

分 类 号: O241.8
研究生学号: 2018312026

单位代码: 10183
密 级: 公开



吉 林 大 学

硕士学位论文

(学术学位)

基于 YOLO 系列的目标检测改进算法

An Improved Algorithm for Object Detection Based on YOLO
Series

作者姓名: 刘彦清

专 业: 计算数学

研究方向: 深度学习

指导教师: 关玉景 教授

培养单位: 吉林大学数学学院

2021 年 8 月

基于 YOLO 系列的目标检测改进算法

An Improved Algorithm for Object Detection Based on YOLO
Series

作者姓名：刘彦清

专业名称：计算数学

指导教师：关玉景 教授

学位类别：学术学位硕士

答辩日期：2021 年 8 月 29 日

吉林大学硕士学位论文原创性声明

本人郑重声明：所呈交学位论文，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：刘新强

日期：2021年8月29日

关于学位论文使用授权的声明

本人完全了解吉林大学有关保留、使用学位论文的规定，同意吉林大学保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权吉林大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。


（保密论文在解密后应遵守此规定）

论文级别： ☒ 硕士 ☐ 博士

学科专业： 计算数学

论文题目： 基于 YOLO 系列的目标检测改进算法

作者签名： 

指导教师签名： 

2021 年 8 月 29 日

摘要

基于 YOLO 系列的目标检测改进算法

目标检测是计算机视觉领域重要的研究课题，在工业界具有广泛的实际应用价值。本文聚焦于目标检测任务，提出了基于 YOLOv5 的目标检测改进算法，在具有挑战性的 MS COCO2017 数据集上进行了实验。实验结果是：以 YOLOv5s 和 YOLOv5m 为基准，本文的改进模型在基本维持推理速度的同时，也提升了检测的精确度，尤其对于中、小目标的检测有较好的提升，具有一定的实际意义。本文的主要内容可以分为如下几个部分：

第一部分介绍了目标检测的研究背景，综述了基于深度学习的主流目标检测算法以及典型的特征融合结构。

第二部分介绍了 YOLO 系列的最新改进算法 YOLOv5，从数据增强、网络结构、损失函数等方面进行了具体的阐述。

第三部分提出了本文改进的新型特征融合方式。首先，考虑到 BiFPN 结构的优势，我们将双向特征融合应用到检测网络中，并将原来按像素求和的融合策略更改为按通道拼接的方式；其次，改进低层级特征的生成方式，充分利用低层级特征的信息，以达到提高小目标检测精度的目的。

实验部分，我们选择具有挑战性的包含 11 万张图像的 MS COCO2017 数据集。与 YOLOv5s 进行对比实验，我们的改进模型精度（mAP）由原来的 36.7% 提升到 38.1%，召回率由 57.4% 提升到 58.4%；在小尺度目标上，改进模型精度由原来的 21.0% 提升到 22.7%；在中尺度目标上，由 42.1% 提升到 43.0%。与 YOLOv5m 进行对比实验，改进模型精度由原来的 44.5% 提升到 44.7%；在小尺度目标上，改进模型精度由 27.4% 提升到 28.6%。其次，对于损失函数部分，我们将原来的二分类交叉熵损失改为 Focal loss 进行了对比实验。我们发现，与二分类交叉熵损失相比，使用 Focal loss 并不一定能够提升精度。此外，我们还将改进后的算法与其他同类方法进行比较。数据结果表明，以 YOLOv5s 和 YOLOv5m 为基准，我们的改进模型在检测的精度和召回率上都有一定的提升，尤其对于小、中尺度目标有较好的提升。最后，测速实验显示我们的模型在提高精度的同时，也基本维持了推理速度。

关键词： 目标检测，YOLOv5 算法，特征融合，MS COCO 数据集

Abstract

An Improved Algorithm for Object Detection Based on YOLO Series

The task of object detection is an important subject in the field of computer vision and has a wide range of practical applications in industry. In this paper, we focus on the task of object detection, and propose an improved object detection algorithm based on YOLOv5. In addition, we conduct experiments on the challenging MS COCO2017 dataset. The experimental result is that based on YOLOv5s and YOLOv5m, our model improves the accuracy of the detection especially for medium and small targets, while basically maintaining the inference speed.

The main contents of this paper can be divided into the following parts:

The first part, we introduce the research background of object detection, summarize the typical object detection algorithms based on deep learning, and expound the typical methods of feature fusion.

The second part, we introduce the latest algorithm of the YOLO series, YOLOv5. Specifically, we expound YOLOv5 algorithm from these aspects: data augmentation, network structure, loss function and so on.

The third part, we propose a novel feature fusion method. Firstly, considering the advantages of BiFPN structure, we apply the bidirectional feature fusion method to our detection network and change the original fusion strategy of pixel-summation to channel concatenate. Secondly, we make full use of the information of low-level features in order to improve the detection ability of small targets.

The experimental part, we select the challenging MS COCO2017 datasets for experiment which including 110,000 images. Compared with YOLOv5s, the average precision of our model is improved from 36.7% to 38.1% and the recall is increased from 57.4% to 58.4%. The average precision of our model is improved from 21.0% to 22.7% on the small scale objects, and from 42.1% to 43.0% on the medium scale objects; Compared with yolov5m, the average precision of our improved model is increased from 44.5% to 44.7%, and from 27.4% to 28.6% on the small scale objects. Moreover, the loss function is changed from binary cross-entropy loss to focal loss for comparative experiments. We find that focal loss does not necessarily improve the accuracy.

Furthermore, we compare our improved algorithm with other similar methods. The results show that based on YOLOv5s and YOLOv5m, our model is improved on both detection average precision and recall, especially for small and medium scale targets. Finally, the speed measurement experiment shows that our model can not only improve the average precision but also basically maintain the speed of inference.

Keywords: Object detection, YOLOv5 algorithm, Feature fusion, MS COCO dataset

目 录

第 1 章 绪论	1
1.1 深度学习的发展与现状	1
1.2 目标检测的发展与现状	2
1.3 基于深度学习的小目标检测	4
第 2 章 基于深度学习的目标检测算法与特征融合结构	6
2.1 引言	6
2.2 基于深度学习的典型目标检测算法	6
2.2.1 两阶段 (two-stage) 目标检测算法	6
2.2.2 一阶段 (one-stage) 目标检测算法	8
2.3 典型的特征融合网络结构	12
2.3.1 引言	12
2.3.2 特征金字塔网络 (FPN)	13
2.3.3 路径聚合网络 (PANet)	14
2.3.4 双向特征金字塔网络 (BiFPN)	14
第 3 章 YOLOV5 目标检测算法	17
3.1 引言	17
3.2 数据增强	17
3.3 网络结构	18
3.3.1 YOLOv5 网络结构	18
3.3.2 Focus 结构	20
3.3.3 CSP 结构	21
3.4 损失函数	23

第 4 章 新型的特征融合网络及其数据实验结果.....	25
4.1 引言.....	25
4.2 新型的双向特征融合	25
4.3 $P2$ 层级特征的充分利用.....	27
4.4 数据实验.....	28
4.4.1 数据集介绍.....	28
4.4.2 评估标准	29
4.4.3 实验设置	30
4.4.4 数据预处理.....	30
4.5 实验结果.....	31
第 5 章 结论和后续研究方向	37
参考文献	38
致谢.....	46

第1章 绪论

1.1 深度学习的发展与现状

深度学习一词最初于 20 世纪 80 年代被提出,是机器学习的一个重要分支。深度学习的前身是人工神经网络,它是深度学习最早期建立的网络模型。人工神经网络走向低迷的原因在于,单层感知机只对线性问题具有分类能力,而无法完成非线性问题的分类。后来,反向传播算法被应用于神经网络的训练中,使神经网络具有了非线性表示能力,解决了多层感知机无法训练的问题。然而,反向传播算法的局限性在于,随着神经元节点的增多,训练速度会大大降低。此外,由于非凸优化的弊端,用梯度下降法训练很容易导致网络收敛到局部最优解。最严重的问题是,由于层数的增多,往往会引发梯度消失现象,导致网络的学习能力无法提升,这些问题制约了神经网络的发展。

2006 年 Geoffrey Hinton 首次提出深度信念网络的概念^[42],该网络由一系列受限玻尔兹曼机^[28]组成。Hinton 等人将该方法应用于手写字体识别的实验中,取得了很好的效果。预训练是深度信念网络的一个重要步骤,该操作能够使网络的参数找到一个接近最优解的初始值,再利用微调技术对整个网络进行训练,从而达到优化网络的效果。自此,神经网络训练的速度得到了大大的提升。与此同时,由反向传播引起的梯度消失问题也因此得到了有效的解决。

此后,深度神经网络快速发展,各种不同结构的神经网络(如 AlexNet^[27]、VGG^[44]、GoogLeNet^[45]、ResNet^[46])相继被提出,卷积神经网络是深度学习最具有代表性的模型。2012 年的 ImageNet 竞赛上,Hinton 教授和他的学生们对包含一千种类别的一百多万张图片进行分类,利用卷积神经网络的优势,达到了错误率仅有 15% 的优秀结果。此外,Hinton 教授和他的团队将权重衰减操作应用到网络的训练中,有效地减少了网络过拟合^[43]现象。后来,随着 GPU 加速技术的发展、计算机计算能力的提升,深度学习模型不仅提升了图像识别的精度,也大量降低了人工提取特征的时间成本。

现如今,深度学习的广泛研究极大地促进了人工智能及机器学习的发展,深度学习在自然语言处理和计算机视觉等多个领域都有很好的研究成果,使得多项技术任务有了突破性的进展。本文所研究的目标检测任务,便是属于计算机视觉的研究领域范畴。

1.2 目标检测的发展与现状

目标检测作为计算机视觉领域中最根本也是最具有挑战性的问题之一，近年来受到社会各界的广泛研究与探索。作为计算机视觉领域的一项重要任务，目标检测通常需要完成的是：提供数字图像中某类视觉对象（如人类、动物或汽车）的具体位置。此外，目标检测也是许多其他任务（例如：实例分割、图像描述生成、目标追踪等）的重要环节。

从应用的角度来看，目标检测可以分为两个研究主题：一般场景下的目标检测和特定类别的目标检测。两者的区别是：前者类似于模拟人类的视觉和认知，主要意图是探索在统一框架下检测出不同类别物体的方法；而后者是指在特定的应用场景下的检测，如人脸检测、行人检测、车辆检测等任务。近年来，深度学习技术的飞速发展给目标检测带来了显著的突破。目标检测目前已经被广泛地应用于许多现实世界的场景中，如自动驾驶、机器人视觉、视频监控等。

过去的20年，目标检测的发展历程大致经历了两个历史时期：传统的目标检测时期（2014年以前）和基于深度学习的检测时期（2014年以后）。传统的目标检测算法可以概括为以下几个步骤：首先，采取滑动窗口的方式遍历整张图像，产生一定数量的候选框；其次，提取候选框的特征；最后，利用支持向量机^[29]（SVM）等分类方法对提取到的特征进行分类，进而得到结果。由于当时缺乏有效的图像表示，人们只能设计复杂的特征表示，并通过各种加速技能来充分利用有限的计算资源。该时期主要的检测方法有：

（1）Viola Jones 检测器^[1]：Viola Jones 检测器由三个核心步骤组成，即 Haar 特征和积分图、Adaboost 分类器^[47]以及级联分类器。

（2）HOG 检测器^[2]：HOG 检测器利用了方向梯度直方图（HOG）特征描述子，通过计算和统计局部区域的梯度方向直方图来构建特征。HOG 特征与 SVM 分类器算法的结合，在行人检测任务中应用广泛且效果显著。然而，HOG 检测器的缺点是始终需要保持检测窗口的大小不变，如果待检测目标的大小不一，那么 HOG 检测器需要多次缩放输入图像。

（3）基于部件的可变形模型（DPM）^[3]：DPM 所遵循的思想是“分而治之”，训练过程中学习的是如何将目标物体进行正确地分解，而推理时则是将不同的部件组合到一起。比如说，检测“汽车”问题可以分解为检测“车窗”、“车身”和“车轮”等。

早期的目标检测任务提取特征时，主要的方式是人工提取，具有一定的局限性，手工特征的性能也趋于饱和。2012年起，卷积神经网络的广泛应用使得目标检测

也开启了新的征程。2014年 R. Girshick 等人提出的 R-CNN^[4]算法横空出世，目标检测开始以前所未有的速度快速发展。深度学习时代，目标检测算法根据检测思想的不同通常可以分为两大类：两阶段（two-stage）检测和一阶段（one-stage）检测。

两阶段检测算法基于提议的候选框，是一个“由粗到细”的过程。首先产生区域候选框，其次提取每个候选框的特征，最后产生位置框并预测对应的类别，特点是精度高但速度慢。最早期的 R-CNN 算法利用“选择性搜索^[40]”方法产生候选框、卷积神经网络提取特征、支持向量机分类器进行分类和预测。虽然 R-CNN 算法具有一定的开创性，但生成的候选框大量重叠，存在计算冗余的问题。2014年，He 等人提出 SPPNet^[5]算法，利用空间金字塔池化层对不同尺度的特征图进行池化并生成固定长度的特征表示，减少反复缩放图像对检测结果造成的影响。然而，SPPNet 的缺点是：模型的训练仍然是分多步的；SPPNet 很难对 SPP 层之前的网络进行参数微调，导致效率降低。2015年，R. Girshick 等人提出 Fast R-CNN^[6]算法，对 R-CNN 与 SPPNet 算法做出进一步改进，提出感兴趣区域池化层（ROI），使得检测的速度和精度大大提升。不久后，Ren 等人提出 Faster R-CNN^[7]算法，实现了端到端地训练，用 RPN 网络代替选择性搜索，大大减少了训练和测试的时间。

一阶段检测算法基于边界框的回归，是一个“一步到位”的过程。一阶段检测网络在产生候选框的同时进行分类和边界框回归，特点是速度快但精度稍逊。2016年，Redmon 等人提出了 YOLOv1^[9]算法，该算法将图像分割成 $S \times S$ 个网格，基于每个网格对应的包围框直接预测类别概率和回归位置信息。同年，Liu 等人提出 SSD 算法^[8]，该算法借鉴 YOLO 算法的思想，并利用多尺度特征图进行预测。2017年，Redmon 等人提出 YOLOv2^[10]算法，用 DarkNet-19 对 YOLOv1 的网络结构进行了修改。2018年，Redmon 等人提出 YOLOv3^[11]算法，改进之处是：YOLOv3 借鉴 FPN 结构，进行了多尺度预测；借鉴 ResNet^[46]网络结构，将 DarkNet-19 改进为 DarkNet-53。研究者们对算法不断地改进，使得检测的性能不断提高。

此外，基于关键点（anchor-free）进行的目标检测是近期一个比较新颖的研究方向。主流的目标检测算法，比如两阶段的 R-CNN 算法和一阶段的 SSD 算法，基本上都是基于边界框（anchor-based）完成的。基于边界框的设计思想存在一些问题：许多目标的形状是不规则的，边界框可能会包含一些非目标的区域，对检测造成干扰；边界框的数量、大小和宽高比等超参数的设置需要根据数据集的不同而调整；边界框的庞大数量可能导致正、负样本的不平衡问题，导致检测的性能不佳。2018年，Law H 等人提出 CornerNet^[24]，将目标检测问题转换为关键点检测问题。

