《计算机视觉导论》
课程结课报告

题目： 基于卷积神经网络的经典目标检测算法复现与比较研究

院　　系	计算机科学与技术学院
专业班级	计科 2300班
姓　　名	losyi
学　　号	U202310000
指导教师	李贤芝

计算机科学与技术学院

目　　录

摘　　要

目标检测是计算机视觉中一个重要问题，在行人跟踪、车牌识别、无人驾驶等领域都具有重要的研究价值。随着深度学习技术的飞速发展，基于卷积神经网络（CNN）的目标检测算法已成为主流。本文聚焦于目标检测发展史上的几个典型网络进行了深入的原理剖析与复现研究。分别是 R-CNN、Fast R-CNN、Faster R-CNN 以及 Mask R-CNN。通过在公开数据集 PASCAL VOC2012 上的实验，我们对比分析了这几种方法的性能和优缺点。最后对深度学习的目标检测算法做出总结, 以及未来的发展方向。

关键词：目标检测；深度学习；卷积神经网络；特征提取；计算机视觉

1 绪论

1.1 研究背景和意义

计算机视觉作为人工智能的核心领域，其发展历史可以追溯到 20 世纪 60 年代初期，当时的研究主要聚焦于边缘检测和简单形状识别。随着计算能力的提升和算法的创新，计算机视觉逐步从传统的手工特征提取转向深度学习驱动的自动化处理。根据相关文献，计算机视觉的演进经历了从基于规则的方法到机器学习，再到深度学习的多个阶段。特别是在 2012 年 AlexNet 在 ImageNet 竞赛中的成功，标志着深度学习在图像分类任务上的突破，这为后续的目标检测奠定了基础[5]。

目标检测 (objection detection) 是机器视觉中最常见的问题。是一种基于目标几何和统计特征的图像分割, 它将目标的分割和识别合二为一, 其准确性和实时性是整个系统的一项重要能力, 近年来, 目标检测在人工智能, 人脸识别, 无人驾驶等领域都得到了广泛的应用[13]。这些应用不仅提升了生产效率，还降低了人为错误风险，推动了智能化转型。

近年来，深度学习，特别是卷积神经网络（CNN）的兴起，极大推动了目标检测的发展[1]-[4],[18][19]。传统方法依赖手工特征，如 HOG（Histogram of Oriented Gradients）[7]和 SVM（Support Vector Machine）[20]，但这些方法在复杂场景下鲁棒性不足。2014 年 R-CNN 的提出，使得基于 CNN 的目标检测算法逐渐成为主流，这些网络通过端到端学习，自动提取高层次特征，解决了传统方法的局限性。因此，本研究聚焦 R-CNN 系列网络的复现与比较，具有重要的理论和实践意义：一方面，它有助于理解目标检测算法的演进路径；另一方面，可为实际应用提供优化指导，促进相关领域的技术创新[6]。

1.2 国内外研究现状

目标检测的研究历史悠久，在深度学习方法普及之前，主流技术主要依赖于传统计算机视觉方法。传统的目标检测算法采用类似穷举的滑动窗口方式或图像分割技术来生成大量的候选区域,然后对每一个候选区域提取图像特征(包括 HOG[7]、SIFT[8]、Haar[9]等),并将这些特征传递给一个分类器(如 SVM[[10]、

Adaboost[11]和 Random Forest[12]等)用来判断该候选区域的类别.由于传统方法提取的特征存在局限性,产生候选区域的方法需要大量的计算开销,检测的精度和速度远远达不到实际应用的要求,这使得传统目标检测技术研究陷入了瓶颈。

2014 年，Ross Girshick 等人发表的论文《Rich feature hierarchies for accurate object detection and semantic segmentation》提出的 R-CNN 算法[1]，是目标检测领域的转折点，显著提升了目标检测精度，随后，Fast R-CNN 引入 RoI Pooling 实现端到端训练[2]，Faster R-CNN 添加 RPN（Region Proposal Network）进一步加速[3]。Mask R-CNN 则扩展到实例分割，添加掩码分支[4]。这些方法的历史地位在于，它们奠定了两阶段检测框架的基础，推动了从多阶段到统一网络的演进。目前，国外如 Facebook AI Research 和 Google 等机构主导创新，YOLO 和 SSD 等单阶段方法进一步追求实时性。

