv5 网络架构划分为：输入、主干、颈部和头部。假定输入图像大小为 $640 \times 640 \times 3$ ，其中 3 表示 RGB 三个颜色通道。特征提取后，该模型生成三个不同尺度的特征图： $20 \times 20 \times 225$ 、 $40 \times 40 \times 225$ 和 $80 \times 80 \times 225$ 。YOLOv5 中的 CSP1 和 CSP2 层控制模型的深度。CSP1_X 和 CSP2_X 层中有 X 个残差块，其中 X 值因不同的 YOLOv5 模型而异。X 值越大，网络模型越深，越能提升特征提取和融合能力。如 YOLOv5s 中，使用了 CSP1_1、CSP1_3 和 CSP2_1，YOLOv5s 网络结构如图 4.2 所示。

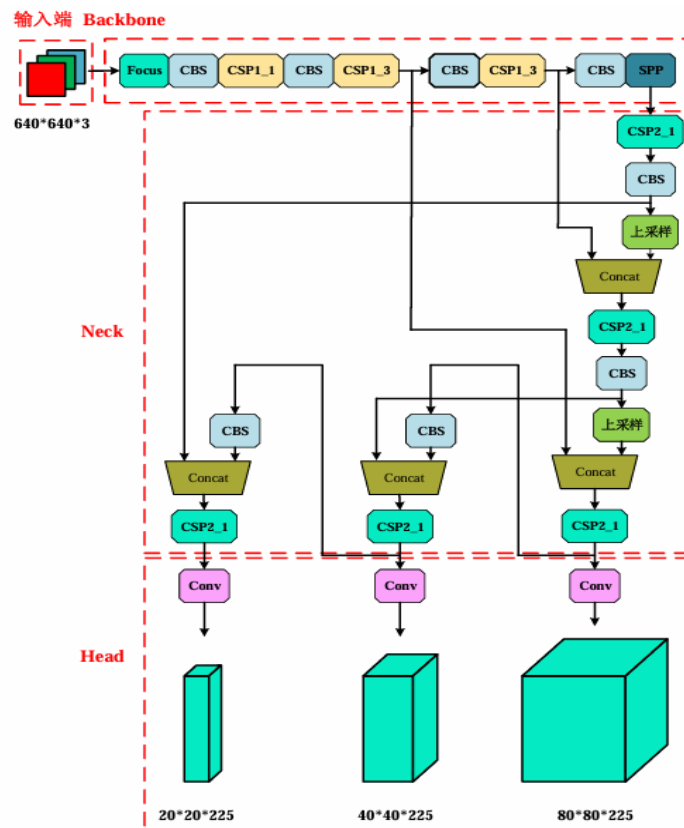


图 2.2 YOLOv5 的网络结构图

图中的 CBS 结构是由 Conv、BN、SiLU 激活函数组成，如图 4.3 所示。



图 2.3 CBS 结构

2.3.1 输入端

(1) Mosaic 数据增强

Mosaic 数据增强是指将四张不同的图像随机组合成一张新的图像，作为模型的输入。在这个过程中，YOLOv5 会对图像进行随机裁剪、缩放、翻转、旋转等操作，通过上述操作获得充足的小目标图像，达到扩充数据集目的，从而增强数据的多样性，提高模型的泛化性能。还可以减少 GPU 使用量，提高检测速率。

(2) 自适应锚框计算

YOLOv5 模型使用锚框（anchor box）来检测物体，而不同大小的物体需要使用不同大小的锚框。在 YOLO 算法中，对于不同的数据集，存在具有初始设置的长度和宽度的锚框。在网格训练过程中，包括预测框与实际目标框比较，用损失信息迭代更新模型参数以提高准确性。为了适应不同大小的物体，YOLOv5 会根据输入图像的尺寸自适应计算锚框的大小和数量，从而使模型能够更好地检测不同大小的物体。

(3) 自适应缩放图像

在一般的目标检测算法中，输入端将不同的图像缩放到统一的大小，这将不可避免地增加一些冗余的黑边，影响推理速度。由于存在部分图像的宽高比不同，YOLOv5 不再将输入图像统一为标准大小，而是根据原始图像的宽高比找到了另一种方法，使用最小黑边策略来减少计算量，这提高了推理速度。YOLOv5 根据输入图像大小自适应调整模型缩放比例以适应不同尺寸的图像，从而使模型能够对不同尺寸的物体进行检测。同时，模型将缩放后的图像分成多个小块，每个小块都进行检测，从而提高模型的精度和效率。

2.3.2 主干网络

主干网络称为 Backbone，主要用于对数据集中的图像进行特征提取，是整个目标检测算法过程中比较重要的模块之一。主干网络主要包括 Focus 模块和 CSP 模块，下面主要介绍这两种模块。(1)Focus 模块 Focus 模块在 YOLOv5m 中的主要作用是减少计算量、增加感受野大小，从而提高模型的感知能力和识别准确率。Focus 模块首先需要对输入图像进行切片（Slice）处理，将输入的高分辨率图像切分成多个低分辨率的子图像^[29]。如图 2.4 所示，输入一个 $4*4*3$ 的图像，将输入的图像划分为四部分，对每部分进行卷积操作，最终得到一个四倍尺寸的输出特征图，即 $2*2*12$ 的特征图。

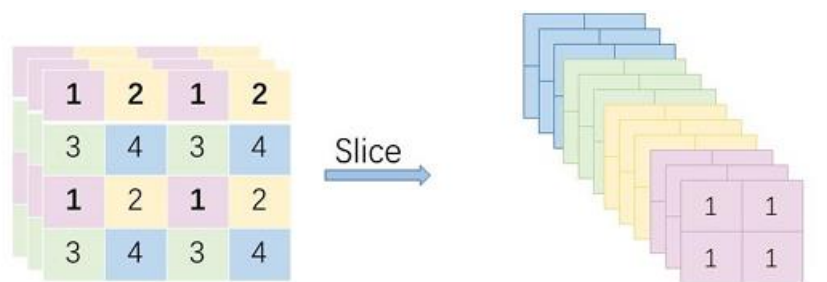


图 2.4 Focus 模块

Focus 模块将输入的图像进行二倍下采样操作，将特征图按照一定的规则进行拼接操作，形成较小尺寸的输出特征图，该特征图被传递给后续的卷积层和检测模块做进一步处理，从而完成目标检测任务。假定在 YOLOv5m 中输入图像的尺寸为 $640 \times 640 \times 3$ ，经过切片处理后，形成 $320 \times 320 \times 12$ 核进行卷积操作，得到 $320 \times 320 \times 32$ 的特征图，再传递给一个 32 个卷积的特征图，Focus 模块操作流程如图 2.5 所示。Focus 模块在 YOLOv5m 网络结构中的表示方式如图 2.6 所示。

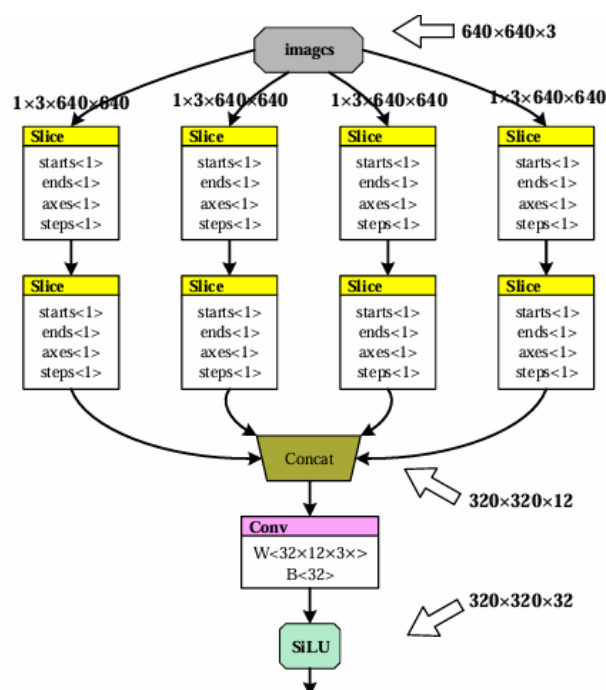


图 2.5 Focus 操作流程

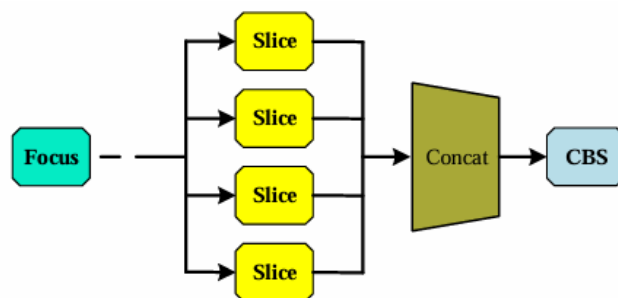


图 2.6 Focus 模块在 YOLOv5m 网络结构

CSP 模块 CSP 模块全称是 Cross Stage Partial Network，被广泛应用于主干网络，用来提升模型的准确性。CSP 模块结构如图 4.7 所示，CSP 模块是将输入特征图分为两部分，将卷积操作与残差组件相结合，通过规则合并结果，提高模型推理速度和梯度值计算效率，同时解决梯度消失问题，提升特征提取能力^[29]。CSP2_X 模块引入卷积层替换了残差组件，卷积层将输入的特征图分为两个部分，每个部分都通过卷积操作进行处理，再经过融合过程，将两个子特征图进行整合，该过程能够更好地保留和利用图像信息。图中 CSP 模块主要应用在两个地方，其中 CSP1_X 应用于主干网络，CSP2_X 应用于 Neck 网络^[34]。