CornerNet 使用单个卷积神经网络, 预测目标的左上角和右下角两个关键点, 进而得到预测框。CornerNet 关注的是目标的边缘, 而对于目标来说, 最具有辨识度的信息应该在目标的内部。因此, 2019 年 Duan K 等人提出 CenterNet^[26], 增加了对中心点的检测, 通过中心点的位置再回归目标的其他属性。

综上, 目标检测是计算机视觉领域中的研究热点。本文对目标检测的主流两阶段和一阶段算法做出了简要的综述, 内容详见第 2 章。

1.3 基于深度学习的小目标检测

深度学习的快速发展使得目标检测技术获益匪浅, 近年来深度学习已被广泛应用于目标检测领域。然而, 小尺度目标在图像中的像素占比少, 自身的语义信息较少。与目前较为成熟的大、中尺度的目标检测技术相对比, 小目标检测的效果相对不佳, 因此如何提高小目标的检测精度是目前计算机视觉领域的一个难点问题。

微软公司提出的 MS COCO^[19]数据集中, 将区域面积小于 32×32 像素值的目标定义为小目标。目前小目标检测性能相对较差的原因可以归结如下: (1) 小目标自身固有的分辨率很低、像素占比少, 因此目标检测网络提取到的有效信息是非常有限的。(2) 输入图像通过卷积神经网络多次下采样后, 会导致小目标的信息损失严重。(3) 缺少大规模的小目标检测数据集, 目前目标检测领域的常用数据集(如 SUN^[30]、PASCAL VOC^{[31][32]}、ImageNet^[33]等) 大多是针对中型、大型尺度的目标进行检测。针对上述问题, 国内外研究者提出了相应的改进方法, 如: 数据增强、多尺度特征融合和超分辨率等。

首先, 数据增强是一种有效的改进技巧, 可以通过数据增强来增加小目标的样本数量。Kisanta 等人^[34]提出使用过采样和增强方法, 先调整小目标的尺度(缩放范围是 $\pm 20\%$)和位置(旋转范围是 $\pm 15\%$), 再复制小目标, 然后将小目标粘贴到新位置并确保新粘贴的位置不与现有目标重叠, 以此增加图像中小目标的数量。此外, YOLOv4^[15]中的 Mosaic 数据增强方法, 也增加了小目标在训练集中的样本数量。本文的实验环节使用了 Mosaic 方法, 该方法的内容将在 3.2 节具体描述。

其次, 大多数目标检测方法都是利用卷积神经网络进行特征提取, 而卷积神经网络大多采用的是最顶端的高层特征。小目标固有的分辨率低, 经过多次下采样后特征图持续不断减小, 导致小目标的细节信息丢失严重。多尺度特征融合方式的提出有效缓解了这一现象, 在计算量消耗不大的同时增强了特征的表达能力, 提高小目标检测的效果。FPN^[12](特征金字塔网络)是典型的多尺度融合结构, 它利用自

上而下的路径和横向连接，将高分辨率的低层特征与丰富语义信息的高层特征进行融合，后来一些基于 FPN 的改进算法^{[13][35][36]}应运而生。

最后，采用超分辨技术也是一种提高小目标检测精度的有效方法。感知生成对抗网络（Perceptual GAN^[37]）将生成对抗网络（GAN^[38]）应用于超分辨率技术上，挖掘小目标与常规目标之间的结构关联。感知生成对抗网络通过学习小目标与常规目标之间的映射关系，缩小不同尺度目标之间的特征差异，使小目标与常规目标有相似的特征表示，以达到提升小目标检测效果的目的。此后，也出现了一些其他基于 GAN 网络的改进方法，如 MTGAN^[39]。GAN 网络的优势在于，能够生成小目标特征相对明显且分辨率高的图像，从而对数据集进行一定的补充。

综上，目前目标检测的主流算法是基于深度学习的，提升小目标检测的精度可以通过数据增强、多尺度融合、超分辨技术等方式。受到前人研究成果的启发，本文提出的改进算法利用 Mosaic 数据增强、多尺度特征融合方法，在 YOLOv5 目标检测算法基础之上，进行了一系列的改进。以 YOLOv5s 和 YOLOv5m 为基准，我们的改进算法在提升中、小型目标检测精度的同时，也基本维持了推理速度。算法的具体内容将在第 3、4 章详细阐述。

第2章 基于深度学习的目标检测算法与特征融合结构

2.1 引言

本章的主要内容将分为两大部分：基于深度学习的目标检测算法综述以及几类典型特征融合结构的介绍。

根据检测方式的不同，基于深度学习的目标检测算法可以分为两阶段（two-stage）算法和一阶段（one-stage）算法。两阶段算法基于提议的候选区域，先提取目标的候选框，再基于候选框二次修正得到分类和回归的结果；一阶段算法基于边界框的回归，在产生边界框的同时进行分类和回归。首先我们介绍两阶段算法，按照时间顺序，依次介绍 R-CNN、SPPNet、Fast R-CNN、Faster R-CNN 算法，概括每种算法主要解决的问题以及有待后续方法改进的地方；其次我们介绍一阶段算法，依次介绍 SSD 和 YOLO 系列算法，重点介绍 YOLO 算法的基本原理和网络结构，为本文第 3、4 章提出的基于 YOLOv5 的改进算法做铺垫。

通过对主流目标检测算法进行分析，我们发现现阶段目标检测算法对小尺度目标的检测效果相对不佳。考虑到多尺度特征融合是一种改进小目标检测的有效措施，我们介绍了三种典型的特征融合结构，即 FPN、PANet 和 BiFPN，分析每一种特征融合结构的特点和作用，并引出第 4 章的内容：将 YOLOv5 与 BiFPN 结构结合，提升原始 YOLOv5 算法的检测效果。

2.2 基于深度学习的典型目标检测算法

2.2.1 两阶段（two-stage）目标检测算法

2.2.1.1 R-CNN 算法

R-CNN(区域卷积神经网络)^[4]是基于卷积神经网络目标检测算法的奠基之作。R-CNN 算法的主要步骤如下：首先，采用“选择性搜索^[40]”算法产生若干个可能包含物体的区域；其次，利用卷积神经网络（如 AlexNet^[27]）在选取的区域上进行特征提取；接着使用支持向量机（SVM）^[29]分类器进行分类；最后采用边界框回归的方式进行准确的定位。

尽管 R-CNN 算法已经取得了很大的进步，但它仍然存在一些有待改进的问题。例如：模型的训练不是以端到端的方式，而是分多个步骤在不同平台上分别进行的；训练过程和推理过程在空间和时间上的消耗都尤为巨大。

2.2.1.2 SPPNet 算法

空间金字塔池化网络（SPPNet）^[5]主要解决的问题是：R-CNN 算法中，对于每一个候选区域来说，都需要将图像缩放为固定的尺寸（如 AlexNet 中固定的尺寸大小是 224×224）作为网络的输入，该做法可能导致图像失真。因此，何凯明团队提出的 SPPNet 在网络结构的全连接层之前加入空间金字塔池化（SPP）层，该层对不同尺度的特征图进行池化并生成固定长度的特征表示，避免缩放图像导致的几何形变，进而影响目标检测网络的性能。

与 R-CNN 相比，SPPNet 虽然有效地提高了精度，但依然存在不足：SPP 结构只作用于最后的全连接层之前，忽略了对 SPP 层之前的网络进行改进；与 R-CNN 相同，训练过程分为多步且单独运行，仍需要巨大的存储空间。

2.2.1.3 Fast R-CNN 算法

Fast R-CNN^[6]是对 R-CNN 算法和 SPPNet 算法的进一步改进，是一个更快版本的 R-CNN 算法。Fast R-CNN 算法将整张图像作为输入来提取特征，同时引入一个新的概念：感兴趣区域池化层（ROI）。ROI 层从不同尺寸的提议区域中提取到固定大小的特征，这些特征再作为后续用于分类和回归的全连接层的输入。本质上来说，Fast R-CNN 不再像 R-CNN 那样对每一个提议区域分别卷积，而是直接对整张图像做卷积操作，并利用 ROI 池化及映射关系得到统一大小的特征，这样就减少了大量的重复计算。PASCAL VOC 2007 数据集上的实验结果显示：与 R-CNN 相比，Fast R-CNN 的平均精确度（mAP）从 58.5% 提升到 70.0%，速度也提升了近 200 倍。虽然 Fast R-CNN 实现了多任务端到端训练，但美中不足的是，用于产生候选区域的选择性搜索算法在实际过程中依然十分耗时，需要更加高效的候选框提议方法来替代选择性搜索。

2.2.1.4 Faster R-CNN 算法

Ren 等人提出的 Faster R-CNN^[7]算法，在卷积层后面添加区域提议网络（RPN），代替 Fast R-CNN 的选择性搜索方法。RPN 是全卷积网络，它将输出结果映射为一个概率值和四个坐标值，将二分类损失、边界框回归损失统一起来作为训练过程的目标损失函数。PASCAL VOC 2007 数据集上的实验结果表明：在相同的 VGG 骨干网络下，Faster R-CNN 与 Fast R-CNN 相比，精度（mAP）相差不大，但训练时间和测试时间都大大缩短了。Faster R-CNN 在速度上的提升效果明显，但仍存在计算冗余，需要后续进一步的改进和加强。

2.2.2 一阶段 (one-stage) 目标检测算法

2.2.2.1 SSD 算法

SSD (单发多框检测器)^[8]算法在不同尺度的特征图上进行预测,训练过程同样是端到端的。与 Faster R-CNN 相比,SSD 算法去掉了生成提议区域这一步骤,能够做到直接预测目标的类别和位置,从而极大地提高了运算的速度,因此 SSD 算法也被归类为一阶段检测算法。SSD 算法效果较好的原因可以归结为以下三点:使用多尺度特征图预测、设置了多种宽高比的默认框以及运用了数据增强技术。

首先,SSD 的网络结构中,分别使用 6 种尺度不同的特征图 (38×38 、 19×19 、 10×10 、 5×5 、 3×3 、 1×1) 来预测不同大小的目标;其次,SSD 引入了默认框。默认框是指,特征图上的每一个小格都匹配一系列固定大小的框。实验表明,默认框的形状和数量越多,检测效果越好。默认框在不同的特征层有不同的尺度,在相同的特征层又有不同的宽高比,因此基本能够覆盖输入图像中各样形状和大小的目标;最后,SSD 采用缩小操作等数据增强的方式,缩小操作和随机剪裁能够产生更多尺度较小的目标,提高模型对不同尺度目标的泛化能力。SSD 算法在速度和精度上都有一定的优势,但主要的缺点是:对于小目标检测的效果还是一般,主要原因是小目标在高层级没有足够多的特征。

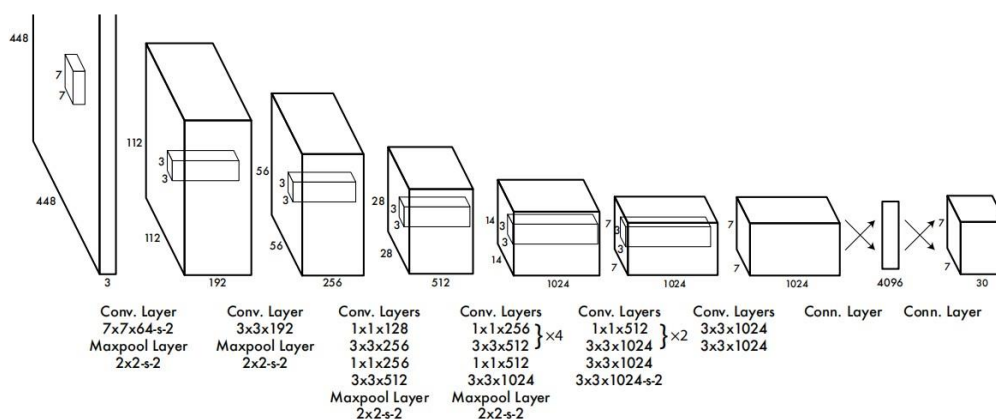
2.2.2.2 YOLO 系列

(一) YOLOv1 算法

YOLOv1 (You Only Look Once)^[9]可以说是一阶段目标检测算法的开山之作,也引发了后续一系列改进的 YOLO 算法的产生。YOLO 算法最显著的优势是计算速度非常快。在 PASCAL VOC 2007 数据集上,YOLOv1 的快速版本能够达到精度 $mAP=52.7\%$ 的同时运行速度为 155fps;增强版本能够达到精度 $mAP=63.4\%$ 的同时运行速度为 45fps。YOLOv1 的核心思想是:将目标检测问题统一成一个回归算法,用一个卷积神经网络结构从输入图像直接预测框的位置和类别概率。

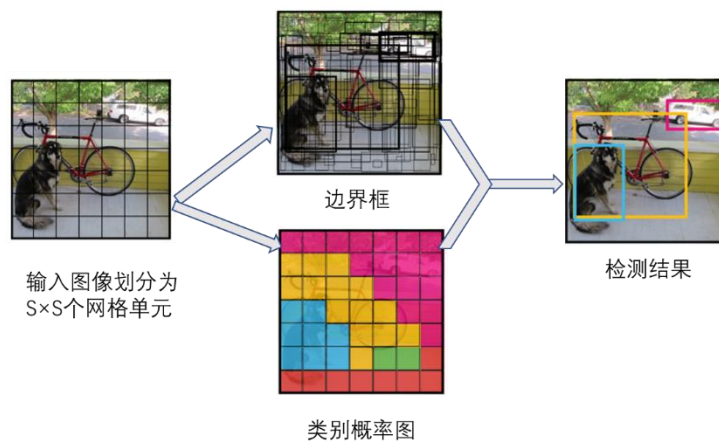
(1) 网络结构

算法的骨干网络结构如图 2.1 所示,与 GoogleNet 相类似,YOLOv1 的网络由 24 个卷积层、4 个池化层和 2 个全连接层构成。首先,我们需要把输入图像的大小统一为 448×448 ,然后经过多个卷积层和池化层得到 $7\times7\times1024$ 维的张量,最后经过两个全连接层输出 $7\times7\times30$ 维的张量。其中,卷积层用来提取特征,全连接层用来获取预测的类别概率和位置信息。

图 2.1 YOLOv1 网络结构^[9]

(2) 基本原理

我们通过解释输出张量数值的含义，说明 YOLOv1 的基本原理。网络经过最后一个全连接层，得到 7×7×30 的输出张量。7×7×30 中的 7×7 含义为：如图 2.2 左图所示，将输入图像分成 S×S 个网格单元（参数 S 可以自行选择，YOLOv1 选用的是 S=7）。如果某类物体的中心点落在了 7×7 个网格单元中的某个时，那么该网格将负责预测这类物体。以图 2.2 的左图为例，假设左下角网格的坐标为(1,1)，小狗所在的最小包围框的中心点为(2,3)，那么在这些 7×7 的网格单元中，网格(2,3)负责预测小狗。