1.3 研究内容与任务

本研究聚焦基于经典卷积神经网络的目标检测算法比较，具体复现 R-CNN、Fast R-CNN 和 Faster R-CNN 三种模型，并学习 Mask R-CNN 作为扩展。数据集选用 PASCAL VOC 2012，实验环境基于 windos11 操作系统，使用 PyTorch 框架和 GPU 硬件(NVIDIA GeForce RTX 4070 Laptop GPU)等。研究任务包括：

（1）阅读原始论文，分析核心原理；

（2）使用开源代码复现模型，获取 mAP 和 precision 等指标，并可视化结果；

（3）对比分析几种网络的优缺点，探讨性能提升机制。最终目标是通过实验验证模型演进的逻辑，并为后续改进提供参考。

2　几种网络的原理分析

2.1 R-CNN (Region-based CNN)

R-CNN 的基本思想是将深度学习引入目标检测，通过区域提议生成候选框，然后用 CNN 提取特征，并结合 SVM 分类和边界框回归实现检测.具体而言：首先在输入图片上生成大约 2000 个与目标类别无关的候选区域，然后在每一个候选区域上用 CNN 提取出固定长度的特征向量，最后用线性 SVM 对每一个候选区域进行分类，用仿射变换从每一个候选区域中计算固定大小 CNN，而不管候选区域的大小，图 2-1 给出了 R-CNN 的目标检测流程[1]。

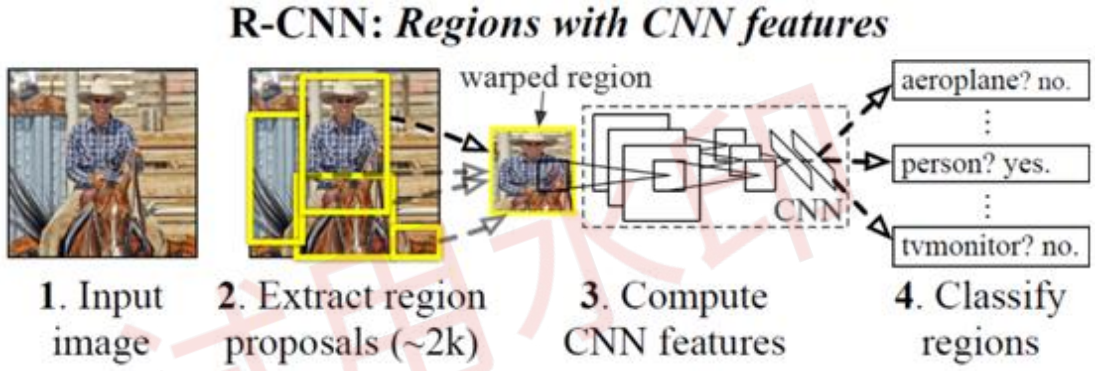

图 2-1 R-CNN 流程

R-CNN 成功地将深度学习的强大能力从图像分类任务迁移到了目标检测，并取得了当时最高的检测精度，将检测效果从 OverFeat 的 24.3%大幅提升至 31.4%（ILSVRC 2013 数据集），并在 VOC2007 数据 集上获得 58.5%的准确率，远超此前的最佳方法[14]。这证明了基于 CNN 的特征学习范式在目标检测任务中的巨大潜力。尽管精度很高，R-CNN 的缺点也十分明显，对约 2000 个区域提议独立进行 CNN 前向传播，存在大量的计算冗余，导致测试一张图像需要数十秒，完全无法满足实时性要求；整个流程被分解为 CNN 微调、SVM 训练和边界框回归器训练三个独立的阶段，训练过程繁琐且耗时；由于需要为每个区域提议提取特征并存储，训练过程需要大量的磁盘空间来缓存这些特征 。

2.2 Fast R-CNN

为了解决 R-CNN 的性能瓶颈，Girshick 在 2015 年提出了 Fast R-CNN，其核心思想是通过共享计算来大幅提升效率。他们受到 SPP-Net 算法[15]的启发,将 SPP 层简化成单尺度的 ROI Pooling 层以统一候选区域特征的大小。它不再对每个区域提议独立进行卷积计算，而是对整张输入图像进行一次 CNN 前向传播，得到一个全局的卷积特征。然后，对于每个区域提议，它将其坐标映射到这个共享的特征图上，并使用一个名为 RoI (Region of Interest) Pooling 的层，从对应的特征区域中提取出一个固定尺寸的特征向量。这个固定尺寸的输出使得网络可以处理任意大小的输入区域，并将其送入后续的全连接层[2]。通过这种方式，卷积计算在所有 2000 个提议之间得到了共享，极大地减少了冗余计算。Fast R-CNN 的结构如图 2-2 所示。