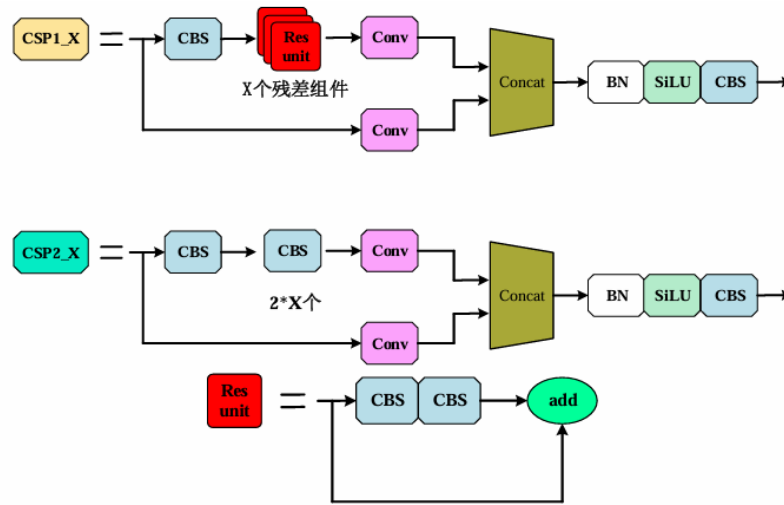


图 2.7 CSP 模块结构图

2.3.3 Neck 网络

在目标检测算法中，主干网络负责输入图像的特征提取，而 Neck 模块是一个特征融合模块，其主要作用是将来自不同网络层的特征图进行融合，以获得更好的目标检测结果^[30]。Neck 模块结构通常是在主干网络的输出层之后添加一个或多个模块，其作用是对特征图进行特征融合、特征增强等操作，以便更好地识别和定位目标。(1) SPP 结构 SPP (Spatial Pyramid Pooling) 结构主要作用是对不同大小的目标进行特征提取，以便实现更准确的目标检测结果。SPP 网络结构流程如图 4.8 所示，SPP 结构一般是作为卷积神经网络的最后一层，它将不同大小的特征图通过池化操作划分成四块，每块分别经过 1×1 、 5×5 、 9×9 、 13×13 的最大池化，并将得到的四个特征图进行拼接，得到一个固定大小的特征向量^[31]。通过这种方式，SPP 结构可以捕捉不同尺度的特征，从而提高目标检测的准确性和稳定性。

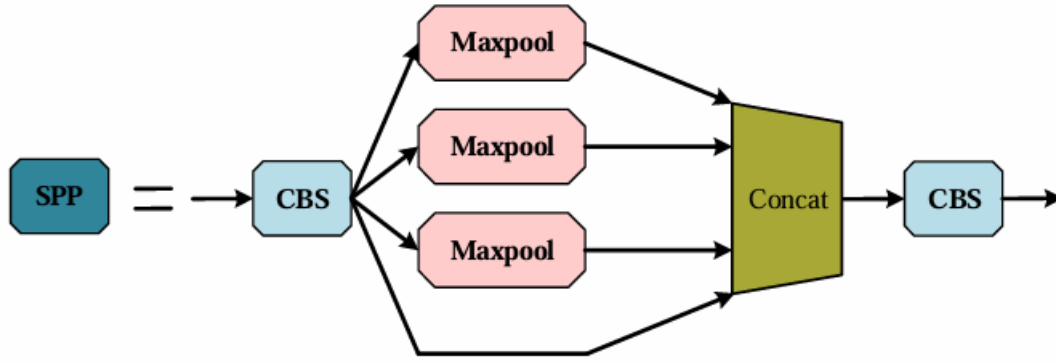


图 2.8 SPP 网络结构流程

FPN 和 PAN 结构 FPN (Feature Pyramid Network) 和 PAN (Path Aggregation Network) 结构主要是解决目标检测算法中尺度不变性问题^[32]。FPN 结构是通过建立特征金字塔，从而得到多个特征图，进而实现不同尺度的特征提取。PAN 结构是在 FPN 结构的基础上引入路径聚合模块，将获得的特征图进行融合，获得图像更加全面和丰富的特征信息。FPN 和 PAN 多尺度特征融合示意如图 4.9 所示，PAN 结构建立一个自下而上的特征金字塔，当输入一张 $640 \times 640 \times 3$ 的图像，会得到 80×80 、 40×40 、 20×20 三个不同尺寸的特征图^[33]。PAN 结构是一个自上而下的特征金字塔，只将强的语义特征信息从上层传递到下层，但不传递定位特征信息。PAN 针对这一问题，强化了 FPN。经过自下而上的特征金字塔，将低级别的强定位特征信息传输到上层。通过这种结构可以获取更多的特征信息，使网络模型的检测性能得到极大的提高^[36]。

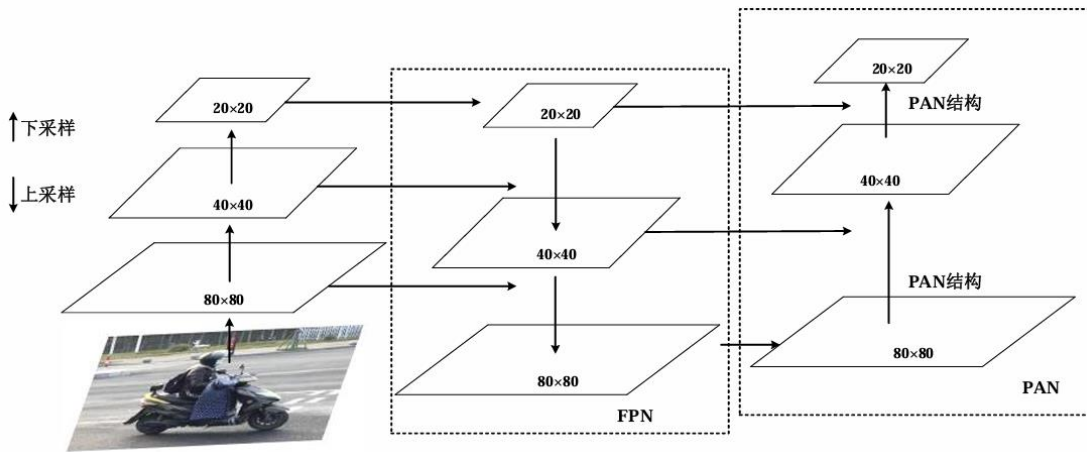


图 2.9 FPN+PAN 结构

2.3.4 输出端

PAN 在 YOLOv5 算法中输出端 (Head) 的主要作用是将经过主干网络与 Neck 结构的网络特征图进行后处理，并最终输出检测结果。输出端包含三个检测层，分别对应三个不同尺度的检测目标。每个检测层由一些卷积层和池化层组成，最终将特征图映射到一个固定的输出大小。通过应用逆卷积等操作将输出大小调整为输入图像的大小，并在每个网格上

预测出目标的类别和位置信息^[37]。YOLOv5 的损失函数是由目标置信度损失函数、分类损失函数、回归损失函数组成，主要目的是最小化检测网络的预测结果与真实标注之间的差距。(1) 损失函数 YOLOv5 使用二元交叉熵损失函数作为分类损失函数，该损失函数适用于多标签 分类任务，每个目标的标签可以被视为相互独立的二元分类任务，并且每个标签的分 类损失是相互独立的^[38]。二元交叉熵函数 (BCELoss) 公式为：

$$BCELoss = -\frac{1}{n} \sum (y_n * \ln x_n + (1 - y_n) * \ln(1 - x_n)) \quad (2.1)$$

式中： y_n --表示二元标签值 0 或 1

x_n --表示二元标签值的概率

YOLOv5 使用 sigmoid 函数作为激活函数，对每个目标的每个标签进行二元分类。对于边界框回归，YOLOv5 使用 CIoU 损失函数作为损失函数，该损失函数是 YOLOv4 中引入的^[39]。与 YOLOv4 使用的 GIoU 相比，CIoU 损失函数更加平滑，并且在检测 任务中表现更好。CIoU 损失函数在计算边界框之间的距离时，考虑到了中心点之间 的距离，可以更好地捕捉到目标的位置信息。CIoU Loss 的计算公式(2.2)为：

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (2.2)$$

$$\alpha = \frac{v}{(1 - IOU) + v} \quad (2.3)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (2.4)$$

式中： α --是权重函数

v --是宽高比相似性

YOLOv5 使用 sigmoid 函数对每个标签进行二元分类，将预测得分限制在 0 到 1 之间。YOLOv5 输出的标签不是排他的，从而更适合处理目标检测任务。对于每个目标，YOLOv5 预测位置和标签的置信度得分，该得分表示每个标签的可能性，而不是总和为 1 的概率^{[39][40]}。(2)NMS 非极大值抑制 非极大值抑制需要经过迭代-遍历-消除这三步。通过 NMS 算法会筛选出一个最佳检测框。NMS 的函数公式为：

$$s_i = \begin{cases} s_i, iou(M, b_i) < N \\ 0, iou(M, b_i) \geq N \end{cases} \quad (2.5)$$

式中： s_i --表示第 i 个预测框置信度 M --表示当前置信度最高的预测框 b_i --表示第 i 个预测框 N --是 IoU 阈值

2.4 目标检测算法

YOLOv5 是一种基于深度学习的目标检测算法，它在 YOLOv4 的基础上进行了 升级

优化，具有更快的检测速度和更高的检测精度。在 YOLOv5 系列中，包括 YOLOv5s、YOLOv5m、YOLOv5l 和 YOLOv5x 等等的模型。本文数据集在 YOLOv5s 和 YOLOv5m 进行预测试。