图 2.2 YOLOv1 算法流程^[9]

输出张量 7×7×30 指最终每个网格都有一个 30 维度的输出，其中 30 是由 (4+1)×2+20 得到的。每一个网格单元产生 B 个矩形框（YOLOv1 取 B=2），每个矩形框预测 4 个位置信息值（物体中心点的坐标 x, y 及框的长宽 w, h）和 1 个置信度值 c。置信度 c 的计算公式如公式(2.1)所示：

$$c = Pr(object) \times IoU_{pred}^{truth} \quad (2.1)$$

其中, $Pr(object)$ 的值取 0 或 1, 即当物体的中心落在网格单元内, 则置信度值为真实框与预测框之间的 IoU 值; 而物体中心不在该网格时, 则置信度为 0。此外, 每一个网格单元还预测 C 个类别 (如 PASCAL VOC 2007 总共有 20 类物体, 所以 $C=20$) 的概率分数, 即 $Pr(class|object)$ 。综上, 输出的 $7 \times 7 \times 30$ 维张量提取了类别概率和边界框回归信息。注意, 置信度是针对边界框的, 而类别概率是针对网格单元的。

最后根据之前步骤, 预测 $7 \times 7 \times 2 = 98$ 个目标窗口, 然后根据阈值 (通常取 0.2) 去除可能性比较低的目标窗口, 再经过非极大值抑制 (NMS) 去除冗余窗口即可。

测试阶段, 将每个框的置信度与每个类别概率相乘, 从而得到每个框属于哪一个类别的置信度得分, 如公式(2.2)所示。

$$\begin{aligned} Pr(Class_i|object) \times Pr(object) \times IoU_{pred}^{truth} \\ = Pr(Class_i) \times IoU_{pred}^{truth} \end{aligned} \quad (2.2)$$

(3) 损失函数

YOLOv1 损失函数的定义如(2.3):

$$\begin{aligned} Loss = & \lambda_{coord} \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 \\ & + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] + \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} ((C_i - \hat{C}_i)^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbf{1}_{ij}^{noobj} ((C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S \times S} \mathbf{1}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \quad (2.3)$$

损失函数公式的前两项表示的是对边界框的位置预测, 其中 $\mathbf{1}_{ij}^{obj}$ 表示示性函数, 即当第 i 个网格的第 j 个边框负责某个目标时结果为 1, 其余情况为 0。公式的第 3、4 项表示框的置信度预测, 最后一项表示类别的预测, 示性函数 $\mathbf{1}_{ij}^{obj}$ 用来判断目标的中心是否落在了网格中心 i 上。

(二) YOLOv2 (或称 YOLO9000) 算法

YOLOv2^[10]对原来的 YOLOv1 检测框架进行了改进,在保持原来速度的基础之上,做到了预测的效果更好、速度更快、识别对象的种类更多。YOLOv2 采用联合训练算法,即利用分类数据集(如 ImageNet)的数据学习更多类别的分类、利用检测数据集(如 MS COCO)的数据学习目标的位置信息,提高模型的能力。此外,YOLOv2 采用了许多的技巧,每一种技巧均对模型的性能产生了一定的改进,改进结果如表 2.1 所示。

YOLOv2 算法的主要技巧有:①批量归一化:对中间层的特征分别进行归一化处理,使特征的数据分布变换为均值为 0、方差为 1,再增加线性变换使其恢复原本的数据表达能力。该做法能够有效解决反向传播过程中的梯度消失和梯度爆炸问题。②高分辨率分类器:YOLOv1 在 ImageNet 上预训练的输入分辨率为 224×224 ,检测的输入分辨率为 448×448 ,这样的切换会对模型性能有一定的影响。因此 YOLOv2 在输入大小为 224×224 的预训练之后,再利用 448×448 的高分辨率样本对模型进行微调,使网络能够逐渐适应高分辨率的输入,缓解分辨率的突然切换带来的影响。③使用先验框:每一个单元网格预先设置 9 个不同大小和宽高比的先验框,虽然 mAP 轻微下降 0.2 但召回率大幅度提升。④用聚类方式选取先验框:对训练集中标注的边框采取 K-means 聚类方法,选取更加匹配样本尺寸的先验框。⑤约束边框的位置预测:YOLOv2 调整了预测边框位置的公式,将中心点坐标的预测约束在了特定的网格单元内,使得数值更加参数化,网络更加稳定和易于学习。⑥多尺度训练:网络训练过程中会从 32 的倍数 {320, 352.....608} 中随机选择输入大小,使得同一网络能在不同分辨率的输入情况下依然达到较好的效果。

	YOLOv1									YOLOv2
批量归一化		✓	✓	✓	✓	✓	✓	✓	✓	✓
高分辨率分类器			✓	✓	✓	✓	✓	✓	✓	✓
卷积				✓	✓	✓	✓	✓	✓	✓
使用先验框				✓	✓					
DarkNet 网络					✓	✓	✓	✓	✓	✓
聚类方式选取先验						✓	✓	✓	✓	✓
约束边框的位置预						✓	✓	✓	✓	✓
直通层							✓	✓	✓	✓
多尺度训练								✓	✓	✓
高分辨率分类器									✓	✓
VOC2007 mAP (%)	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8		78.6

表 2.1 YOLOv2 算法的主要改进技巧

（三）YOLOv3 算法

改进的 YOLOv3^[11]可以做到精度和 SSD 算法相近，而速度快了将近 3 倍。YOLOv3 算法的改进之处为三个方面：应用多尺度的预测、采用全新的网络结构 DarkNet-53 以及分类损失采用二元交叉熵损失。

网络结构方面：YOLOv3 借鉴 ResNet 的思想，采用了大量的残差跳跃连接层，设计并训练了新的网络 DarkNet-53。同时，为了降低池化带来的梯度负面影响，网络采用全卷积层并通过调节卷积的步长实现下采样。

为了加强算法对小目标的检测精度，YOLOv3 采用特征金字塔网络（FPN）进行多尺度特征融合。若输入图像大小为 $416 \times 416 \times 3$ ，则输出三条预测支路。三条预测支路的特征图尺度分别是 $13 \times 13 \times 255$ 、 $26 \times 26 \times 255$ 以及 $52 \times 52 \times 255$ ，采用多尺度的方式对不同大小的目标进行检测，其中 255 的含义为： $255 = 3 \times (4 + 1 + 80)$ ，3 表示 1 个网格单元预测 3 个边界框，4 表示边界框坐标的 4 个信息，1 表示边界框的置信度，80 表示 MS COCO 数据集共 80 个类别。

2.3 典型的特征融合网络结构

2.3.1 引言

尽管近年来目标检测任务取得了巨大的进展，但目标检测的尺度问题对于深度学习算法而言也始终是个难题，即模型面对尺度过大或者过小的目标时，检测的性能会相对下降。针对多尺度目标检测的问题，之前的论文提出过几种利用特征的方式，如下图 2.3 所示，本文举其中的 4 个例子进行简单的介绍。

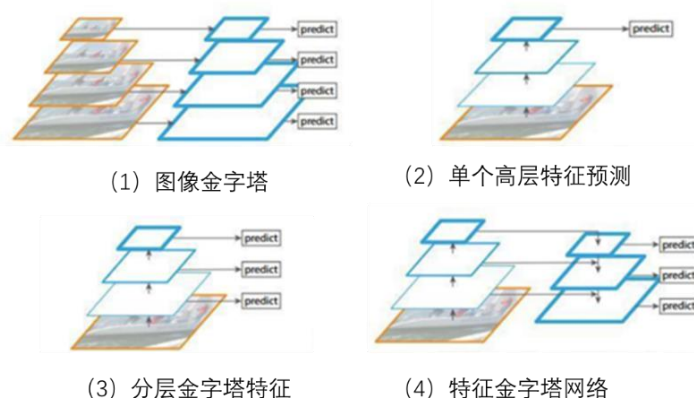


图 2.3 常用的几种特征利用形式

（1）图像金字塔结构：如图 2.3(1)，利用图像金字塔来构建不同尺度的特征金字塔，该方法对于不同尺度的图像进行独立的特征提取，从而产生多尺度的特征表示，且所有层级的特征图都保留了较强的语义信息。然而，图像金字塔方法的缺

陷在于,推理的时间大幅度增加,内存占用过大,使得端到端地进行神经网络训练的难度加大。

(2) 单个高层特征的预测:例如 SPPNet、Faster R-CNN 算法,利用单个高层特征进行目标的分类和回归,如图 2.3(2)。

(3) 分层金字塔特征:如图 2.3(3),该方法的一个典型的应用是 SSD 目标检测算法。由于卷积神经网络计算时本身就会产生多层级的特征图,且不同层特征图的尺度不一定相同,所以该方法提出可以重用卷积网络计算的金字塔层级结构,类似于特征化的图像金字塔。然而,SSD 方法为了避免使用低层级的特征,直接放弃了浅层的特征。对于小目标检测来说,舍弃高分辨率特征会导致检测效果不佳。

(4) 特征金字塔网络(FPN):为了解决以上三种方法的不足,FPN 自然地利用 CNN 层级特征的金字塔形式,不同层级的尺度不同的特征图都具有较强的语义信息,如图 2.3(4)。FPN 结构设计出自上而下的网络结构和横向连接,将具有高分辨率的低层特征与具有丰富语义信息的高层特征融合,FPN 结构的具体形式将在下节详细介绍。

2.3.2 特征金字塔网络(FPN)

特征金字塔网络(FPN)^[12]的提出旨在解决目标检测任务在处理多尺度变化问题上的不足。如前文所提到的,在以往的目标检测(如 Faster R-CNN)过程中,无论是区域提议网络(RPN)还是感兴趣的区域(ROI),基本都是作用于网络的最后一层。对于小目标而言,其自身所具有的像素数量较少,在反复下采样的过程当中很容易丢失信息,所以当卷积层池化层进行到最后的层级时,有意义的语义信息已经相对较少了,因此这类网络起到的作用是非常有限的。

FPN 结构利用自上而下的路径和横向连接,将高分辨率的低层特征与多语义信息的高层特征进行融合。FPN 利用 CNN 的前馈计算,以 ResNet 为例,将每个阶段的最后一个残差结构的特征激活作为输出,分别记为 $\{C_1, C_2, C_3, C_4, C_5\}$ 。特征 C_i 的大小是输入图像大小的 $1/2^i$ 。例如,如果输入图像的分辨率为 640×640 ,那么 C_3 表示第3层级的特征,大小为 80×80 。多尺度特征产生的具体步骤为:如图 2.4,以 P_4 层特征的生成为例,首先将 C_5 层经过 1×1 的卷积层改变特征图的通道数得到 F_5 , F_5 先经过上采样(通常选用最近邻插值法),再与 C_4 经过 1×1 卷积得到的特征图按特征图上的每一个相同位置元素相加得到 F_4 , F_4 再经过 3×3 卷积得到最终的 P_4 层特征。其他层级根据同样的原理,可以得到最终的金字塔特征 $\{P_2, P_3, P_4, P_5\}$ 。FPN 结构及各层特征获取过程如图 2.4 所示。

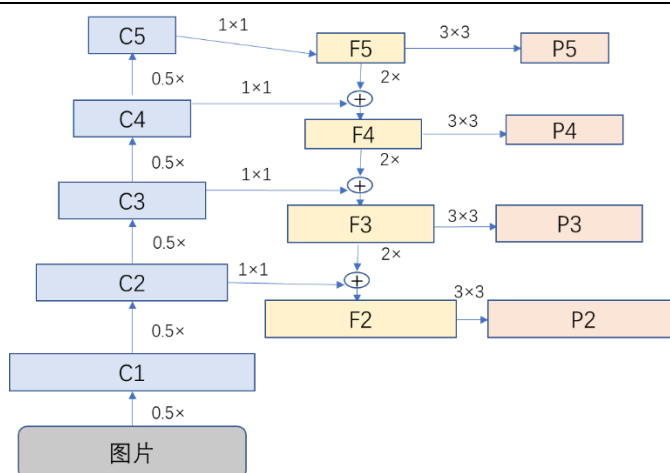


图 2.4 FPN 网络结构图

2.3.3 路径聚合网络（PANet）

传统的自上而下的 FPN 结构本质上受到了信息单向传递的限制，为了解决这一问题，路径聚合网络（PANet）^[13]添加了一个额外的自下而上的聚合网络。FPN 的自上而下结构，将高层级丰富的语义信息传递下来，对金字塔特征进行了增强。然而，增强的特征只是语义信息，而忽略了对定位信息的传递。因此，PANet 结构对 FPN 结构进行了补充，在 FPN 后面添加了自下而上的金字塔结构。如图 2.5 所示，类似于 FPN 的计算，每一层的 N_i 特征首先经过步长为 2 的 3×3 卷积进行下采样，再通过横向连接，将 P_{i+1} 特征图中的每一个元素与 N_i 下采样后得到的特征图相加，得到的结果经过 3×3 卷积生成 N_{i+1} 层特征。

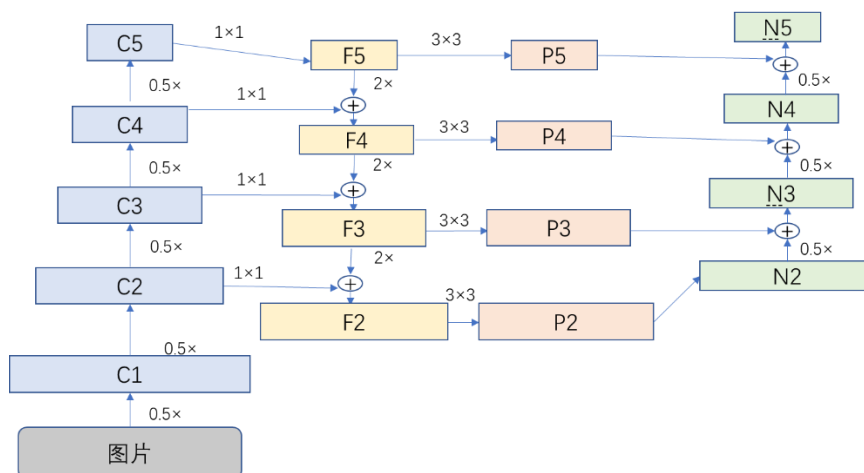


图 2.5 PANet 网络结构图

2.3.4 双向特征金字塔网络（BiFPN）

双向特征金字塔网络（BiFPN）是由谷歌大脑团队在 EfficientDet^[14]目标检测算法中首次提出。该网络结构意图解决网络多尺度特征融合问题，BiFPN 的主要思

想可以概括为：有效的双向跨尺度连接和加权特征融合。当融合具有不同分辨率的特征时，目标检测算法中以往的特征融合方式都是先将它们调整到相同的分辨率，再对它们进行加和。以前的方法大多是平等地对待不同尺度的特征，即以相同的权重进行特征融合。然而我们发现，不同分辨率的输入特征通常对输出特征的贡献是不均等的。为了解决这个问题，BiFPN的想法是为每个输入增加一个额外的权重，并让网络学习每个输入特征的重要程度。基于这种思想，我们对权重进行三种类型的讨论：