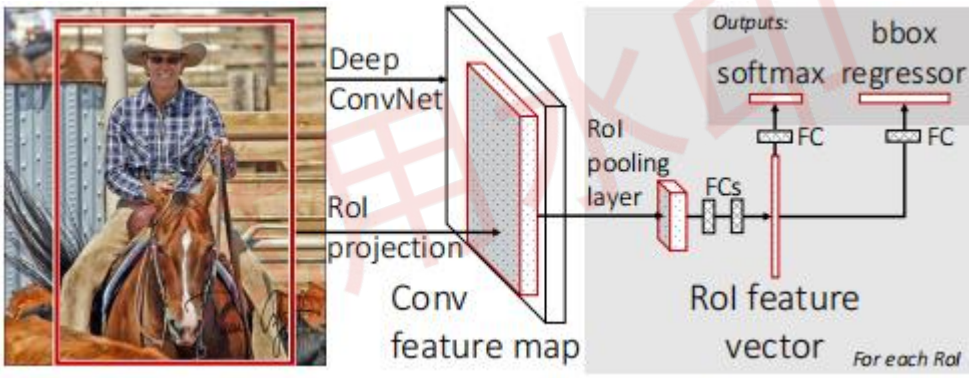

图 2-2 Fast R-CNN 结构示意图

Fast R-CNN 的改进带来了显著的性能飞跃。相比 R-CNN，其训练速度快了约 9 倍，测试速度快了超过 200 倍，在 VOC 2012 上 mAP 达 66%，高于 R-CNN 的 62%。此外，单阶段的训练流程也使得模型部署和调优更为便捷。尽管 Fast R-CNN 极大地优化了检测网络的计算，但它仍然依赖于选择性搜索算法[16]来产生候选区域，耗时较长（每张图约 2 秒），成为了整个系统新的性能瓶颈。

2.3 Faster R-CNN

为了解决 Fast R-CNN 遗留的问题，Ren 等人于 2015 年提出了 Faster R-CNN[3]，其目标是构建一个完全端到端的、统一的目标检测网络。

Faster R-CNN 引入 RPN 实现端到端检测，解决提案瓶颈。RPN 是一个全卷积网络，在共享特征图上滑动窗口预测边界框和物体性分数，使用锚框（anchors）处理多尺度和比例,代替了选择性搜索等传统的候选框生成方法，实现了网络的端到端训练，提高了网络计算速度 Faster R-CNN 网络由卷积层(conv layers)、RPN 网络、ROI(regions of interest)pooling 层、分类和回归层等 4 部分组成，如图 2-3 所示[17]。卷积层作为共享特征提取器，从输入图像中生成高维语义特征图；RPN 在此特征图上通过滑动窗口和预设锚框预测候选区域及其物体性得分，替代了传统耗时的候选框生成方法，并支持端到端训练；RoI Pooling 层将 RPN 输出的不规则候选区域映射并池化为固定尺寸的特征表示，以适配后续网络；最后，分类与回归层基于这些统一尺寸的区域特征，分别预测目标类别和精细化边界框坐标，从而完成最终的检测任务。四者协同工作，在保证检测精度的同时显著提升了计算效率。