2.4.1 目标检测评价指标

(1) 精确率、召回率和 F1-分数 精确率（Precision, P）是指分类器正确分类为正样本的能力。精确率计算公式如 (2.5)所示。召回率（Recall, R）是指被分类器正确识别正样本的能力。召回率计算公式为(2.6):

$$P = \frac{TP}{TP + FP} \quad (2.5)$$

$$R = \frac{TP}{TP + FN} \quad (2.6)$$

式中所含参数为混淆矩阵各个值的代表，混淆矩阵是一种用于评估分类模型性能的表格，包含了四个统计量，如表 2.1 所示：真反例（True Negative, TN）、真正例（True Positive, TP）、假正例（False Positive, FP）和假反例（False Negative, FN）^[35]。以车牌检测检测为例，TP 代表正确检测到车牌的数量，TN 代表正确判断非车牌的数量，FP 代表错误地将非车牌判断为车牌的数量，FN 代表将车牌错误地判断为非车牌的数量。

表 2.1 混淆矩阵

	Positive	Negative
True	TP	TN
False	FP	FN

(2) AP 和 mAP AP(Average Precision)是每个类别的平均精度，是对精确度和召回率的综合评价。AP 是用精确度和召回率所形成的 (P-R) 曲线通过计算面积得到，公式如(2.7)所示。在实际计算中由于曲线并不是光滑的，会上下浮动，需要对曲线进行平滑处理。AP 值越高代表模型检测效果越好。

$$AP = \int_0^1 p(r)dr \quad (2.7)$$

式中:p--表示精确率

r--表示召回率

MAP（Mean Average Precision）平均精确度是鉴定检测系统的考核指标之一。它考虑到了检测结果的正确性和排序正确性，是多个检测目标的平均精确度。平均精确度是对全部类别的 AP 值先进行求和，再求取平均值，公式(2.8)为：

$$mAP = \frac{1}{Q} \sum_{q \in Q} AP(q) \quad (2.8)$$

式中：Q--表示所有类别数量之和

(3) FPS FPS(Frames Per Second)是指每秒传输帧数，用于衡量模型每秒可以检测多少张图像，公式如(2.9)所示。t 表示检测单张图像耗时多少 ms.

$$FPS = \frac{1}{t} \quad (2.9)$$

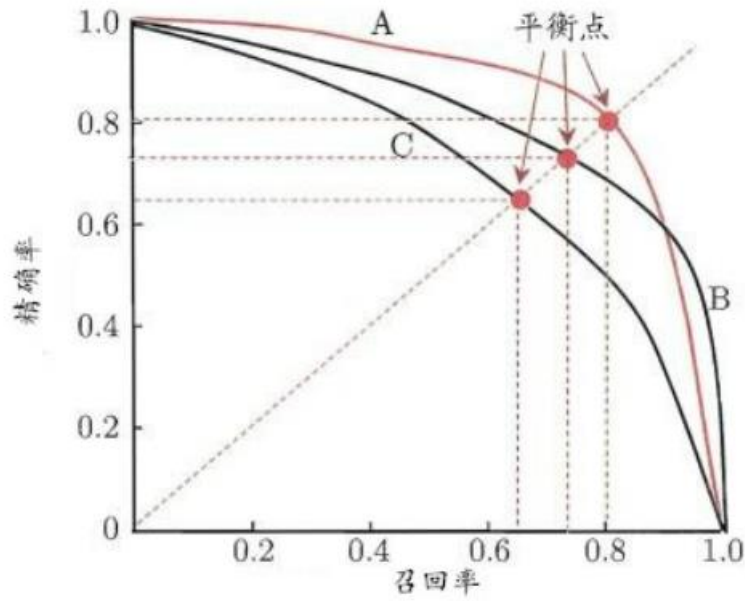


图 2.10 R-P 曲线与平衡点示意图

如图 2.10,如果其中一个学习器的 PR 曲线 A 完全包住另一个学习器 B 的 PR 曲线，则可断言 A 的性能优于 B。但是 A 和 B 发生交叉，可以根据曲线下方的面积大小来进行比较，但更常用的是平衡点 F1。平衡点（BEP）是 $P=R$ 时的取值（斜率为 1），F1 值越大，可以认为该学习器的性能较好。

2.4.2 目标检测算法选择

YOLOv5 在数据集上各个指标如图 2.11 所示。通过同一 CCPD2019 数据集训练 300 次得到 YOLOv5m 模型精确率与召回率曲线图如图 2.12 所示。YOLOv5s 模型精确率与召回率曲线图如图 2.13 所示。

Model	size (pixels)	mAP ^{val} 50-95	mAP ^{val} 50	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7

图 2.11 性能指标

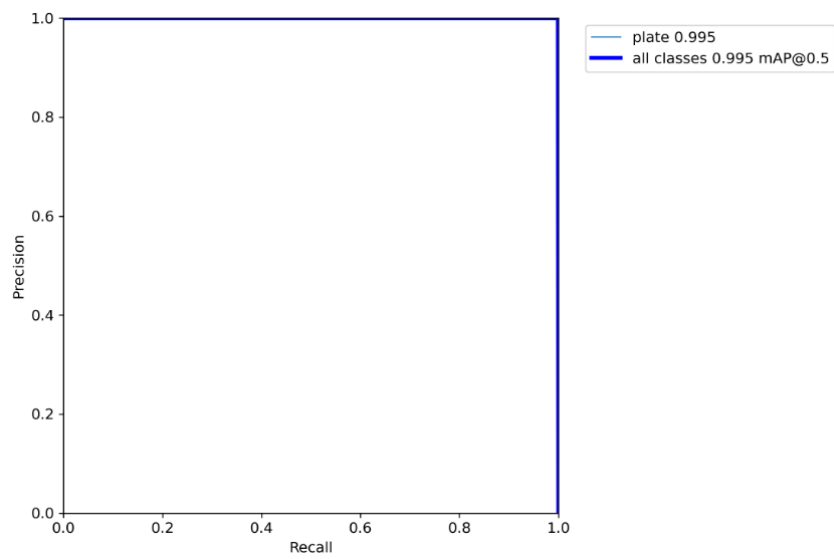


图 2.12 YOLOv5m 模型精确率与召回率曲线图

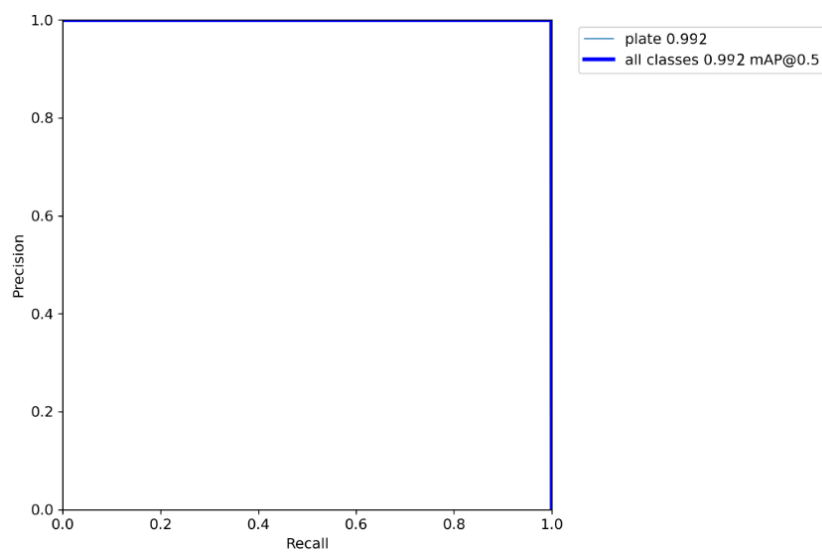


图 2.13 YOLOv5s 模型精确率与召回率曲线图

由表 2.2 可知，本文测试再经过 100 轮迭代和 6000 张图像上进行训练，YOLOv5m 模型性能稍微比 YOLOv5s 略有提升。YOLOv5m 模型的验证精度为 91.13，召回率为 80.29%，mAP 为 91.8%。YOLOv5m 对比 YOLOv5s 模型精确度提高 0.42%。

YOLOv5m 和 YOLOv5s 模型测试性能如表 2.2 所示

Model	Precision%	Recall%	mAP(0.5)%	FPS	Paramete	AP%
YOLOv5m	91.13	80.29	95.3	20.2	21.2	98.1
YOLOv5s	90.71	81.65	94.1	20.2	7.2	97.8

表 2.2 模型测试性能

综上所述，YOLOv5s 具有模型小、检测速度快、参数数量少、在 CCPD2019 数据集上精度相对于 YOLOv5m 表现良好等优点，适合于对实时性要求较高的场景。虽然 YOLOv5s 在目标检测方面检测精度表现良好，但其检测精度还有待提高，本文因为车牌识别需要更高的检测准确度，因此选择 YOLOv5m 模型。

上文训练循环 300 次，得到以下数据分别为图 2.14，图 2.15，图 2.16 训练结果，以及分别使用 yolov5s 和 yolov5m 图 2.17、图 2.18 的识别结果

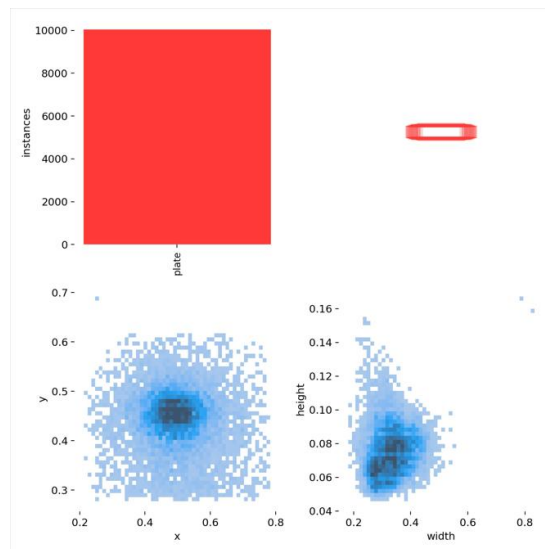


图 2.14 训练集得数据量(左上), labels (右上), center xy (左下) , label 得高宽 (右上)