(1) 无界的融合：

$$O = \sum_i w_i \cdot I_i, \quad (2.4)$$

无界融合的方式如公式(2.4)，其中 w_i 是可学习的权重，对每一个特征而言 w_i 是标量，对每一个通道而言 w_i 是向量，而对于每一个像素点来说 w_i 则是多维张量。然而，该做法可能导致训练不稳定，因为标量形式的权重是无界的。因此，可以选择归一化权重的思想来约束每一个权重的取值范围。

(2) 基于 Softmax 的融合：

$$O = \sum_i \frac{e^{w_i}}{\sum_j e^{w_j}} \cdot I_i \quad (2.5)$$

基于 Softmax 的融合方式如公式(2.5)，产生每一个权重的方式可以借鉴 Softmax 的思想，这样所有的权重都被归一化为一个 0 到 1 之间的概率值，以此体现每一个输入的重要性。然而，额外的 Softmax 计算会导致 GPU 硬件的明显减速。为了最小化额外的延迟时间成本，进一步可以提出一种快速融合的方法。

(3) 快速归一化融合：

$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i \quad (2.6)$$

快速归一化融合方式如公式(2.6)，其中 $w_i \geq 0$ 且取 $\epsilon = 0.0001$ （为避免数值不稳定），同样每一个归一化后的权值范围也在 0 到 1 之间。由于这里没有用到 Softmax 操作，所以效率要高很多。实验结果显示：这种快速融合方法与基于 Softmax 融合方法相比，二者具有相近的学习能力和准确率，但快速融合方式在 GPU 上的运行速度要快 30%。

最终的 BiFPN 结构集成了跨尺度连接和快速归一化融合。EfficientDet 中，骨干网络所利用的特征层级为 $C_3 \sim C_7$ ，完整的 BiFPN 特征融合方式如图 2.6 所示。

我们以 BiFPN 结构第 6 层级的融合特征 N_6 为例，介绍 BiFPN 融合特征的思路。参考图 2.6，可以看到 N_6 生成过程中需要用到的特征有 F_6 、 P_6 和 N_5 。公式(2.7)和(2.8)描述了生成特征 N_6 的过程，所有其他层级的特征都是以类似的方式构造的。

$$P_6 = \text{Conv} \left(\frac{w_1 \cdot F_6 + w_2 \cdot \text{Resize}(F_7)}{w_1 + w_2 + \epsilon} \right) \quad (2.7)$$

$$N_6 = \text{Conv} \left(\frac{w'_1 \cdot F_6 + w'_2 \cdot P_6 + w'_3 \cdot \text{Resize}(N_5)}{w'_1 + w'_2 + w'_3 + \epsilon} \right) \quad (2.8)$$

综上，BiFPN 结构通过加权特征融合和双向跨尺度连接的方式，加强了多尺度特征的融合，丰富了特征的语义信息。受此启发，本文提出在 YOLOv5 的基础之上添加调整后的 BiFPN 结构，使目标检测算法具有更强的性能，算法的具体内容详见第 4 章。

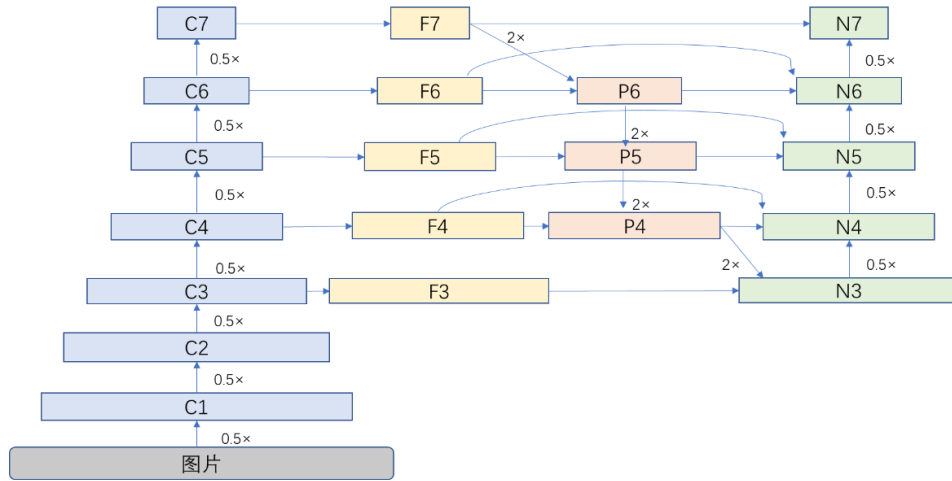


图 2.6 BiFPN 网络结构图

第3章 YOLOv5 目标检测算法

3.1 引言

本章节,我们将介绍 YOLO 系列的最新算法:YOLOv5 目标检测算法。YOLOv5 与 YOLOv4 实质上都是在 YOLOv3 算法的基础之上,进行了网络结构及训练技巧等方面的改进,使得检测性能得到进一步的提升。

YOLO 系列目标检测的框架,通常可以分为如下几个部分: 输入端、骨干网络、Neck 网络和输出端。本章将具体阐述 YOLOv5 算法的改进之处,主要内容如下:(1)输入端的改进,主要增加了 Mosaic 数据增强方法和自适应图片缩放方法。

(2)骨干网络的改进,考虑了 Focus 结构和 CSPNet 结构。(3)输出端的改进,主要对损失函数的构建进行了研究。

3.2 数据增强

YOLOv5 算法实验部分采用的数据增强方式是马赛克(Mosaic)方法, Mosaic 方法原理上与 CutMix^[20]方法相类似。CutMix 运用的是两张图片,而 Mosaic 方法运用了四张图片。我们先来说明 CutMix 数据增强的原理。

如图 3.1(左)所示, CutMix 方法是将图像的一部分区域剪裁掉,但不是以 0 像素填充,而是随机填充训练集中的其他数据图像部分区域的像素值,分类结果按一定的比例分配。原理如下: x_A 和 x_B 是训练集中两张不同的图片, y_A 和 y_B 是与之相对应的标签值。经过 CutMix 之后的新训练样本 \tilde{x} 和对应的标签 \tilde{y} 的生成方法如公式所示(3.1):

$$\begin{aligned}\tilde{x} &= M \odot x_A + (1 - M) \odot x_B \\ \tilde{y} &= \lambda y_A + (1 - \lambda) y_B\end{aligned}\quad (3.1)$$

其中, $\mathbf{1}$ 指所有元素都为 1 的二进制掩码, λ 服从 Beta 分布, $M \in \{0,1\}^{W \times H}$, $W \times H$ 指的是图像的大小。 M 是一个与原图尺寸一致的由 0 和 1 标记的掩码矩阵,它标记了需要裁剪的区域和保留的区域,裁剪的区域值均为 0,其余位置为 1。为了对掩码矩阵 M 进行采样,需要先对裁剪区域的边界框 B 进行采样。我们将边界框 B 记作 $B = (r_x, r_y, r_w, r_h)$, 其中 (r_x, r_y) 指裁剪区域的中心点坐标, r_w 和 r_h 指裁剪区域的宽和高,采样公式如(3.2)所示:

$$\begin{aligned}r_x &\sim \text{Unif}(0, W), \quad r_w \sim W\sqrt{1 - \lambda}, \\ r_y &\sim \text{Unif}(0, H), \quad r_h \sim H\sqrt{1 - \lambda}\end{aligned}\quad (3.2)$$

剪裁区域 B 确定后,将区域内的掩码值置为 0,其余区域置为 1。按照公式将图片 A 中相对于区域 B 的部分擦除,并将图片 B 中相对于区域 B 的部分填充到图片 A 中,完成 CutMix 操作。

Mosaic 数据增强参考了 CutMix 方法,理论上相似,但 Mosaic 利用了四张图片,采用随机剪裁、缩放、排布的方式进行拼接,如图 3.1 (右)所示。Mosaic 数据增强的优点是更加丰富了检测物体的背景,且在批量标准化计算的时候会一次性计算四张图片的数据,使得 Mini-batch 的设置不需要很大,那么使用较少 GPU 即可达到较好的效果。同时,Mosaic 方法也能够提高小目标的检测效果。MS COCO 数据集的大、中、小目标分布不均,而 Mosaic 方法利用随机缩放进行拼接,增加了小目标的数据,丰富了数据集,从而使网络更加鲁棒。



图 3.1 CutMix 方法 (左图), 第一行两张图片为训练集中的两个样本;
第二行为 CutMix 的过程; Mosaic 方法 (右图)

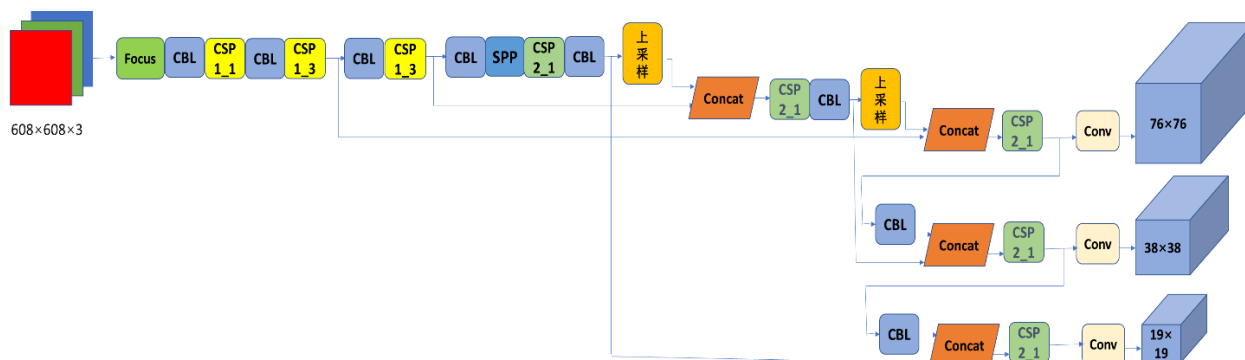


图 3.2 YOLOv5 的网络结构

3.3 网络结构

3.3.1 YOLOv5 网络结构

目前 YOLOv5 算法共有 4 个版本,分别是: YOLOv5s、YOLOv5m、YOLOv5l 和 YOLOv5x。本节以 YOLOv5s 为例进行说明,其他版本的结构原理

与之类似，是在 YOLOv5s 的基础之上对网络加宽。YOLOv5s 目标检测算法的整体网络结构如图 3.2 所示。YOLOv5s 的输入图像大小为 608×608 ，下面分别介绍网络结构的每一个组件：

(1) CBL 模块：该模块由卷积操作（Conv）、批量标准化操作（BN）和激活函数（Leaky-Relu^[50]）三部分组成，如图 3.3 所示。



图 3.3 CBL 模块结构

(2) Focus 模块：如图 3.4 所示，该结构将输入图像进行切片操作，再将切片后的结果拼接。Focus 结构的内容将在 3.3.2 节具体说明。

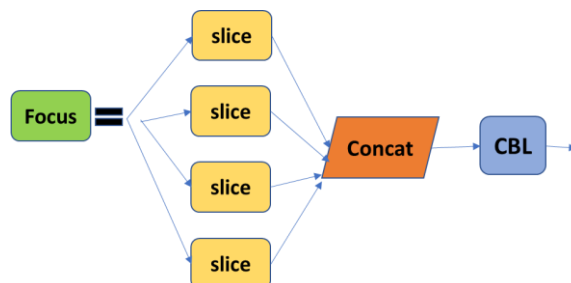


图 3.4 Focus 模块结构

(3) SPP 模块：如图 3.5 所示，该结构分别采用 1×1 、 5×5 、 9×9 和 13×13 的最大池化，再将所得的结果拼接，得到融合后的特征。

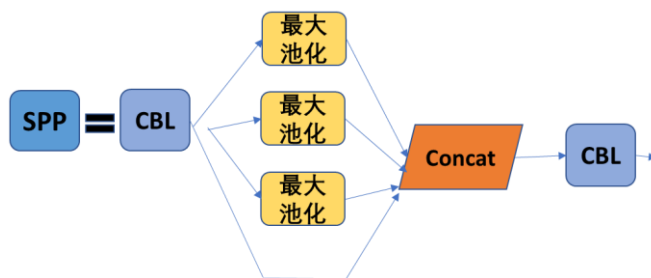


图 3.5 SPP 模块结构

(4) CSP1_X 和 CSP2_X 模块：借鉴 CSPNet 的思想（CSPNet 将在 3.3.3 节具体说明），该模块由卷积层、CBL 模块和 Res Unit 模块三部分组成。其中，Res Unit 模块借鉴 ResNet 残差结构，由 CBL 模块组成，如图 3.6 所示。CSP1_X 和 CSP2_X 模块结构如图 3.10 和 3.11 所示。

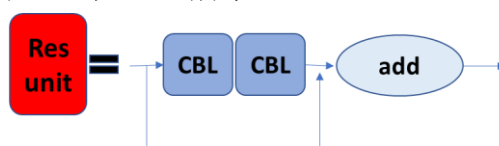


图 3.6 Res Unit 模块结构

3.3.2 Focus 结构

与 YOLOv4 相比, Focus 结构是 YOLOv5 中特有的结构, 其中的关键是切片操作。Focus 结构是指, 在图片进入骨干网络之前, 将图片每隔一个像素取一个值, 与邻近下采样操作相似, 即可得到四组图片。Focus 操作将图像的宽和高信息集中到了通道空间, 通道的维数扩大为原来的 4 倍, 即拼接后的结果相当于是将原先的 RGB 三通道模式变成了 12 个通道, 最后将得到的结果再经过卷积操作, 得到最终的二倍下采样的特征图。切片操作的过程如下图 3.7 所示。

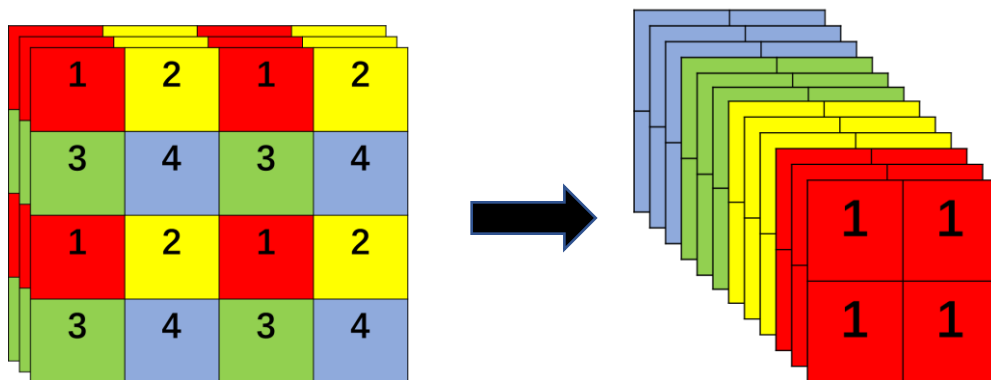


图 3.7 Focus 结构中的切片操作过程

我们发现, 图片在经过 Focus 结构后, 最直观的是它起到了下采样的作用, 但是 Focus 操作和常用的卷积下采样又略有不同。我们可以对 Focus 的计算量和普通卷积的下采样计算量进行对比, 计算量以浮点运算数 (FLOPs) 为评估标准。以 YOLOv5s 为例, 输入图像大小为 $608 \times 608 \times 3$, 计算量的对比结果如下:

(1) 如果采用普通卷积下采样, 卷积核大小是 3×3 , 步长为 2, 输出通道数为 32, 则经过下采样后得到的特征图维数是 $304 \times 304 \times 32$, 那么普通卷积下采样无偏差时的计算量理论上的结果如公式(3.3)所示。

$$FLOPs(Conv) = 3 \times 3 \times 3 \times 32 \times 304 \times 304 \quad (3.3)$$

(2) 如果采用 Focus 结构, 先采用切片操作, 特征图大小变为 $304 \times 304 \times 12$, 再经过一次卷积核大小为 3×3 的卷积操作, 最终输出的特征图大小为 $304 \times 304 \times 32$, 过程如图 3.8 所示。Focus 结构无偏差时的计算量理论上的结果如公式(3.4)所示。

$$FLOPs(Focus) = 3 \times 3 \times 3 \times 4 \times 32 \times 304 \times 304 \quad (3.4)$$

可以看到, 采用 Focus 结构的计算量要比普通卷积下采样的计算量多一些, 大约是普通卷积下采样的 4 倍, 但 Focus 结构的好处是下采样的过程中减少了信息的丢失。综上所述, YOLOv5 算法采用 Focus 结构的主要作用在于, 图片进行下采样的过程中, 将宽和高的信息集中到了通道维度上, 再利用卷积层进行特征提取, 使得特征提取更加充分。虽然带来了一些计算量的增加, 但减轻了信息的丢失现象。

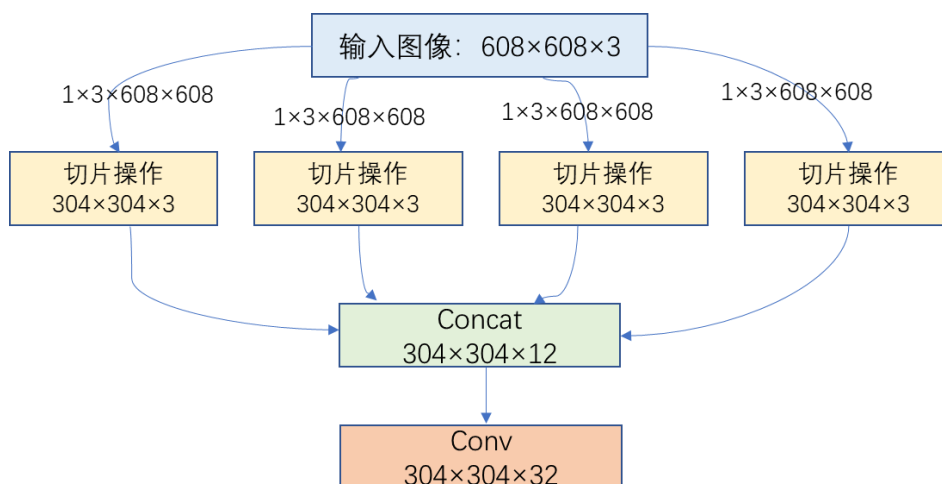


图 3.8 Focus 结构示意图

3.3.3 CSP 结构

CSPNet^[16]（Cross Stage Partial Network）主要从网络结构的设计方面减少推理时的计算量。CSPNet 的提出主要做出了以下改进：增强 CNN 的学习能力，能够在轻量化的同时维持准确性；降低计算瓶颈以及降低内存成本。CSPNet 将底层的特征图按通道拆分为两部分，一部分经过密集块（由多个全连接层组成）和过渡层（通常是卷积核大小为 1×1 的卷积层），另一部分与传输的特征图结合，在减少计算量的同时也提高了推理的速度和准确性。CSPNet 不仅是一种网络结构，也是一种处理思想，它可以与 ResNet、ResNeXt 以及 DenseNet 等网络进行结合。以 DenseNet 为例，DenseNet 与 CSPDenseNet 的对比示意图如图 3.9 所示。

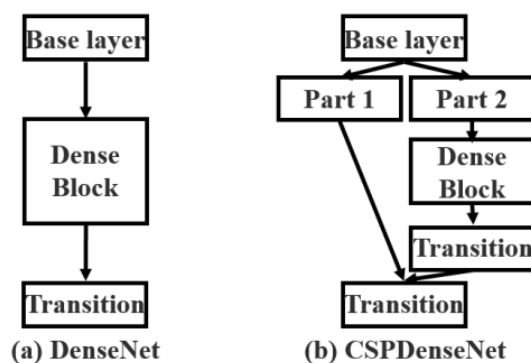


图 3.9 CSPNet 结构示意图: (a)原始 DenseNet 结构, (b)改进后的 CSPDenseNet 结构

如图 3.9(a)，以 DenseNet 为例，前向传播过程如公式(3.5)，反向传播权重的更新过程如公式(3.6)。其中，“*”代表卷积算子， $[x_0, x_1, \dots, x_k]$ 表示将 x_0, x_1, \dots, x_k 进行拼接， f 是更新权重的函数， g_i 表示传递到第 i 个全连接层的梯度， w_i 和 x_i 分别是第 i 个全连接层的权重和输出。

$$\begin{aligned}
x_1 &= w_1 * x_0 \\
x_2 &= w_2 * [x_0, x_1] \\
&\dots \dots \\
x_k &= w_k * [x_0, x_1, \dots, x_{k-1}]
\end{aligned} \tag{3.5}$$

$$\begin{aligned}
w_1' &= f(w_1, g_0) \\
w_2' &= f(w_2, g_0, g_1) \\
w_3' &= f(w_3, g_0, g_1, g_2) \\
&\dots \dots \\
w_k' &= f(w_k, g_0, g_1, g_2, \dots, g_{k-1})
\end{aligned} \tag{3.6}$$

我们发现,在利用反向传播更新全连接层之间的权重的计算过程中,大量的梯度信息是重复利用的,导致不同全连接层重复学习相同的梯度信息。为了解决这一问题,CSPDenseNet 将底层特征按通道分为两部分,记作 $x_0 = [x_0', x_0'']$ 。 x_0' 直接连接到该阶段的末尾,而 x_0'' 需要经过密集块和过渡层。如图 3.9(b),密集块的输出 $[x_0'', x_1, \dots, x_k]$ 经过过渡层,得到输出 x_T 。 x_T 与 x_0' 拼接后经过过渡层,得到输出结果 x_U 。CSPDenseNet 前向传播过程如公式(3.7),反向传播权重的更新过程如公式(3.8)。

$$\begin{aligned}
x_k &= w_k * [x_0'', x_1, \dots, x_{k-1}] \\
x_T &= w_T * [x_0'', x_1, \dots, x_k] \\
x_U &= w_U * [x_0', x_T]
\end{aligned} \tag{3.7}$$

$$\begin{aligned}
w_k' &= f(w_k, g_0'', g_1, g_2, \dots, g_{k-1}) \\
w_T' &= f(w_T, g_0'', g_1, g_2, \dots, g_k) \\
w_U' &= f(w_U, g_0', g_T)
\end{aligned} \tag{3.8}$$

可以看到,两边路径的梯度是单独集成的。更新权重的梯度信息时,两边都不包含属于另一边的重复梯度信息。CSPNet 通过截断梯度流,防止梯度信息过多地重复计算。

CSPNet 的思想在 YOLOv4 和 YOLOv5 目标检测算法中继续被延用。YOLOv4 将 CSPNet 与 YOLOv3 的骨干网络 DarkNet53 结合,增强了卷积神经网络的学习能力,也降低了计算瓶颈和内存成本。YOLOv5 与 YOLOv4 结构的不同点在于,YOLOv4 只是在骨干网络部分运用了 CSPNet 结构,而 YOLOv5 设计了两种 CSPNet 结构,分别为 CSP1_X 和 CSP2_X。如图 3.10 和 3.11 所示,CSP1_X 结构应用于骨干网络,另一种 CSP2_X 结构则应用于 Neck 部分。总结来说,在目标检测任务中,利用 CSPNet 改进骨干网络能够提升检测的性能,增强了 CNN 的学习能力,同时减少计算量,提升推理的速度。

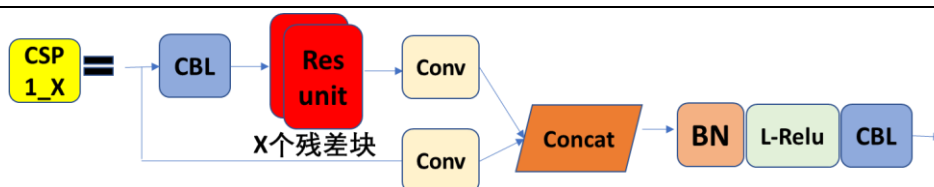


图 3.10 CSP1_X 结构

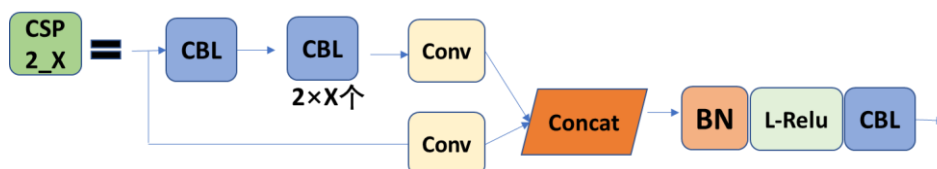


图 3.11 CSP2_X 结构

3.4 损失函数

目标检测任务的损失函数一般由边界框回归损失和分类损失两部分构成，本节将说明实验训练阶段运用的边界框回归损失和分类损失。

边界框回归损失中最常用的计算指标是交并比 (IoU)，计算方法如公式(3.9)。

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (3.9)$$

交并比可以获得预测框和真实框之间的距离，从而反映检测的效果，但是它作为损失函数会有以下的缺点：如果两个框没有相交，根据定义， $IoU = 0$ ，不能充分反映两者的重合程度；同时因为 $IoU = 0$ 时， $loss = 1 - IoU = 1$ ，没有梯度回传，所以无法进行学习和训练。

YOLOv5 算法采用的是 GIoU^[17]作为边界框回归的损失函数，GIoU 方法在克服了 IoU 缺点的同时又充分利用 IoU 的优点。假设 A 为预测框，B 为真实框，令 C 表示包含 A 与 B 的最小凸闭合框。GIoU 的计算公式如(3.10)，损失函数 GIoUloss 的计算如公式(3.11)。

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|} \quad (3.10)$$

$$GIoUloss = 1 - GIoU \quad (3.11)$$

训练阶段的分类损失采用的是二元交叉熵损失 (BCE loss)。因此，如公式(3.12)所示，完整的损失函数由边界框回归损失（第一项）、置信度预测损失（第二三项）和类别预测损失（第四项）三部分构成。

$$Loss(obj) = GloUloss$$

$$\begin{aligned}
& + \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} [C_i \log(C_i) \\
& + (1 - C_i) \log(1 - C_i)] \\
& - \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbf{1}_{ij}^{noobj} [C_i \log(C_i) \\
& + (1 - C_i) \log(1 - C_i)] \\
& + \sum_{i=0}^{S \times S} \sum_{j=0}^B \mathbf{1}_{ij}^{obj} \sum_{c \in classes} [p_i(c) \log(p_i(c)) \\
& + (1 - p_i(c)) \log(1 - p_i(c))]
\end{aligned} \tag{3.12}$$

第4章 新型的特征融合网络及其数据实验结果

4.1 引言

本章，我们阐述了基于 YOLOv5 的改进算法，提出了新型的特征融合网络结构。考虑到目标检测任务的实际工程需求，我们聚焦于在轻量级的 YOLOv5s 和 YOLOv5m 基础之上进行改进。我们发现，多尺度目标的检测能力是衡量目标检测算法性能的重要标准，中小尺度目标的检测也越来越有意义。因此，我们对 YOLOv5s 和 YOLOv5m 模型的结构做了改进，在基本维持推理速度的同时提高了检测精度，尤其对中、小目标的检测效果有较好的提升。

YOLOv5 原本的程序中，Neck 部分和 YOLOv4 一样，采用的都是 PANet 结构，结构的具体形式是在自上而下的 FPN 结构后面加上了自下而上的 PANet 结构。如前文 2.3.4 节所言，考虑到双向特征金字塔网络（BiFPN）的结构优势，我们提出将 BiFPN 的思想应用到多尺度特征融合部分。与此同时，为了提高中小尺度目标的检测效果，我们改进特征融合的网络结构，充分利用高分辨率的低层级特征，扩大模型的感受野。

为了验证本文提出的改进算法的性能，我们在具有挑战性的 MS COCO 数据集上进行数据实验。实验结果表明：以 YOLOv5s 和 YOLOv5m 为基准，我们的模型提高了检测的精确度，尤其对于小、中型目标的检测有较好的提升。测速实验显示我们的模型 YOLOv5s_ours 在提高精度的同时，也基本维持了推理速度。除此之外，我们也将本文提出的改进算法与其他经典目标检测算法进行了精确度、速度方面的对比。对比结果显示，本文提出的改进算法在检测性能方面有了一定的提升。

4.2 新型的双向特征融合

如前文所述，YOLOv5 目标检测框架可以分为如下几个部分：输入端、骨干网络、Neck 部分和输出端，本节将聚焦于改进 Neck 部分的多尺度特征融合结构。由前文所提到的 BiFPN 的具体算法，我们可以知道 BiFPN 的特征融合策略是将双向特征金字塔提取的三部分特征加权后按像素求和；而 YOLOv5 使用的特征融合策略是按照 PANet 的想法。PANet 结构对 FPN 结构进行了补充，在 FPN 后面添加了自下而上的金字塔结构。值得注意的是，YOLOv5 算法用到的 PANet 结构与传统的 PANet 结构在特征的融合方式上有一定的区别。YOLOv5 算法对于提取到的特征没有采取按像素求和的融合策略，而是采取按通道维度进行拼接的方式。

按像素求和与按通道拼接这两种方式中，我们的改进算法选择的融合方式是按通道维度拼接。注意到双向特征网络的成功，我们产生了是否可以将 BiFPN 多尺度特征融合与 YOLOv5 目标检测框架合二为一的想法，即考虑在模型特征融合的问题上采用按通道维度拼接方式的同时利用双向网络。在这种改用按通道维度拼接的策略下，我们需要考虑模型容量的变化。考虑到本文聚焦于轻量级模型 YOLOv5s 和 YOLOv5m，它们的通道数比较少，因此不会带来较多的计算量，换言之不会在速度上造成较大的影响。这部分的实验数据也验证了我们的想法，具体数据将在 4.5 节以表格的方式呈现。

下面，我们将说明改进的新型特征融合结构的具体形式。为了方便表示，我们将特征融合的方式用式子(4.1)体现：

$$feature = [f_1; f_2; f_3] \quad (4.1)$$

其中，符号 $[\cdot]$ 表示按通道维度拼接， f_1 、 f_2 、 f_3 代表双向特征网络中的三个特征， $feature$ 表示融合之后用于后续目标检测的特征。如图 4.1 和 4.2， C_i 表示前馈网络提取到的多尺度特征。