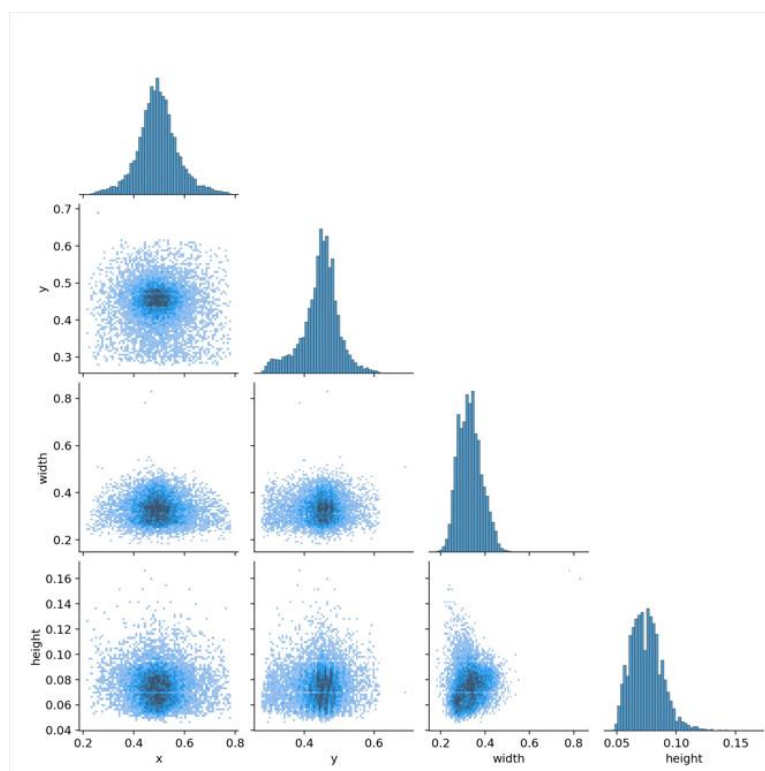


图 2.15 标记框的统计图，

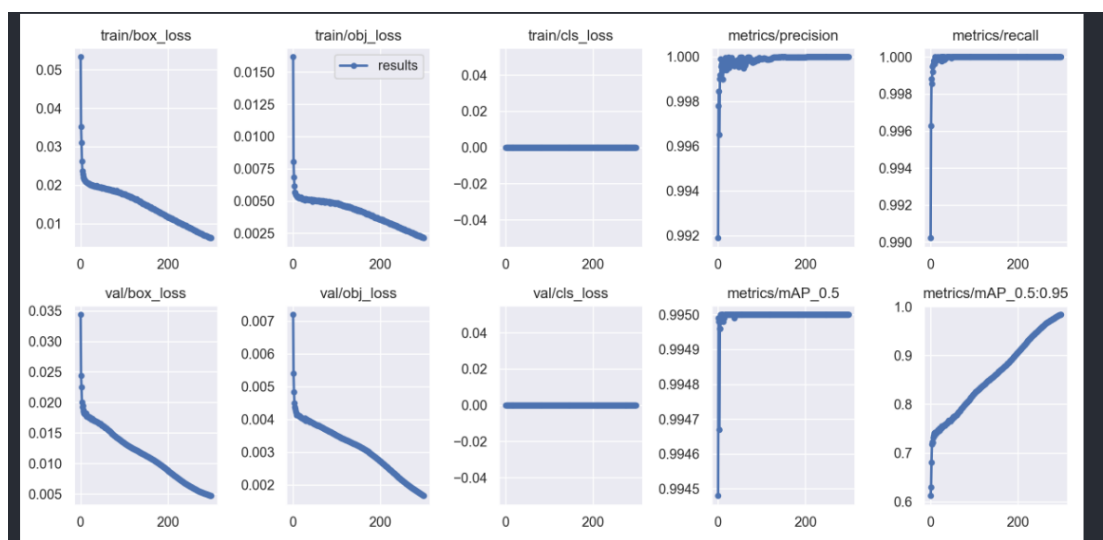


图 2.16 结果

其中：

Box: YOLO V5 使用 GIoU Loss 作为 bounding box 的损失，Box 推测为 GIoU 损失函数均值，越小方框越准；

Objectness: 推测为目标检测 loss 均值，越小目标检测越准；

Classification: 推测为分类 loss 均值，越小分类越准；

Precision: 精度（找对的正类/所有找到的正类）；

Recall: 真实为 positive 的准确率, 即正样本有多少被找出来了 (召回了多少)。

Recall 从真实结果角度出发, 描述了测试集中的真实正例有多少被二分类器挑选了出来, 即真实的正例有多少被该二分类器召回。

val BOX: 验证集 bounding box 损失

val Objectness: 验证集目标检测 loss 均值

val classification: 验证集分类 loss 均值

mAP 是用 Precision 和 Recall 作为两轴作图后围成的面积, m 表示平均, @后面的数表示判定 iou 为正负样本的阈值, @0.5:0.95 表示阈值取 0.5:0.95 后取均值。

mAP@.5:.95 (mAP@[.5:.95])

表示在不同 IoU 阈值 (从 0.5 到 0.95, 步长 0.05) (0.5、0.55、0.6、0.65、0.7、0.75、0.8、0.85、0.9、0.95) 上的平均 mAP。

mAP@.5: 表示阈值大于 0.5 的平均 mAP

一般训练结果主要观察精度和召回率波动情况 (波动不是很大则训练效果较好)

然后观察 mAP@0.5 & mAP@0.5:0.95 评价训练结果。



图 2.17 yolov5s 的检测结果



图 2.18 yolov5m 的检测结果

可以看到可以看到 YOLOv5 都能在不同天气不同光照的情况下也能够识别到车牌, 达到不错的效果, 差别不大。

2.5 文章小节

首先是对深度学习的提出和构成进行阐述，其次介绍构建算法所需的卷积神经网络和循环神经网络内容。随后对 YOLO 进行介绍，并对 YOLO 系列算法发展和改进进行研究。通过对上述理论知识分析和介绍，为后续 YOLOv5 的选择使用提供了理论支持。后面使用数据集对 YOLOv5s 和 YOLOv5m 模型进行训练和测试。根据实验结果，YOLOv5m 在准确率上表现相对较好，被选为目标检测算法的模型。

第三章 车牌识别技术算法

3.1 LPRNet 模型

LPRNet 是一种非常高效的神经网络，它只需要 0.34 GFLOps 就可以进行一次前向传播。此外，模型在 i7-6700K 上运行时在保证实时性的同时，准确率也有保证。面对中文车牌，LPRNet 可进行端到端的训练。LPRNet 的优点可以总结为如下三点：

- (1) LPRNet 不需要字符预先分割，车牌识别的准确率高、算法实时性强、支持可变长字符车牌识别。对于字符差异比较大的各国不同车牌均能够端到端进行训练。
- (2) LPRNet 是第一个没使用 RNN 的实时轻量级算法，能够在包括嵌入式设备在内的各式设备上运行。
- (3) LPRNet 在实际交通监控视频中的应用表明，该算法在视角和摄像畸变、光照条件恶劣、视角变化等复杂的情况下仍表现出很好的识别效果。

3.1.1 相关研究

早期的车牌识别算法由字符分割和紧随其后的字符分类两个阶段构成：

(1) 字符分割通常使用不同的手工算法，结合投影、连接和基于图像组件的轮廓。由于采用二值图像或中间表示作为输入，因此字符分割质量受输入图像噪声、低分辨率、模糊或变形等因素的影响较大。

(2) 字符分类阶段用 OCR 方法实现车牌字符识别。

因为字符分类发生在字符分割之后，所以识别的质量高度依赖分割方法。为了解决字符分割的问题作者采用了基于端到端的卷积神经网络，其将整张车牌的图像作为网络的输入，最终网络输出得到字符序列。

文献^[40]中的无分隔模型是基于变长序列 CTC 损失。使用基于二值图像的 LBP 手工特征作为 CNN 的输入来产生字符类的概率。在输入图像上通过滑动窗口方法获得双向 LSTM 输入序列。由于解码器输出和目标字符序列长度不同，采用 CTC 损耗进行端到端自由分割训练。

文献^[42]中的模型除了将滑动窗口方法替换为 CNN 输出空间分裂到 RNN 输入序列的方法外，基本遵循文献^[41]中描述的方法(“滑动窗口”在 feature map 上，而不是输入)。

相比之下，文献^[43]使用 CNN 模型从整张车牌图像提取特征，通过 11 个全连接模型头将其解码为 11 个字符长度的序列。模型将车牌字符填充为定长序列，每一个全连接头训练来分类指定位置的字符，所以整个网络通过一次前向传递即可完成车牌识别。文中也采

用文献^[44]STN 网络来应对输入图像形变。文献^[45]尝试用一个神经网络来完成车牌的检测与识别。近期一些研究者，尝试使用文献^[46]所述生成对抗网络来进行数据增强，文献^[42]就尝试用该方法获取大型车牌数据集。

LPRNet 并没有在二值图像上提取手工设计特征，而将原始 RGB 图像作为 CNN 的输入。LPRNet 采用全卷积+CTC 损失来取代 LSTM+CTC 损失的模式，训练阶段用 CTC 损失来拟合 CNN 网络，推理阶段用贪婪搜索和前缀搜索完成字符推理。为了获得更好的性能，通过^[47]所述全局上下文嵌入增强了预解码器中间特征映射。文中 LPRNet 轻量级开销网络的构建受到了 SqueezeNet^[48]、Inception^[49-51]、Batch Normalization 和 Dropout 等技术的启发。

3.1.2 LPRNet 结构

LPRNet 是一种为车牌识别任务特别设计的全卷积神经网络模型它通过四个关键组成部分实现高效准确的车牌图像识别。首先，使用 VGGNet 作为骨干网络提取车牌图像中的关键特征，如边缘和纹理。其次，空间变换网络对这些特征进行调整，以适应车牌图像可能出现的旋转和倾斜，增强模型的鲁棒性。接着，双向长短期记忆网络（Bi-LSTM）对特征序列进行建模，捕获字符间的时序关系，确保字符顺序和结构的正确识别。最后，通过双线性池化操作，将特征图转换为固定长度的特征向量，以捕获特征间的复杂关系并实现多尺度特征融合。此外，LPRNet 通过预矫正步骤进行几何校正，采用全卷积设计减少计算量，并通过精简的网络结构和多尺度特征融合技术，有效降低模型的计算复杂度，同时保持对图像特征的高效捕捉能力。这些设计和优化使 LPRNet 在车牌识别任务中不仅表现出高准确性和高鲁棒性，而且具有较低的计算成本，适合在各种实际应用场景中部署。LPRNet 结构如图 3.1 所示。

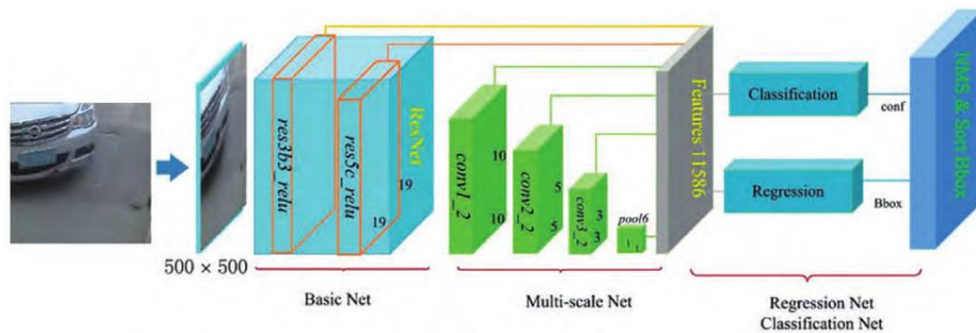


图 3.1 LPRNet 结构

3.1.3 LPRNet 的骨干网络

LPRNet 的骨干网络架构如图 3.1 所示，其中 small_block 结构如图 3.2 所示。

Layer Type	Parameters
Input	94x24 pixels RGB image
Convolution	#64 3x3 stride 1
MaxPooling	#64 3x3 stride 1
Small basic block	#128 3x3 stride 1
MaxPooling	#64 3x3 stride (2, 1)
Small basic block	#256 3x3 stride 1
Small basic block	#256 3x3 stride 1
MaxPooling	#64 3x3 stride (2, 1)
Dropout	0.5 ratio
Convolution	#256 4x1 stride 1
Dropout	0.5 ratio
Convolution	# class_number 1x13 stride 1

图 3.1 LPRnet 骨干网络

Layer Type	Parameters/Dimensions
Input	$C_{in} \times H \times W$ feature map
Convolution	# $C_{out}/4$ 1x1 stride 1
Convolution	# $C_{out}/4$ 3x1 strideh=1, padh=1
Convolution	# $C_{out}/4$ 1x3 stridew=1, padw=1
Convolution	# C_{out} 1x1 stride 1
Output	$C_{out} \times H \times W$ feature map

图 3.2 Small Basic Block

骨干网络以原始 RGB 图像作为输入，用 CNN 提取图像特征。使用结合上下文的 1*13 卷积核代替基于 LSTM 的 RNN。骨干子网络的输出可认为是一个代表对应字符概率的序列，它的长度与输入图像的宽度相关。因为网络的输出编码与车牌字符的长度不相等，CTC 损失常用于输入和输出序列未对齐且长度可变情况。此外，CTC 将每个时间步的概率转换为输出序列的概率。关于 CTC 的详细内容见文献^[53]。

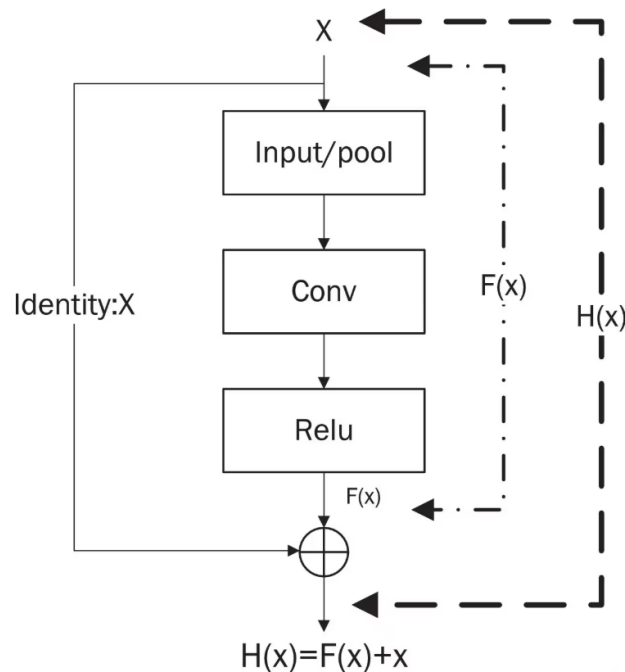


图 3.3 残差网络（ResNet）体系结构

图 3.3 说明了残差网络（ResNet）体系结构的基本块，该体系结构通常用于深度学习，也包括用于车牌识别的 LPRNet 等模型的开发。以下是对每个组成部分含义的分析：

Input/pool: 该块表示输入层或池化层。在神经网络中，输入层接收输入数据（如图像）。如果它是池化层，它减少了特征映射的空间维度（高度和宽度），这有助于减少网络中的计算和权重。

Conv: 这代表一个卷积层。卷积层对输入应用卷积操作，将结果传递给下一层。该操作是车牌识别等图像处理任务中特征提取的基础。卷积操作有助于检测边缘、纹理等特征。

Relu: 这是整流线性单元激活功能。ReLU 用于将非线性引入网络，使其能够学习更复杂的模式。如果输入为正，则函数直接输出输入；否则，输出为 0。这有助于加快训练过程，缓解梯度消失问题。

F(x): 这表示在应用像卷积和激活这样的转换后，上面的层的输出。在残差块中，这是网络在将其添加到原始输入之前学习输出的内容。

残差块: X: 这部分表示“跳过连接”或“标识映射”，这是残余网络的标志。输入“X”直接添加到层的输出“F(X)”中。这使得梯度更容易在网络中流动，这对于非常深的网络特别有益。

$H(x) = F(x) + x$: 这个方程显示了加法运算。在这里，学习到的特征“F(x)”被添加到输入“x”中。这个添加是至关重要的，因为它可以通过允许替代来帮助缓解渐变消失的问题

也就是说 LPRNet 的工作原理是基于深度学习中的卷积神经网络（CNN）和循环神经网络（RNN）。CNN 部分负责从原始图像中提取特征，而 RNN 部分则负责序列预测，通过 CTC 解码输出最终的车牌字符序列。通过端到端的训练，LPRNet 可以直接从车牌图像中预测出完整的车牌号码。

3.2 训练步骤:

训练 LPRNet 通常遵循以下步骤:

- （1）数据准备：收集并预处理大量车牌图像和相应的标签数据，进行标准化处理以适应网络输入。
- （2）模型配置：设置网络参数，例如卷积层的数量和类型，RNN 的配置等。
- （3）损失函数选择：确定使用 CTC 损失函数，并对其进行配置。
- （4）优化器选择：选择适合该任务的优化器，如 Adam 或 SGD，并设置适当的学习率和其他超参数。
- （5）训练：输入数据，进行前向传播，计算损失，然后通过反向传播进行权重更新。

(6) 验证和测试：在独立的验证集上测试模型性能，根据需要进行调整，并使用测试集进行最终评估。

通过上述步骤，LPRNet 可以训练出一个高效准确的车牌识别模型，能够在多种环境和条件下识别车牌号码，适用于智能交通系统和自动化监控领域。

3.3 实验结果与分析

3.3.1 数据集获取与处理

本文使用裁剪后的 CCPD2019 数据集来进行训练模型首先利用基于 LPB (Local Binary Pattern) 的检测器获取图像中车牌的 bounding boxes。然后对所有的车牌进行人工标记。数据集一共包含 1000 张裁剪的车牌图片，其被按照 9:1 的比例分别分成训练集以及验证集。使用自动裁剪的牌照图像用于训练可以使网络对于检测伪像增加稳定性，因为在某些情况下，被裁剪的车牌边缘包含一些周围的背景，而在其他情况下，车牌裁剪得太靠近边缘，没有背景甚至有些时候车牌的某些信息会被裁剪从而丢失。将车牌字符设定为数据标签，并对车牌图像进行了精细裁剪与调整，令车牌的倾斜角度最小化，趋近于零度，这样的处理有助于提高模型对车牌字符的识别准确度，在收集的数据中，标签即对应车牌上的字符信息。从而得到数据集图 3.4。