如图 4.2，以 P_4 特征的生成过程为例。 P_4 特征是由 P_3 下采样后得到的特征、 C_4 和 F_4 三部分特征按通道维度拼接后得到的。生成的过程如公式 (4.2) 所示，其中我们将 CSP2_1 算子记作函数 F （CSP2_1 算子结构如图 3.11）， $Downsample$ 指的是通过卷积操作实现两倍的下采样。

$$P_4 = F([Downsample(P_3); F_4; C_4]) \quad (4.2)$$

同理，可得 P_3 特征的生成过程如公式 (4.3) 所示。

$$P_3 = F([Downsample(P_2); F_3; C_3]) \quad (4.3)$$

在本文的实验中，我们是在用于生成中小型目标检测的特征 $\{P_3, P_4\}$ 的过程中，采用了双向特征融合方法。而在用于大物体检测的特征 P_5 上没有使用该方法，因为针对大目标检测问题，双向特征网络带来的效果不是非常显著。如图 4.1 和 4.2 所示，用于 YOLOv5 检测头的三个尺度的特征分别是最终得到的 $\{P_3, P_4, P_5\}$ 。

我们将原始 YOLOv5 算法的特征融合网络和本文改进算法的特征融合网络都绘制出来，分别如图 4.1 和 4.2 所示。可以发现，本文提出的受 BiFPN 启发的特征融合结构与原始 YOLOv5 算法 PANet 形式的特征融合网络之间的区别是：本文的特征融合网络利用了双向特征融合的思想。除此之外，为了达到提高小目标检测精度的效果，我们利用了比 C_3 更低层级的特征 C_2 ，如何充分利用低层级特征的过程将在下节具体说明。

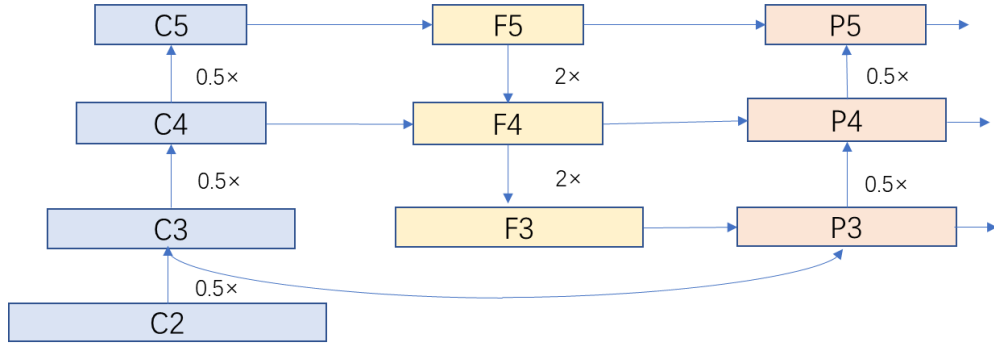


图 4.1 原始 YOLOv5 算法的特征融合网络

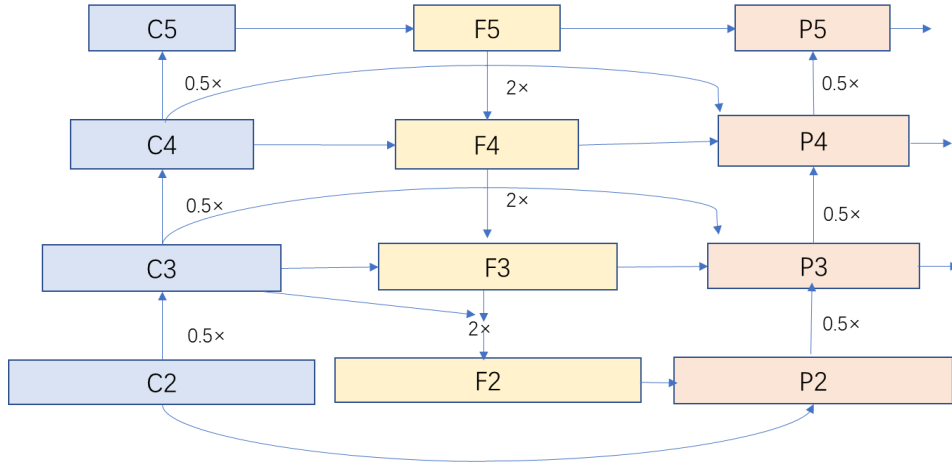


图 4.2 本文改进算法的特征融合网络

4.3 P_2 层级特征的充分利用

如图 4.1，原始 YOLOv5 目标检测框架的检测头之前，用于小目标检测的特征 P_3 的获取方式如公式(4.4)所示。其中，符号 $[\cdot]$ 表示按通道维度拼接， F 表示 CSP2_1 算子，*Upsample* 指的是通过双线性插值操作来实现两倍的上采样。

$$\begin{aligned} P_3 &= F([C_3; F_3]) \\ F_3 &= \text{Upsample}(F_4) \end{aligned} \quad (4.4)$$

正如前文所言，考虑到与中、大型尺度的目标相比，小目标检测的效果相对不佳，因此我们尝试采用充分利用低层级特征的方式来提高小目标检测的精度。

我们注意到用于小目标检测的特征 P_3 并没有和前一层级的特征 C_2 建立联系，而是利用 C_3 特征和 F_3 特征， F_3 特征是通过 F_4 特征上采样得到的。然而由于网络下采样的限制， F_4 的信息缺失较为严重，这可能会使网络对一部分小目标失去监督的能力。因此，我们的想法是充分利用 P_2 特征，将高分辨率的 P_2 特征的信息引入到特征融合中。考虑到双向网络的广泛应用，我们做了一系列权衡，最终得到了如下的算法，我们通过公式(4.5)~(4.8)来具体说明：

$$F_2 = \text{Upsample}([F_3; C_3]) \quad (4.5)$$

公式中符号的定义与之前的定义方式相同, *Upsample*指的是通过双线性插值操作来实现两倍的上采样。低层级的 P_2 特征是通过将 C_2 特征和 F_2 特征融合得到的, 如公式(4.6)所示。

$$P_2 = F([C_2; F_2]) \quad (4.6)$$

考虑到我们前文提到过的双向特征融合方式, 生成 P_3 特征需要 P_2 下采样后得到的特征、 C_3 和 F_3 三部分。本文用于小目标检测的 P_3 特征的获取方式如公式(4.7)所示。

$$P_3 = F([\text{Downsample}(P_2); F_3; C_3]) \quad (4.7)$$

同理, 可得 P_4 特征的生成过程如公式(4.8)。

$$P_4 = F([\text{Downsample}(P_3); F_4; C_4]) \quad (4.8)$$

值得注意的是, 我们是在获取用于最终检测的 P_3 和 P_4 特征时运用了双向特征网络的概念。出于计算量和推理时间的考虑, 我们并没有在 P_2 特征级别上使用双向特征融合。从我们的预想上来看, 充分利用低层级的 P_2 特征应该会提高检测模型在小目标检测上的效果, 最终 4.5 节的实验数据结果也印证了我们的猜想。

4.4 数据实验

4.4.1 数据集介绍

本文的实验部分选择了 Microsoft COCO 2017^[19]作为实验数据集, 因为使用 MS COCO 数据集能够更加严格地评估模型的检测质量。MS COCO 数据集由微软团队提出和构建, 是一个大型的用于目标检测、语义分割、关键点检测等经典计算机视觉任务的数据集。MS COCO 数据集主要由图片和 json 标签文件组成, 其中包含 33 万张图像、150 万个目标实例、80 个目标类别、91 个物体类别以及 25 万个关键点人物。2017 版本的 MS COCO 数据集中, 训练集图片数量为 118287 张, 验证集图片数量为 5000 张。

目标检测任务的另一个常用数据集是 PASCAL VOC^{[31][32]}数据集, 它总共包含 20 个类别, 平均每张图片包含 1.4 个类别。与 PASCAL VOC 数据集相比, MS COCO 数据集平均每张图片包含 3.5 个类别, 仅有不到 20% 的图片只包含一个类别且小目标的占比也更多。因此, 我们的实验选择 MS COCO 数据集的原因在于, 该数据集中的图片包含了自然图片以及生活中常见的目标图片, 其背景比较复杂, 目标的种类和数量相对较多, 因而在 MS COCO 数据集上进行检测任务是更难的。与此同时, 对于本文想要优化小目标检测效果的任务来说, 由于数据集超过一半的

图像中包含大小不超过 32×32 像素的小目标，检测难度较大，是目前最具有挑战性的目标检测数据集。综上，实验部分使用 MS COCO 数据集评估我们的改进模型的质量是更加具有说服力的。

4.4.2 评估标准

常用评估指标的表示		预测情况	
		Positive (预测结果为正例)	Negative (预测结果为反例)
真实情况	True (正例)	TP	FN
	False (反例)	FP	TN

表 4.1 目标检测常用评估指标的表示

本节将列举目标检测任务中常用的几项评估指标的计算方法，分别为召回率、精确率（或称查准率）和准确率等。

(1) 召回率 (Recall)：表示正确检测的物体 M 的个数占测试集中物体 M 的总数的百分比。 $Recall = \frac{TP}{TP+FN}$ 。

(2) 精确率 (Precision)：表示正确检测的物体 M 的个数占总检测物体个数 N 的百分比。 $Precision = \frac{TP}{TP+FP}$ 。

(3) 准确率 (Accuracy)：表示正确分类的样本数占全部样本数的百分比。正确率越高，分类的效果越好。 $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ 。

此外，以召回率 (Recall) 为横坐标，准确率 (Precision) 为纵坐标，在一定阈值的基础之上形成的曲线称为 P-R 曲线。P-R 曲线下方围成的面积即为平均精确度 (AP)，它也是 MS COCO 数据集的主要评估指标。如果对多个类别求平均 AP 值，则得到 mAP 值。MS COCO 数据集中计算 mAP 的方式是：计算 80 个类别在 10 个 IoU 值下的平均精确度 (AP) 值。10 个 IoU 的选择是，IoU 从 0.5 到 0.95 每变化 0.05 就测试一次精确度，再将这 10 次测量结果的数值求平均得到最终的平均精确度。

与此同时，MS COCO 数据集还将物体分为大、中、小三种尺度分别进行测量 mAP 值，其中小目标占比 41%（大小 $< 32 \times 32$ 像素），中等目标占比 34%（ 32×32 像素 $<$ 大小 $< 96 \times 96$ 像素），大目标占比 24%（大小 $> 96 \times 96$ 像素）。

4.4.3 实验设置

实验中,我们选择的优化方法是带动量的随机梯度下降法,初始学习率设置为 $1e-2$,学习率的下降方式为余弦下降,最终下降到 $1e-4$,权重衰减参数设置为 $5e-4$ 。在训练的初始阶段,我们采用了3个 epoch 的热身训练,在使用 Focal loss 训练时 gamma 参数设置为 1.5。为了方便进行对比实验,其他设置我们沿用了 YOLOv5 的基础训练策略。我们的训练显卡为 Tesla V100, YOLOv5s 的训练时间约4天, YOLOv5m 约为一周。

4.4.4 数据预处理

实验的数据预处理方面采用了随机尺度变换、随机移位、随机左右翻转、Mosaic 和色彩增强算法(HSV 空间)等数据增强方式,其中 Mosaic 数据增强方法已经在 3.2 节具体阐述。

除了上述几种数据预处理方法之外,我们还在推理阶段采用了自适应图像缩放的方法。传统的目标检测算法中,数据集中的图像的长宽各不相同,而输入到检测网络的图像则需要尺度相同,比如 YOLO 算法输入到检测网络的图像大小一般为 608×608 或者 416×416 ,因此需要将原始图像进行一定的缩放并统一尺度后再输入到检测网络中。然而,在项目的实际应用中,由于数据集中图像的长宽比不同,所以缩放后填充的黑边大小各不相同。如果填充的黑边面积过大,那么会造成信息冗余,影响推理的速度。因此,我们对原始图像自适应地添加尽可能少的黑边,减少计算量,进而提升算法推理的速度。自适应缩放图片的具体过程如下:

(1) 计算缩放系数: 设原始图像的大小为 $W \times H$, 缩放后网络的输入图像的大小为 $S \times S$, 计算宽和高的缩放系数分别为 S/W 和 S/H 。选择两者中较小的 $\min\{S/W, S/H\}$ 作为缩放系数。

(2) 计算缩放后的图像大小: 用步骤(1)得到的缩放系数与原始图片相乘, 得到缩放后图像的长宽分别为 $W \times \min\{S/W, S/H\}$ 和 $H \times \min\{S/W, S/H\}$

(3) 计算填充的黑边量: 首先计算原本需要填充的黑边量 X , X 的计算方法如公式(4.9)。

$$X = |W \times \min\{S/W, S/H\} - H \times \min\{S/W, S/H\}| \quad (4.9)$$

再将 X 与 32 取余后得到的结果再除以 2, 得到自适应缩放方法两边各填充的黑边量。其中, 数值 32 的含义是: YOLOv5 算法的网络在前向传播的过程中共经历了 5 次下采样, 即 $32 = 2^5$ 。

总结来说,在实验的推理阶段采用自适应图像缩放的操作,能够有效地提高检测算法的推理速度。

4.5 实验结果

YOLOv5 系列网络共有 YOLOv5s、YOLOv5m、YOLOv5l 和 YOLOv5x 四种结构。YOLOv5s 是其中特征图宽度最小、参数量最少的网络,其余三种网络都是在 YOLOv5s 的基础之上不断加宽的结构。本文的实验部分,数据集选择了具有挑战性的 MS COCO 数据集。本节的主要内容可以分为以下几个部分:

(1) 参考原始 YOLOv5 算法的实验结果,将 YOLOv5 与当前性能较好的 EfficientDet 算法进行比较,如图 4.3,可以发现:YOLOv5 算法是更具有优势的。因此,我们的算法在 YOLOv5 的基础上进行改进,具有较好的实际应用价值。

(2) 我们参考 EfficientDet 的实验数据,列出数据表格,将改进后的算法 YOLOv5_ours 与 YOLOv5 系列算法以及其他与 EfficientDet 同类的目标检测算法进行比较。观察实验数据,我们发现改进算法的精确度(AP 值)有了一定的提升。

(3) 我们聚焦于轻量级的 YOLOv5s 和 YOLOv5m 在小、中尺度目标上的改进结果,与运算速度相近的同类方法进行比较。可以发现,我们的改进模型 YOLOv5s_ours 和 YOLOv5m_ours 对于小目标的检测有较好的提升。

(4) 我们改动了损失函数并进行对比实验,将原本的二分类交叉熵损失更改为用于平衡正负样本的 Focal loss。实验数据显示,使用 Focal loss 的模型精度有一定的下降。因此,对于 YOLOv5 系列模型来说,二分类交叉熵损失是更加合适的损失函数。