图 3.4 处理后数据集展示

3.3.2 系统性能与评价指标

本文使用识别精度与识别效率两个指标进行系统的优劣评价，正确识别且正确反馈结果是车牌识别系统的关键，详细介绍如下：

车牌识别准确率 (%)：识别车牌字符号码与车牌自身号码匹配的比例是衡量车牌识别效果的重要指标。比例越高，说明车牌识别效果越好。车牌识别准确率公式为：

$$ACC = \frac{\text{识别正确的车牌数量}}{\text{总测试样本}} \quad (3.1)$$

识别时间 (ms): 系统识别时间快慢对于实际应用非常重要。识别时间过长, 会导致系统响应缓慢, 影响用户体验和实时应用的效果。

3.3.3 性能评估

本章节在实验中对比了 C-RNN 算法和 LPRNet 算法模型从识别准确率和识别时间两个指标进行对比, LPRNet 模型识别结果如图 3.5 所示。



图 3.5 LPRNet 算法模型车牌识别结果

C-RNN 网络与 LPRNet 进行车牌识别实验结果如表 3.1 所示。在测试中, LPRNet 模型车牌识别准确率 96.6%, 单张检测时长 76.45ms。相较于 C-RNN 检测准确率提升了 1.2%, 检测时间提高了 8.09ms。数据证明本文算法模型相较于 C-RNN 表现效果较好。如表 3.1 所示

模型	识别准确率	总共识别时间/ms
C-RNN	95.4%	84.54
LPRNet	96.6%	76.45

表 3.1 测试结果

使用训练集对车牌字符识别网络进行训练，在测试集上进行测试。部分结果如表 3.2 所示。由于两轮电动车车牌同行文字大小差异过大、车牌变形、脏污等导致文本部分识别效果较差，如表 3.3 所示。



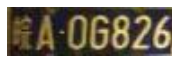


车牌					
预测	ChuanX90621	SuB810FT	WanA0G826	ZheD911SY	YueB0GJ76
标签	ChuanX90621	SuB810FT	WanA0G826	ZheD911SY	YueB0G76
	(a)	(b)	(c)	(d)	(e)

表 3.2 文本识别结果






车牌					
预测	WanA5G895	WanAB361J	ZheJ2K181	YuPU4520	WanNVV229
标签	WanA56895	YuAB361J	ZheJ2K181	YuPU4520	WanNW229
	(a)	(b)	(c)	(d)	(e)

表 3.3 部分误识结果

从文本识别结果表 5.2 中可以看出对不同车牌颜色，模型能够给出正确的预测效果。当车牌发生扭曲如表 5.3(a)所示、由于光照不足如表 5.3(b)所示、形近字识别错误如表 5.3(a)(e)所示，都会对模型的识别效果造成干扰。

在测试集中，对于误识别的车牌，有 39.8%是由于数字识别错误，60.2%是英文识别错误造成的，如表 5.4 所示。由于两轮电动车车牌有至少 1 位英文字符和至少 4 位 数字字符组成。数据集所裁剪的车牌数据集相对较少，导致在训练过程中出现次数较少，模型对其识别的拟合程度不佳。

表 3.4 误识别原因分析表

误识别种类				误识别原因	
字符类型	数字	字母	形近字	图像质量	其他
误识别率	29.8%	61.3%	40.5%	32.34%	28.26%

3.4 本章小节

本章节针对车牌字符识别进行研究，首先是对 LPRNet 的提出和构成进行阐述，其次介绍构建算法所需的卷积神经网络和循环神经网络内容。随后对 LPRNet 进行介绍，并对 LPRNet 系列算法发展和改进进行研究，后面使用数据集对 LPRNet 模型进行训练和测试。根据实验结果与 CNN 算法进行对比，并对 LPRNet 的识别结果进行分析。通过实验指标证明，从 CNN 算法改进后的 LPRNet 性能更为优越。

第四章 车牌识别系统设计与实现

本章介绍了一套基于改进后的 YOLOv5m 与 LPRNet 网络模型相结合,使用 Pyside6 设计了 GUI 界面的道路车牌佩戴与车牌识别检测系统,该系统能实现车牌定位并对车牌进行识别,并对图像进行保存。为验证模型是否具有较好的泛化性能,重新采集数据集对模型性能进行评估。并对系统进行压力测试分析,确保系统在实际使用中能够稳定高效运行。

4.1 系统可行性分析

4.1.1 需求分析

为确保车牌检测系统的顺利研发,需要进行合理的需求分析。在生活中,人们需遵守交通法规,否则将承受经济上的处罚,甚至威胁生命安全。骑行人员需遵守交通规则,但违章现象仍普遍存在。传统的现场执法效率低下,监管难度大,需要开发机动车违章自动检测系统。该系统能够与自动识别违章算法配合,避免过度依赖交通警察现场执法。通过需求分析,明确系统的开发任务和实现目标,可以为系统的研发提供有力支撑。

(1) 人力资源浪费

城市道路网络通常错综复杂,交通密集且人口稠密,这意味着交通警察必须在不断变化的城市环境中保持高度警惕,并快速应对突发事件。尽管交通警察数量在不断增加,但是实现对城市道路的全覆盖仍然是一项艰巨任务。

(2) 检测效率低下

在城市中,驾驶员违法行为时有发生,虽然城市管理部门和交通警察进行了一系列努力来应对问题,但是仍然面临巨大挑战。首先,当在检测过程中发现了违法行为,由于涉及证据收集、罚款等程序,处理时间也相对较长。其次,进行普法教育、罚款等措施也需要耗费时间和精力。特别是一些不配合的骑行人员可能会拒绝接受处罚,这就进一步浪费交通警察和城市管理部门的时间和精力。

4.1.2 系统可行性分析

机动车违章自动检测系统的可行性分析主要从技术和经济两个方面考虑。

(1) 技术可行性分析

从技术角度来看,车牌识别自动检测系统需要准确定位车辆和识别骑行人员车牌然后再与某种检测违规行为判定的系统配合来达到执法作用。在此过程中需要采用计算机视觉技术、图像处理算法、字符识别算法,以实现车牌准确检测与车牌字符识别。本文选择 YOLOv5m 的目标检测算法对车牌进行检测,这一步实现了系统的主要功能。此外,利用 LPRNet 车牌识别技术,进行机动车的车牌识别,为将来车牌和驾驶员建立对应关系做准

备。本文选用的目标检测算法，通过数据集训练后能实现对驾驶员车牌检测功能，并且具有较好的检测性能。虽然技术方案具有较高的复杂度和难度，但随着人工智能技术的发展，机动车违章自动检测系统的开发是可行的。

(2) 经济可行性分析

从经济角度来看，车牌识别自动检测系统的开发需要考虑成本和收益的平衡。一方面，系统的开发和维护需要投入大量的资金和人力资源，这需要对系统的经济效益进行充分评估和预估，但是现有道路监控系统的基础设备已基本完善，只需要将该系统部署到道路监控系统中即可。另一方面，车牌识别自动检测系统能够快速对违规车辆做出响应，有效地减少交通事故，提高城市交通管理的效率和质量，从而产生显著的社会效益，促进城市交通的健康和可持续发展。

4.1.3 系统流程分析

系统使用 YOLOv5m 算法车牌检测，采用 LPRNet 算法模型对车牌字符进行识别。经过车牌检测和车牌识别的流程，系统获得每一图像中的车牌位置以及识别出的车牌号码。这些信息可以被集成到一个最终的结果中，系统还可以将结果保存到文件中，以供后续分析和记录。系统流程如图 4.1 所示。

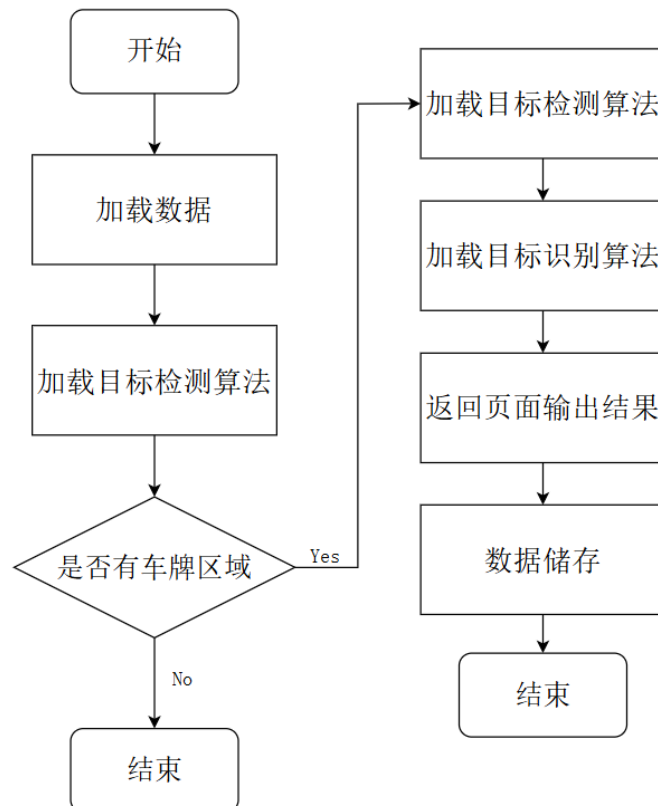


图 4.1 系统工作流程

4.1.4 系统架构设计

本文的车牌检测识别系统架构由硬件层，软件层，加载层，算法层，界面层五部分组成。硬件层和软件层如表 4,1 实验环境所示。 数据加载层：主要是由图像加载、目标检测算法模型加载、车牌字符识别模型加载三部分组成。 算法层：本区域主要是使用 YOLOv5m 算法模型进行车牌区域定位检测。使用 LPRNet 模型进行车牌字符识别，形成一个由上往下一体的算法流程。 展示层：将头车牌信息可视化展示。使用 Pyside6 设计了 GUI 界面使结果更加直观的展示出来，进而判断算法的准确性。整体系统架构图如图 4.2 所示。系统的视频截取模块使用了视觉图形库 OpenCV。



图 4.2 系统架构

环境配置	名称	描述
硬件环境	GPU	NVIDIA GeForce RTX 3060
	CPU	12 th Gen Intel(R) Core(TM)i7-12700H 2.30GHz
	内存	16.0 GB (15.7 GB 可用)
软件环境	操作系统	Window11
	Python	3.8.20
	CUDA	12.4.74
	Anaconda	3
	Vscode	1.81
	Opencv	4.10.0.84
	Torch	1.8.2+cu111

表 4.1 实验环境

4.2 模块设计

结合检测与车牌识别系统的需求分析和现有的技术支撑，对系统所具备的模块进行设计，主要检测出违规车辆的车牌所在的图像，再对车牌信息进行识别，方便后期对骑行人员的追责。该检测系统主要分为四大模块，分别是目标检测模块、人工检测模块、车牌识别模块、可视化模块。

4.2.1 各模块功能

目标检测模块：该模块是使用 YOLO 算法针对车牌目标进行检测。

人工检测模块：主要分为图像检测、视频检测、摄像头检测。人工检测模块可以将待检测或者需复核图像视频进行检测。

车牌识别模块：使用 LPRNet 算法模型对车牌进行识别提取。

可视化模块：主要包括车牌定位及车牌号提取，可以实时了解到检测结果，方便结果更加直观展示给用户。骑行人员车牌识别检测系统功能结构图如图 4.3 所示。

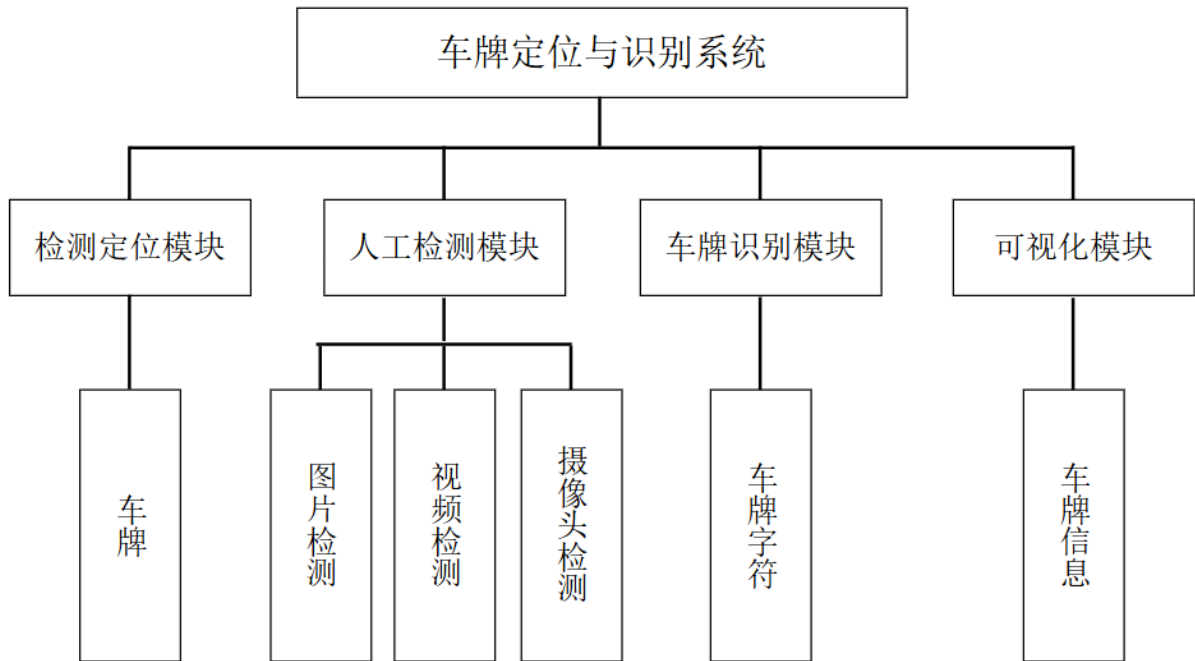


图 4.3 车牌识别检测系统功能结构图

4.2.2 模块功能实现

车牌检测使用 YOLOv5m 实现，将 YOLOv5m 代码封装整合到一个可调用的接口中，使其成为一个可被调用的独立模块，调用实现系统的检测。车牌识别使用 LPRNet 实现，同理封装代码整合到一个接口。本系统使用 PySide6 构建应用程序，创建图像用户界面。该模块允许用户选择对本地图像、本地视频、本地摄像头进行检测。可根据用户不同需求，选择合适的检测方式，无论是分析录制视频还是监控都能得到检测结果。检测结果包括类别信息、预测框、置信度。系统中包括图像检测、视频检测这两部分。检图像大小不能超过 150MB，格式为 JPG 或 PNG，以确保系统的可靠性和性能。若上传视频检测时输入所需检测的视频地址，等待检测即可。视频格式为 mp4。系统检测完后将自动进行数据保存，方便后面数据检查搜集。

4.3 系统测试与效果分析

4.3.1 系统功能测试

车牌识别系统的图像检测界面展示如图 4.4 示。图中车牌用红色框标注：

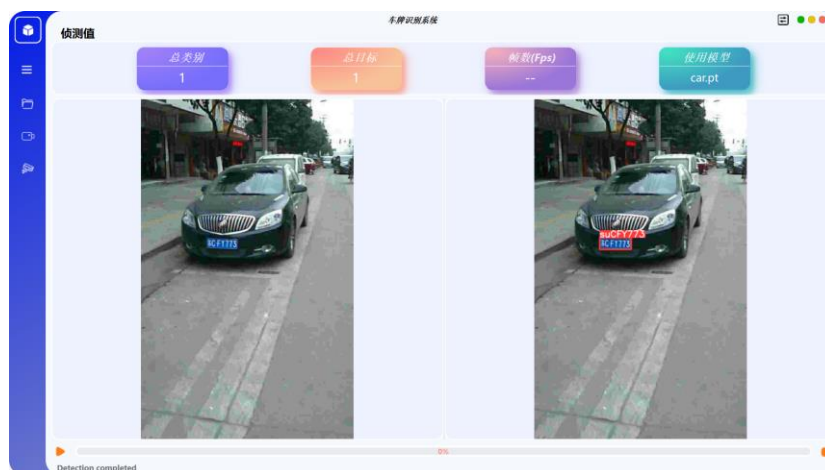


图 4.4 图像检测识别界面

车牌识别系统的视频检测界面展示如图 4.5 和图 4.6 所示。图中展示的是 mp4 视频文件中任意帧画面检测结果。



图 4.5 视频选取



图 4.6 视频检测识别界面

车牌识别系统的摄像头检测界面展示如图 4.7 所示。图中展示的是摄像头任意帧画面检测结果。



图 4.7 摄像头检测识别界面

4.3.2 系统检测和定位效果分析

为检验与测试本文提出的系统综合性能，选择网上相机拍摄道路视频，用于车牌检测并识别。正常光照下效果如图 4.8 所示。光照不足时效果如图 4.9 所示。从拍摄照片中选择了 1000 张图像，其中包括不同光照影响下的图像。图像检测结果如表 4.2 所示，图像识别结果如表 4.2 所示。

图 4.8 正常光照时效果图



图 4.9 光照不足时效果图



表 4.2 系统在不同场景下图像检测结果

场 景 环 境	样 本 数 量	正确检测量	错误检测量	检测准确率	检测时间/ms
正常光照	500	481	19	96.2%	30.1
光照不足	500	470	30	94.0%	30.9

表 4.3 系统在不同场景下车牌识别结果

场 景 环 境	样 本 数 量	正确识别量	错误识别量	识别准确率	识别时间/ms
正常光照	500	467	33	93.4%	78.43
光照不足	500	451	49	90.2%	79.82

从整体上看天气晴朗时检测效果最佳，当碰到雨天、雾天、夜晚时由于光线不够充足和遮挡严重检测效果较差。光照充足的情况下，检测结果 mAP 达到了 96.2%，证明该系统在全新的测试环境下表现出了良好的鲁棒性。该系统能够满足实际检测的基本要求。

从识别效果看，在全新的测试环境中光照充足时车牌的识别率为 93.4%，当光照不充足时，识别率略有下降，但仍能保持 90.2%的识别准确率。从系统测试效果可以看出光照不充足时，仍能识别出车牌信息。通过本次测试证明系统具有较好的识别准确率。

4.4 本章小结

本章主要对车牌检测与识别系统进行设计与实现。首先进行了系统设计的需求分析，根据需求制定了系统各个功能模块，主要分为目标检测模块、人工检测模块、车牌识别模块、可视化模块，并对各个模块进行了实现。最后对车牌检测与识别系统进行了功能测试与性能测试，测试结果表明该检测系统基本满足检测要求。

第五章 总结与展望

5.1 总结

为提高机动车规范信息化管理能力,增强驾驶人员的安全意识,提高交通工作效率,本文设计并实现了一套基于神经网络深度学习车牌检测与识别系统。该系统对道路画面进行检测定位以及识别车牌号码。主要工作如下:

(1)针对数据集问题,本文通过网上的CCPD2019 对其进行处理及训练;对于YOLOv5模型可以对其使用LabelImg 进行人工数据集标注与深度学习方式辅助标注相结合方式进行训练,而LPRNet则需要将其数据集进行裁剪处理,保留车牌区域。防止模型出现过拟合现象,提高模型的泛化能力。选用留出法将数据集划分为训练集与测试集,比例都为9:1,为此提高准确性和效率。

(2)在目标检测算法中考虑到通用性与检测性能选择使用YOLOv5作为基础网络模型。本文数据集在YOLOv5s和YOLOv5m两个模型上进行测试,经过基本性能对比以及训练结果对比,选择YOLOv5m作为目标检测算法。

(3)在车牌字符识别方面,提出一种基于深度学习的车牌识别算法,该算法分为两个阶段:车牌检测和字符识别。在车牌检测阶段使用了YOLOv5m进行车牌区域定位,在字符识别阶段采用一种轻量级的LPRNet网络模型进行车牌的字符识别。实现了车牌检测和文本识别的有效结合,使得车牌识别系统在识别效率方面取得了较好的表现。