(5) 我们比较改进后的模型 YOLOv5s_ours 与原始 YOLOv5s 模型的召回率和推理速度。我们发现,与 YOLOv5s 算法相比,改进算法 YOLOv5s_ours 的召回率有一定的提升。此外,YOLOv5_ours 检测精度提升的同时,基本维持了推理的速度。下面,我们将具体阐述每一部分的具体内容。

(一) YOLOv5 系列算法对比 EfficientDet 系列算法

因为 EfficientDet 是当前推理速度较快且检测精度较高的算法,所以 YOLOv5 原本的程序代码将 YOLOv5 系列与 EfficientDet 算法进行了性能的比较。如图 4.3,横轴代表模型在 GPU 上的推理速度,横坐标的数值指的是每计算一张图像所用的时间(单位是毫秒);纵轴代表模型的精确度,纵坐标的数值表示模型在 MS COCO

验证集上测得的精确度 (AP 值)。可以看到,代表模型的点坐标越靠近左上角点,则表示模型的性能越好。从图 4.3 中我们可以发现, YOLOv5s 与 EfficientDet-D0 相比, YOLOv5s 的推理速度更快且在验证集上的精度更高; 同样, YOLOv5m、YOLOv5l、YOLOv5x 与 EfficientDet-D1、EfficientDet-D2 相比, YOLO 系列算法是速度更快、精度更高的。综上, 与 EfficientDet 的各个模型相比, YOLOv5 系列的各个模型的性能相对较好。因此, 我们的算法 YOLOv5_ours 在 YOLOv5 系列算法的基础上进行改进, 具有较好的实用性。

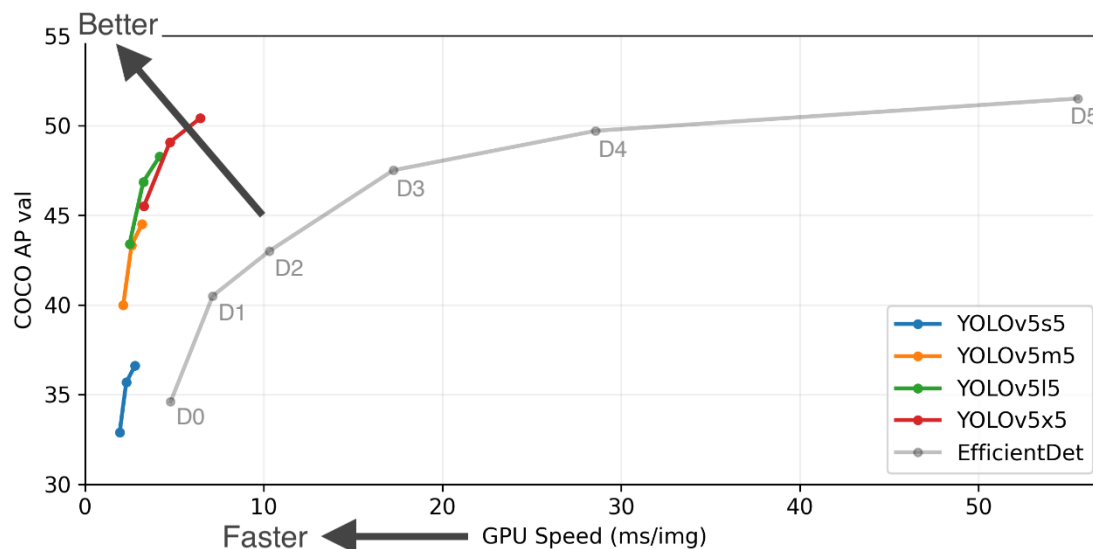


图 4.3 YOLOv5 系列算法对比 EfficientDet 系列算法

(二) YOLOv5_ours、YOLOv5 系列以及其他同类算法的对比

我们列出数据表格, 将改进模型 YOLOv5_ours、YOLOv5 系列以及其他与 EfficientDet 同类的目标检测算法进行比较。由 YOLOv5 系列与 EfficientDet 系列的对比图(图 4.3), 我们可以作出如下分组分别进行比较: YOLOv5s 和 EfficientDet-D0 一组、YOLOv5m 和 EfficientDet-D1、D2 一组、YOLOv5l 和 EfficientDet-D3 一组以及 YOLOv5x 和 EfficientDet-D4 一组。参考 EfficientDet^[14]原论文中的表 2, 我们再将精确度 (AP) 相近的算法分为一组, 如表 4.2 所示。我们把参数量最少、精确度最高的数据进行加粗。通过比较各个模型在 MS COCO 数据集的测试集和验证集上的平均精确度, 我们发现, YOLOv5 算法是每一组中精确度最高的, 且我们的改进算法 YOLOv5s_ours 和 YOLOv5m_ours 是分别比 YOLOv5s 和 YOLOv5m 算法精确度还要高的。从参数量的比较上, YOLOv5 算法的参数量略高于 EfficientDet 算法但远少于其他同类算法。虽然 EfficientDet 算法参数量较少, 但该算法在计算过程中内存的占用量较高, 推理时所用的计算时间比 YOLOv5 算法消耗更多。综上, 与其他同类检测算法相比, YOLOv5 改进算法的精确度有了一定的提升。

方法（输入图像大小）	参数量 (M)	AP (%) 测试集	AP (%) 验证集
YOLOv3 (608) ^[11]	-	33.0	-
EfficientDet-D0 (512) ^[14]	3.9	34.6	34.3
YOLOv5s (640)	7.3	36.7	36.7
YOLOv5s_ours (640)	-	38.1	38.1
RetinaNet-R50 (640) ^[25]	34	39.2	39.2
RetinaNet-R101 (640) ^[25]	53	39.9	39.8
EfficientDet-D1 (640) ^[14]	6.6	40.5	40.2
Detectron2 Mask R-CNN R101-FPN ^[41]	63	-	42.9
Detectron2 Mask R-CNN X101-FPN ^[41]	107	-	44.3
EfficientDet-D2 (768) ^[14]	8.1	43.9	43.5
YOLOv5m (640)	21.4	44.5	44.5
YOLOv5m_ours (640)	-	44.7	44.7
ResNet-50 + NAS-FPN (1024) ^[35]	60	44.2	-
ResNet-50 + NAS-FPN (1280) ^[35]	60	44.8	-
EfficientDet-D3 (896) ^[14]	12	47.2	46.8
YOLOv5l (640)	47.0	48.2	48.2
AmoebaNet+ NAS-FPN +AA(1280) ^[51]	209	-	48.6
EfficientDet-D4 (1024) ^[14]	52	49.7	49.3
YOLOv5x (640)	87.7	50.4	50.4

指标：AP@[IOU=0.50:0.95]

表 4.2 YOLOv5_ours、YOLOv5 系列以及其他同类算法的对比

（三）YOLOv5s_ours、YOLOv5m_ours 在小、中尺度目标上的改进

如 4.2~4.3 节所言，我们聚焦于在轻量级的 YOLOv5s 和 YOLOv5m 基础之上进行改进，将新型的双向特征融合结构和充分利用低层级特征应用到原始的检测网络中，以达到提高小、中尺度目标检测效果的目的。此外，针对小、中尺度目标的检测精度，我们还将改进算法分别与其他同类的方法进行了比较。

参考 YOLOv4^[15]的实验数据，我们将与 YOLOv4 同类且在 Tesla V100 上测试推理时间相近的算法划分为一组，比较同组算法的平均精确度，实验数据如表 4.3 所示。其中，AP_S、AP_M、AP_L 分别指的是小、中、大尺度目标上检测的平均精确度。我们可以看到，与其他同组算法相比，在推理时间相近甚至更少的情况下，基于 YOLOv5 改进算法的平均精确度以及在小、中尺度目标上的平均精确度都相对较高。

此外,我们还观察到:在YOLOv5s基础上的改进算法YOLOv5s_ours在MS COCO测试集上的平均精确度由原来的36.7%提升至38.1%。在小尺度目标上,YOLOv5s的平均精确度为21.0%,而我们改进算法的平均精确度为22.7%,提升了1.7%;在中尺度目标上,YOLOv5s的平均精确度为42.1%,而我们改进算法的平均精确度为43.0%,提升了0.9%。

在YOLOv5m基础上的改进算法YOLOv5m_ours在MS COCO测试集上的平均精确度由原来的44.5%提升44.7%。在小尺度目标上,YOLOv5m的平均精确度为27.4%,而我们改进算法的平均精确度为28.4%,提升了1.0%。可以看到,相较于YOLOv5m来说,我们的改进算法YOLOv5m_ours可以提升平均准确率,在小目标检测上的提升较为明显。然而,对于大、中尺度的目标而言,改进算法的检测效果提升相对较小。我们分析原因,可以发现:伴随着模型容量的增大,我们的改动带来的增益在大尺度目标上相对较弱。因此,与原始的YOLOv5算法相比,本文提出的改进算法可以提升平均精确度,且在小目标检测上的提升效果相对明显。

方法(输入大小)	骨干网络	AP	AP_S	AP_M	AP_L
SSD(512) ^[8]	VGG-16	28.8	10.9	31.8	43.5
YOLOv3(320) ^[11]	Darknet-53	28.2	11.9	30.6	43.4
YOLOv3(416) ^[11]	Darknet-53	31.0	15.2	33.2	42.8
RFBNet(512) ^[52]	HarDNet68	33.9	14.7	36.6	50.5
YOLOv3+ASFF*(320) ^[53]	Darknet-53	38.1	16.1	41.6	53.6
EfficientDet-D0 ^[14]	Efficient-B0	33.8	12.0	38.3	51.2
YOLOv5s(640)	CSPDarknet-53	36.7	21.0	42.1	47.4
YOLOv5s_ours(640)	CSPDarknet-53	38.1	22.7	43.0	47.7
YOLOv3(608) ^[11]	Darknet-53	33.0	18.3	35.4	41.9
YOLOv3+ASFF*(416) ^[53]	Darknet-53	38.1	16.1	41.6	53.6
YOLOv3+ASFF*(608) ^[53]	Darknet-53	42.4	25.5	45.7	52.3
YOLOv3+SPP(608) ^[5]	Darknet-53	36.2	20.6	37.4	46.1
ATSS(800) ^[54]	ResNet-101	43.6	26.1	47.0	53.6
RDSNet(600) ^[55]	ResNet-101	36.0	17.4	39.6	49.7
CenterMask(800) ^[56]	ResNet-101-FPN	44.0	25.8	46.8	54.9
EfficientDet-D1(640) ^[14]	Efficient-B1	39.6	17.9	44.3	56.0
EfficientDet-D2(640) ^[14]	Efficient-B2	43.0	22.5	47.0	58.4
YOLOv4(608) ^[15]	CSPDarknet-53	43.5	26.7	46.7	53.3
YOLOv5m(640)	CSPDarknet-53	44.5	27.4	50.0	56.3
YOLOv5m_ours(640)	CSPDarknet-53	44.7	28.4	49.9	58.4

指标: AP@[IOU=0.50:0.95]

表4.3 YOLOv5_ours、YOLOv5系列以及其他同类算法不同尺度目标的检测对比

(四) 二分类交叉熵损失 (BCE loss) 与 Focal loss 的对比

以改进模型 YOLOv5s_ours 为基准,我们将分类的损失函数由原来的二分类交叉熵损失改为 Focal loss^[25],进行了对比实验。Focal loss 是何凯明团队提出的用于目标检测的损失函数,主要是为了解决一阶段目标检测算法中正负样本比例严重失衡的问题。Focal loss 给予简单样本较小的损失权重,主要挖掘困难样本。因此,Focal loss 在目标检测领域应用十分广泛。Focal loss 用公式表示如(4.10)所示。实验结果如表 4.4 所示,我们注意到:模型的损失函数使用 Focal loss 与使用二分类交叉熵损失(BCE)相比,模型的精确度有所下降。因此,损失函数如果使用 Focal loss,则不能取得更好的效果。所以说,Focal loss 可以加速模型的收敛但不一定可以提升检测的精度。

$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$$

$$p_t = \begin{cases} p & y = 1 \\ 1 - p & \text{其他} \end{cases} \quad (4.10)$$

方法	损失函数	AP	AP_S	AP_M	AP_L
YOLOv5s	BCE loss	36.7	21.0	42.1	47.4
YOLOv5s_ours	BCE loss	38.1	22.7	43.0	47.7
YOLOv5s_ours	Focal loss	38.0	22.7	42.9	47.4

表 4.4 以 YOLOv5s_ours 为基准,比较二元交叉熵损失和 focal loss 的结果

(五) YOLOv5s_ours 与 YOLOv5s 召回率、推理时间的对比

除了比较模型的精确度 (AP) 之外,我们的实验还比较了目标检测任务的另一项重要指标:召回率。与计算精确度类似,我们计算召回率时同样计算的是 IoU 从 0.5 增加到 0.95 且步长为 0.05 的 10 次召回率的平均值。此外,我们将置信度设置为 0.001。实验数据如表 4.5 所示,可以看到,我们新改进的模型召回率相较于原模型也有了一定的提升,并且对于不同尺度目标的召回率均有一定的提升。

方法	AR	AR_S	AR_M	AR_L
YOLOv5s	57.4	38.9	63.6	71.5
YOLOv5s_ours	58.4	39.5	64.2	72.3

指标: AR@[IOU=0.50:0.95]

表 4.5 YOLOv5s_ours 与 YOLOv5s 召回率的对比

此外,我们还通过绘制折线统计图的方式,比较 YOLOv5s_ours、YOLOv5 系列以及其他同类算法的检测性能。如图 4.4 所示,我们可以发现,与其他同类

算法相比，YOLOv5 系列算法是在精度和速度两方面权衡之下性能较好的，而我们的改进算法 YOLOv5s_ours 与 YOLOv5s 相比，在速度（FPS：每秒传输帧数）相近的情况下，精度（mAP）也有了一定的提升。

最后，我们以 YOLOv5s 为基准进行了测速实验。实验阶段，我们选用的是模型的 FP16 版本，测试的设备是 Tesla V100。我们在测速时，去除掉非极大值抑制（NMS）的时间消耗，只记录推理时间。如表 4.6 所示，可以看到，我们的改进模型 YOLOv5s_ours 在提升小目标检测精度的同时，依然维持了推理的速度。此外，我们发现随着模型通道数的增加，我们的改进算法如果以 YOLOv5l 或 YOLOv5x 为基准，那么会出现以下情况：虽然检测的精度会有一定的提升，但是推理时间会增加。因此，我们在轻量级的 YOLOv5s 基础之上的改进算法是更具有实际意义的，该算法能够做到基本维持推理速度的同时，提升目标检测的精确度，并且在小、中尺度目标上的提升效果较好。