(4)根据检测与车牌识别系统需求分析,进行系统软件设计与实现。将改进后的YOLOv5m算法模型融合到基于PySide6界面上进行检测系统开发工作,该系统能够实现对图像和视频画面两个方面的检测,能够直观呈现车牌号码。最后进行保存,对系统进行功能测试和性能评估,验证设计的可行性与系统的实现程度。

5.2 不足与展望

本文在车牌检测与识别系统的检测方法和模型检测效果方面进行了训练与加强,基本实现了设计的要求,但在实际应用中仍需要进一步研究和优化,以满足更加复杂和多样化的需求,以实现在实际生活中的落地应用,具体总结如下:(1)在实际场景中,车牌有可能被其他特殊物品进行遮挡或车牌表面四周模糊受损,为提高算法的识别能力,可以增加这些特殊物品的样本作为训练类别,并使用传统数字图像处理技术对置信度较低的检测框进行筛选,进而提高算法的准确性。(2)在车牌识别部分,由于拍摄距离较远,导致部分图像较为模糊,系统识别错误率较高,后期考虑采用复层运动图像去模糊网络算法或者引入特征重建模块,从而提高车牌识别率。本文对车牌识别工作只针对车辆编号信息进行识别,对双排车牌识别可考虑使用SeqToSeq模型,先进行车牌切分再转换为单行车牌识别。

(3) 在模型识别过程中，可视化界面的美观度和整洁度还需进一步完善。系统的功能性仍有待提升。车牌与用户信息没有建立，系统的安全性也不可忽视，需要严格的安全保障措施。

引用文献

- [1] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[J].Computer Vision and Pattern Recognition,2001,1:511-518.
- [2] Dalal N, Triggs B.Histograms of oriented gradients for human detection[C]//International Conference on computer vision & Pattern Recognition(CVPR05).IEEE Computer Society,2005,1:886--893.
- [3] Felzenszwalb P,Mcallester D,Ramanan D.Adiscriminatively trained, multiscale,deformable part model[J].Computer Vision and Pattern Recognition,2008: 1-8.
- [4] Oren M,Papageorgiou C,Sinha P,et al.Pedestrian detection using wavelet templates [C].Proceedings ofIEEE computer society Conference on computer vision and pattern recognition,1997: 193-199.
- [5] Ojala T,Pietikainen M,Maenpaa T.Multiresolution gray-scale and rotation invariant texture classification with local binary patterns[J].IEEE Transactions on pattern analysis and machine intelligence,2002,24(7): 971-987.
- [6] Girshick R,Donahue J,Darrell T,et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition.2014:580-587.
- [7] 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.
- [8] Redmon J, Divvala S,Girshick R,et al. You only look once:Unified real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016:779-788.
- [9] Girshick R.Fast R-CNN[C]//Proceedings of the IEEE international conference on computer vision. 2015:1440-1448.
- [10] Liu W,Anguelov D,Erhan D, et al. Ssd:Single shot multibox detector[C]//European conference on computer vision.Springer,Cham,2016:21-37.
- [11] Lin T,Goyal P,Girshick R,He K.Focal loss for dense object detection[J].IEEE transactions on pattern analysis and machine intelligence,2018: 2999-3007.

- [12] 丁鹏欣. 基于深度学习的图像目标检测关键技术研究[D].成都:四川大学,2021.
- [13] Lindeberg T. Scale invariant feature transform[J]. Scholarpedia.Chapter,2012,7(5): 47-48.
- [14] Dollár P,Wojek C,Schiele B,et al.Pedestrian detection:A benchmark[C].2009 IEEE Conference on Computer Vision and Pattern Recognition,2009:304-311.
- [15] 陈冠宇. 基于深度学习的小目标检测方法研究[D].中国地质大学,2020.
- [16] Schapire R E. Explaining adaboost[M]//Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013:37-52.
- [17] 路齐硕. 基于深度学习的目标检测方法研究[D].北京:北京邮电大学,2020.
- [18] 吴迪. 基于多维感知的周界入侵目标检测系统[D].沈阳:沈阳工业大学,2022.
- [19] Fadhil A A, Adnan M, Radhi H, et al. Automatic Diagnosis of Breast Cancer in Histopathologic Images Based on Convolutional AutoEncoders and Reinforced Feature Selection[J]. Majlesi Journal of Electrical Engineering, 2012, 6(2):36-39.
- [20] 李雪露,杨永辉,储茂祥等.基于改进 Faster C-RNN 的钢板表面缺陷检测[J].安徽大学学报(自然科学版),2023,47(02):66-73.
- [21] Ren S, He K,Girshick R.Faster C-RNN: Towards real-time object detection with region proposal networks[J]. in Advances in neural information processing systems, 2015:91-99.
- [22] He K, Gkioxari G, Dollár P, et al. Mask C-RNN[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [23] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
- [24] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.
- [25] Hayat A, Morgado-Dias F. Deep learning-based automatic safety helmet detection system for construction safety[J]. Applied Sciences, 2022, 12(16): 8268.
- [26] 徐振博. 基于视觉的车牌与车辆的检测、识别与追踪技术研究[D].合肥:中国科学技术大学,2021. Kocer H E, Cevik K K. Artificial neural networks based vehicle license plate recognition[J]. Procedia Computer Science, 2011, 3: 1033-1037.
- [27] 贾世娜. 基于改进 YOLOv5 的小目标检测算法研究[D].南昌:南昌大学,2022.

- [28] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks [J]. Pattern recognition, 2018, 77: 354-377.
- [29] Leshno M, Lin V Y, Pinkus A, et al. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function[J]. Neural networks, 1993, 6(6): 861-867.
- [30] Hu X, Liu Y, Zhao Z, et al. Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network[J]. Computers and Electronics in Agriculture, 2021, 185: 106135.
- [31] 张绍. 激光主动成像下基于卷积神经网络的目标检测算法研究[D].中国科学院大学(中国科学院长春光学精密机械与物理研究所),2021.
- [32] Koutnik J, Greff K, Gomez F, et al. A clockwork rnn[C]//International conference on machine learning. PMLR, 2014: 1863-1871
- [33] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks [J]. Pattern recognition, 2018, 77: 354-377..
- [34] 谭智超. 基于数据增强策略的遥感图像舰船目标检测研究[D].哈尔滨:哈尔滨工程大学,2020.
- [35] Rupapara P, Rangavajjula A, Jain A. Low complexity image fusion in bayer domain using a monochrome sensor and bayer sensor[C]//2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017: 1980-1984.
- [36] Collier D. Comment: QCA should set aside the algorithms[J]. Sociological Methodology, 2014, 44(1): 122-126.
- [37] 孙璐,梁永全.融合网格划分和 DBSCAN 的改进聚类算法[J].计算机工程与应用,2022,58(14):73-79.
- [38] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 875-8768.
- [39] 范振宇. 基于卷积神经网络的大地电磁深度学习反演研究[D].中国地质大学(北京), 2020.
- [40] H. Li and C. Shen, "Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs," arXiv:1601.05610 [cs], Jan. 2016, arXiv: 1601.05610. 2, 3

- [41] H. Li and C. Shen, “Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs,” arXiv:1601.05610 [cs], Jan. 2016, arXiv: 1601.05610. 2, 4
- [42] T. K. Cheang, Y. S. Chong, and Y. H. Tay, “Segmentation-free Vehicle License Plate Recognition using ConvNetRNN,” arXiv:1701.06439 [cs], Jan. 2017, arXiv:1701.06439. 2
- [43] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj, and K. R. Ramakrishnan, “Deep Automatic License Plate Recognition System,” in Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, ser. ICVGIP ’16. New York, NY, USA: ACM, 2016, pp. 6:1–6:8. 2
- [44] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial Transformer Networks,” arXiv:1506.02025 [cs], Jun. 2015, arXiv: 1506.02025. 2, 3
- [45] H. Li, P. Wang, and C. Shen, “Towards End-to-End Car License Plates Detection and Recognition with Deep Neural Networks,” ArXiv e-prints, Sep. 2017. 2
- [46] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Networks,” ArXiv e-prints, Jun. 2014. 2
- [47] W. Liu, A. Rabinovich, and A. C. Berg, “ParseNet: Looking Wider to See Better,” arXiv:1506.04579 [cs], Jun. 2015, arXiv: 1506.04579. 2, 3
- [48] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and
- [49] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” arXiv:1602.07261 [cs], Feb. 2016, arXiv: 1602.07261. 2, 3
- [50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” arXiv:1409.4842 [cs], Sep. 2014, arXiv: 1409.4842. 2, 3
- [51] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” arXiv:1512.00567 [cs], Dec. 2015, arXiv: 1512.00567. 2, 3
- [52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” J. Mach. Learn. Res., vol. 15, no.

1, pp. 1929–1958, Jan. 2014. 2

[53] A. Hannun, “Sequence modeling with ctc,” Distill, 2017, <https://distill.pub/2017/ctc>. 3

致谢

在本次论文设计过程中，感谢我的学校，给了我学习的机会，在学习中，老师从选题指导、论文框架到细节修改，都给予了细致的指导，提出了很多宝贵的意见与建议，老师以其严谨求实的治学态度、高度的敬业精神、兢兢业业、孜孜以求的工作作风和大胆创新的进取精神对我产生重要影响。他渊博的知识、开阔的视野和敏锐的思维给了我深深的启迪。这篇论文是在老师的精心指导和大力支持下才完成的。

感谢所有授我以业的老师，没有这些年知识的积淀，我没有这么大的动力和信心完成这篇论文。感恩之余，诚恳地请各位老师对我的论文多加批评指正，使我及时完善论文的不足之处。

谨以此致谢最后，我要向百忙之中抽时间对本文进行审阅的各位老师表示衷心的感谢。