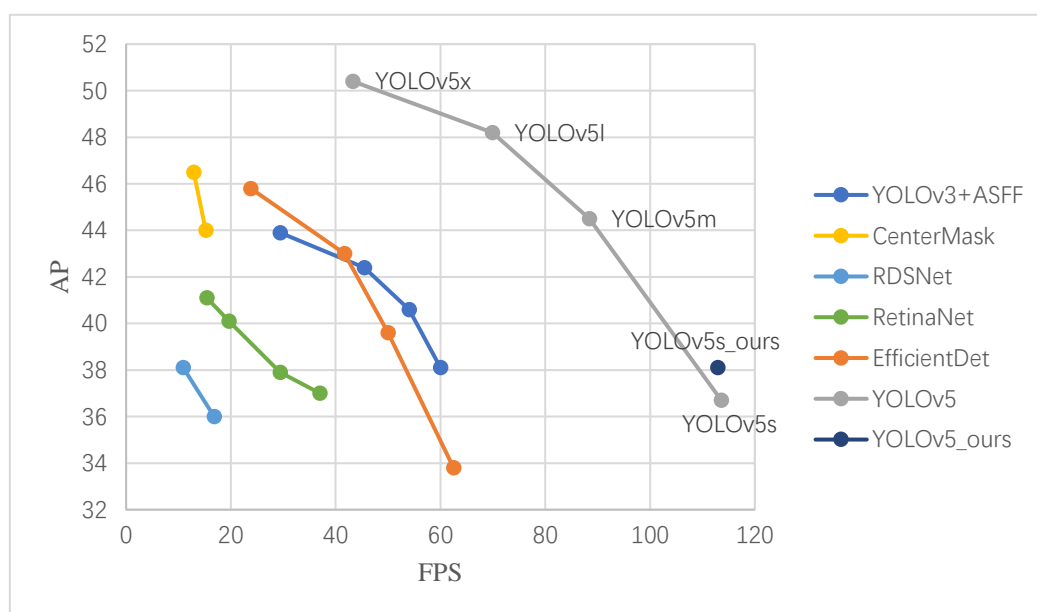


图 4.4 YOLOv5s_ours、YOLOv5 系列以及同类算法的检测性能对比

方法	推理时间	AP	AP_S	AP_M	AP_L
YOLOv5s	1.5ms	36.7	21.0	42.1	47.4
YOLOv5s_ours	1.6ms	38.1	22.7	43.0	47.7

表 4.6 YOLOv5s_ours 与 YOLOv5s 推理速度的对比

第5章 结论和后续研究方向

本文聚焦于目标检测任务，介绍了深度学习的研究背景和目标检测任务的发展现状。基于深度学习的目标检测算法目前可以分为一阶段算法和两阶段算法两大类。首先，我们介绍了基于区域提议的典型两阶段算法，如 RCNN、SPPNet、Fast RCNN、Faster RCNN 算法等；其次，我们介绍了基于边界框回归的一阶段算法：SSD 和 YOLO 系列算法。关于特征融合的方式，本文主要介绍了特征金字塔网络（FPN）、路径聚合网络（PANet）和双向特征融合网络（BiFPN）。此外，因为 YOLOv5 算法是当前效果较好的检测算法，所以我们从数据增强、网络结构、损失函数等方面对 YOLOv5 算法进行了具体的阐述。

受双向特征融合（BiFPN）和 YOLOv5 算法的启发，本文提出一种改进的目标检测算法。首先，考虑到 BiFPN 结构的优势，我们将双向特征融合结构应用于网络的 Neck 部分，并将原来按像素求和的融合策略更改为按通道维度拼接的方式；其次，为了提高网络在小尺度目标上的检测能力，我们对高分辨率的低层级特征进行充分地利用。实验部分，我们选择了具有挑战性的 MS COCO 数据集，并与多个其他同类检测算法进行比较。实验数据结果表明，以轻量级的 YOLOv5s 为基准，我们的改进算法在基本维持推理速度的情况下，能够做到提升检测的精确度，尤其对于小、中型目标的检测精确度提升效果较好。

本文的后续研究方向可以分为以下两个方面：一方面，虽然现有的小目标检测算法已经取得了一些成果，但检测精度依然不够高，小目标检测和旋转目标检测仍然是目标检测领域的困难问题，这部分课题将会是后续研究的重要方向；另一方面，伴随着实例工程的应用，模型是否能够做到更好地适配硬件将变得尤为关键。我们也需要进一步考虑，如何做到使模型具有良好检测性能的同时，降低算法复杂度，提高检测的效率。因此，得到更适合实际应用的新型网络是我们的未来目标。

参考文献

注：极个别文献未标注页码的原因是，该文献尚未在杂志上正式发表而又在线可以得到，部分已附上了 arXiv 网址。

- [1] VIOLA P, JONES M J. Robust real-time face detection[J]. International Journal of Computer Vision, 2004, 57(2): 137-154.
- [2] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection[C]//Proceedings of the IEEE Computer Vision and Pattern Recognition, 2005: 886-893.
- [3] FELZENSZWALB P, MCALLESTER D, RAMANAN D. A discriminatively trained, multiscale, deformable part model[C]//2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008: 1-8.
- [4] 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.
- [5] 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.
- [6] GIRSHICK R. Fast R-CNN[C]//Proceedings of the IEEE International Conference on Computer Vision, 2015: 1440-1448.
- [7] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]//Advances in Neural Information Processing Systems, 2015: 91-99.
- [8] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector[C]//European Conference on Computer Vision. Cham: Springer, 2016: 21-37.

-
- [9] 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.
- [10] REDMON J, FARHADI A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 7263-7271.
- [11] REDMON J, FARHADI A. Yolov3: an incremental improvement[J]. arXiv:1804.02767, 2018.
- [12] LIN T Y, DOLLAR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]//Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition, 2017: 936-944.
- [13] 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: 8759-8768.
- [14] TAN M, PANG R, LE Q V. EfficientDet: scalable and efficient object detection[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020: 10778-10787.
- [15] BOCHKOVSKIY A, WANG C Y. Yolov4: Optimal speed and accuracy of object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. arXiv:2004.10934, 2020.
- [16] WANG C Y, LIAO H Y M, YE H I H, et al. CSPNet: a new backbone that can enhance learning capability of CNN[J]. arXiv:1911.11929, 2019.
- [17] RAHMAN M A, WANG Y. Optimizing intersection-over union in deep neural networks for image segmentation[C]//Proceedings of the International Symposium on Visual Computing, 2016: 234-244.
- [18] REZATOFIGHI H, TSOI N, GWAK J, et al. Generalized intersection over union: A metric and a loss for bounding box regression [C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA: IEEE Press, 2019: 658-666

-
- [19] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft coco: common objects in context[C]//European Conference on Computer Vision. Cham: Springer, 2014: 740-755.
- [20] YUN S, HAN D, OH S J, et al. Cutmix: Regularization strategy to train strong classifiers with localizable features[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 6023-6032.
- [21] BENGIO Y. Deep learning of representations: Looking forward[C]//Proceedings of International Conference on Statistical Language and Speech Processing, 2013: 1-37.
- [22] DIGANTA MISRA. Mish: A self regularized nonmonotonic neural activation function. arXiv preprint arXiv:1908.08681, 2019.
- [23] ZHAO Y N, WU L M, CHEN Q. Small object detection algorithm based on multi-scale fusion SSD[J]. Computer Engineering, 2020, 46(1): 247-254.
- [24] LAW H, DENG J. Corner Net: detecting objects as paired keypoints[C]//Proceedings of the European Conference on Computer Vision, 2018: 734-750.
- [25] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE International Conference on Computer Vision, 2017: 2980-2988.
- [26] DUAN K, BAI S, XIE L, et al. Center Net: keypoint triplets for object detection[C]//Proceedings of the IEEE International Conference on Computer Vision, 2019: 6569-6578.
- [27] KRIZHEVSKY A, SUTSKEVER I, HINTON G. ImageNet Classification with Deep Convolutional Neural Networks[C] //Proceedings of International Conference on Neural Information Processing Systems, 2012: 1097-1105.
- [28] SALAKHUTDINOV R, MNIH A, HINTON G. Restricted Boltzmann machines for collaborative filtering[C]//Proceedings of the 24th international conference on Machine learning. ACM, 2007: 791-798.
- [29] CORTES C, VAPNIK V. Support-vector networks[J].Machine learning, 1995, 20(3): 273-297.

-
- [30] XIAO J, EHINGER K A, HAYS J, et al. Sun database: exploring a large collection of scene categories[J]. International Journal of Computer Vision, 2016, 119(1): 3-22.
- [31] EVERINGHAM M, VAN GOOL L, WILLIAMS C K I, et al. The pascal visual object classes (voc) challenge[J]. International Journal of Computer Vision, 2010, 88(2): 303-338.
- [32] EVERINGHAM M, ESLAMI S M A, VAN GOOL L, et al. The pascal visual object classes challenge: a retrospective[J]. International Journal of Computer Vision, 2015, 111(1): 98-136.
- [33] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.
- [34] KISANTAL M, WOJNA Z, MURAWSKI J, et al. Augmentation for small object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. arXiv:1902.07296, 2019.
- [35] GHIASI G, LIN T Y, LE Q V. NAS-FPN: learning scalable feature pyramid architecture for object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019: 7036-7045.
- [36] XU H, YAO L, ZHANG W, et al. Auto-FPN: automatic network architecture adaptation for object detection beyond classification[C]//Proceedings of the IEEE International Conference on Computer Vision, 2019: 6649-6658.
- [37] LI J, LIANG X, WEI Y, et al. Perceptual generative adversarial networks for small object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 1222-1230.
- [38] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[C]//Advances in Neural Information Processing Systems, 2014: 2672-2680.

-
- [39] BAI Y, ZHANG Y, DING M, et al. SOD-MTGAN: small object detection via multi-task generative adversarial network[C]//Proceedings of the European Conference on Computer Vision (ECCV) 2018: 206-221.
- [40] VAN DE SANDE K E A, UIJLINGS J R R, GEVERS T, et al. Segmentation as selective search for object recognition[C]//2011 International Conference on Computer Vision, 2011: 1879-1886.
- [41] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]// Proceedings of the IEEE International Conference on Computer Vision, 2017: 2961-2969.
- [42] HINTON G, OSINDERO S, TEH Y. A fast learning algorithm for deep belief nets[J]. Neural Computation, 2006, 18(7): 1527-1554.
- [43] MURPHY K P. Machine Learning: A Probabilistic Perspective[M]. Cambridge, MA: MIT Press, 2012: 82-92.
- [44] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv: 1409.1556, 2014.
- [45] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]// Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, USA, 2015: 1-9.
- [46] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.
- [47] FREUND Y, SCHAPIRE R, ABE N. A short introduction to boosting[J]. Journal- Japanese Society For Artificial Intelligence, 1999, 14: 1612.
- [48] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]// Proceedings of the IEEE International Conference on Computer Vision, 2017: 2961-2969.
- [49] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958.

-
- [50] MAAS A L, HANNUN A Y, NG A Y. Rectifier nonlinearities improve neural network acoustic models[C]//Proceedings of the 30th International Conference on Machine Learning. Atlanta: ACM, 2013: 456-462.
- [51] BARRET ZOPH, EKIN D. CUBUK, GOLNAZ GHIASI, LIN TSUNG-YI, JONATHON SHLENS, QUOC V. LE. Learning data augmentation strategies for object detection. arXiv preprint arXiv:1906.11172, 2019.
- [52] CHAO P, KAO C Y, RUAN Y S, HUANG C H, LIN Y L. HarDNet: A low memory traffic network. Proceedings of the IEEE International Conference on Computer Vision (ICCV). arXiv:1909.00948, 2019.
- [53] LIU S T, HUANG D, WANG Y H. Learning spatial fusion for single-shot object detection. arXiv preprint arXiv:1911.09516, 2019.
- [54] ZHANG S, CHI C, YAO Y, et al. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection [C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
- [55] WANG S, GONG Y C, XING J L, HUANG L C, HUANG C, HU W M. RDSNet: A new deep architecture for reciprocal object detection and instance segmentation. arXiv preprint arXiv:1912.05070, 2019.
- [56] LEE Y, PARK J. CenterMask: Real-Time Anchor-Free Instance Segmentation[J]. arXiv preprint arXiv:2004.04446, 2020.
- [57] NAJIBI M, SAMANGOUEI P, CHELLAPPA R, et al. SSH: Single stage headless face detector[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 4875-4884.
- [58] CHEN C, LIU M Y, TUZEL O, et al. R-CNN for small object detection[C]//Asian Conference on Computer Vision. Cham: Springer, 2016: 214-230.
- [59] TAKEKI A, TRINH T T, YOSHIHASHI R, et al. Combining deep features for object detection at various scales: finding small birds in landscape images[J]. IPSJ Transactions on Computer Vision and Applications, 2016, 8(1): 1-7.

- [60] CHEN C, LIU M Y, TUZEL O, et al. R-CNN for small object detection[C]//Asian Conference on Computer Vision. Cham: Springer, 2016: 214-230.
- [61] NAIR V, HINTON G E. Rectified linear units improve restricted Boltzmann machines[C]//Proceedings of the 27th International Conference on International Conference on Machine Learning. Haifa, Israel, 2010: 807–814.
- [62] SHRIVASTAVA A, GUPTA A, GIRSHICK R. Training region- based object detectors with online hard example mining[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 761-769.
- [63] CAI Z, VASCONCELOS N. Cascade R-CNN: delving into high quality object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 6154-6162.
- [64] WAN Li, ZEILER M, ZHANG Sixin, et al. Regularization of neural networks using DropConnect[C]//Proceedings of the 30th International Conference on Machine Learning. Atlanta, USA, 2013: 1058–1066.
- [65] ZITNICK C L, DOLLAR P. Edge boxes: Locating object proposals from edges[C]//Proceedings of the European Conference on Computer Vision, 2014: 391-405.
- [66] 刘洋,战荫伟.基于深度学习的小目标检测算法综述[J].计算机工程与应用, 2021, 57(02): 37-48.
- [67] 梁鸿,王庆玮,张千,李传秀.小目标检测技术研究综述[J].计算机工程与应用, 2021, 57(01): 17-28.
- [68] 陈幻杰,王琦琦,杨国威,韩佳林,尹成娟,陈隽,王以忠.多尺度卷积特征融合的 SSD 目标检测算法[J].计算机科学与探索, 2019, 13(06): 1049-1061.
- [69] 魏玮,蒲玮,刘依.改进 YOLOv3 在航拍目标检测中的应用[J].计算机工程与应用, 2020, 56(07): 17-23.

- [70] 吴天舒,张志佳,刘云鹏,裴文慧,陈红叶.基于改进 SSD 的轻量化小目标检测算法[J].红外与激光工程, 2018, 47(07): 47-53.
- [71] 胡越,罗东阳,花奎,路海明,张学工.关于深度学习的综述与讨论[J].智能系统学报, 2019, 14(01): 1-19.

致谢

时间如白驹过隙，三年研究生时光转眼即将进入尾声。回望过去三年的点点滴滴，内心都心存感激。感谢我的研究生导师，感谢他对本篇论文的辛苦指导。他认真负责、对待学术一丝不苟的专业态度令我敬佩，感谢恩师三年来的严格要求，不仅让我们学到了更多的专业知识，也培养了我们勤恳耐劳的学习态度，这份态度让我们无论今后从事什么行业，都能够从中有所收益。感谢同门的各位师兄姐妹对我学习上的指导和帮助。感谢我的母亲对我的辛苦养育，在外求学无论是我的学习还是我的身体，都让她时时牵挂。最后，祝愿身边每一位尊长未来平安顺遂，每一位朋友前程似锦。