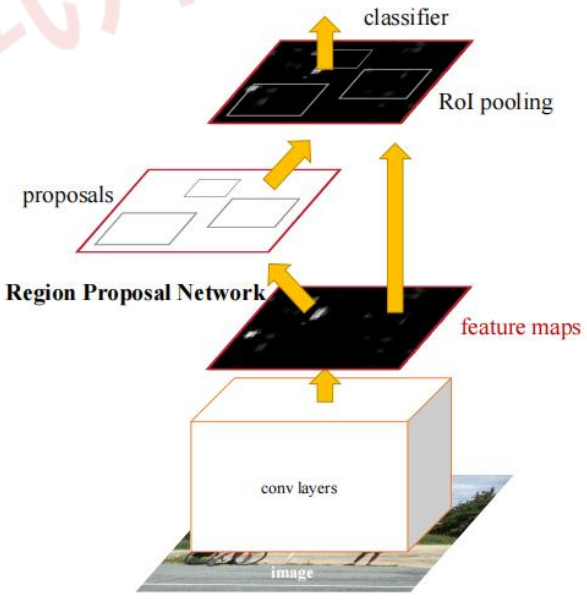

图 2-3 Faster R-CNN 结构图

通过将区域提议整合到 GPU 加速的神经网络中，Faster R-CNN 成功地消除了 CPU 瓶颈，成为第一个真正意义上既准确又高效的深度学习目标检测器。在 GPU 上，它的检测速度可以达到 5 FPS，基本满足了近实时的要求，同时在 PASCAL VOC 2007 数据集上实现了 73.2%的 mAP，在精度和速度上都超越了 Fast R-CNN 。Faster R-CNN 的端到端设计理念，深刻地影响了此后目标检测领域的发展。

2.4 Mask R-CNN

在 Faster R-CNN 成功实现高效、端到端的目标检测之后，He 等人于 2017 年提出了 Mask R-CNN[4]，将该框架的能力从边界框检测扩展到了更具挑战性的实例分割（Instance Segmentation） 任务 。实例分割不仅要检测出物体，还要为每个物体实例生成一个像素级别的精确掩码（mask）。

Mask R-CNN 的基本思想是在 Faster R-CNN 的基础上，增加一个用于预测分割掩码的分支 。具体来说，它保留了 Faster R-CNN 的两阶段结构：第一阶段同样是 RPN，用于生成候选区域。在第二阶段，除了原有的用于分类和边界框回归的分支外，并行地增加了一个新的掩码预测分支（mask head） 。这个掩码分支是一个小型的全卷积网络（FCN），它接收每个 RoI 的特征，并输出一个二值的分割掩码 。其结构如图 2-4 所示。

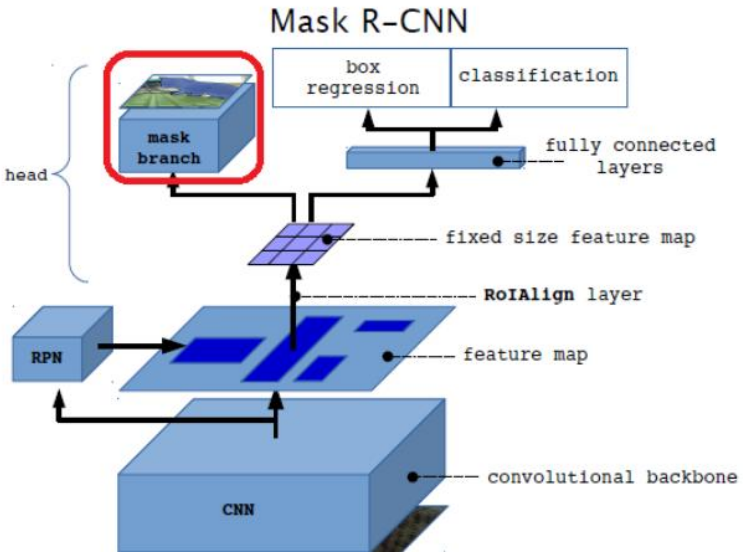

图 2-4 Mask R-CNN 结构图

性能方面，在 COCO test-dev 数据集上，Mask R-CNN ResNet-101-FPN 的掩码 AP 达 35.7（AP50 58.0，AP75 37.8，小/中/大尺度分别为 15.5/38.1/52.4），边界框 AP 达 38.2；ResNeXt-101-FPN 进一步提升到掩码 AP 37.1 和边界框 AP 39.8。该方法在实例分割上优于先前模型，如 MNC（24.6 AP）和 FCIS+++（33.6 AP）。它还泛化到人体姿态估计，将关键点视为 one-hot 二值掩码，APkp 达 63.1，优于 CMU-Pose+++（61.8 APkp）。在 Cityscapes 数据集上，微调+ COCO 预训练的 AP 达 32.0。

3　实验设计与复现过程

3.1 实验环境与数据集

为保证实验的可重复性与高效性，本实验训练在 AutoDL 算力云上进行，如图 3-1 所示。

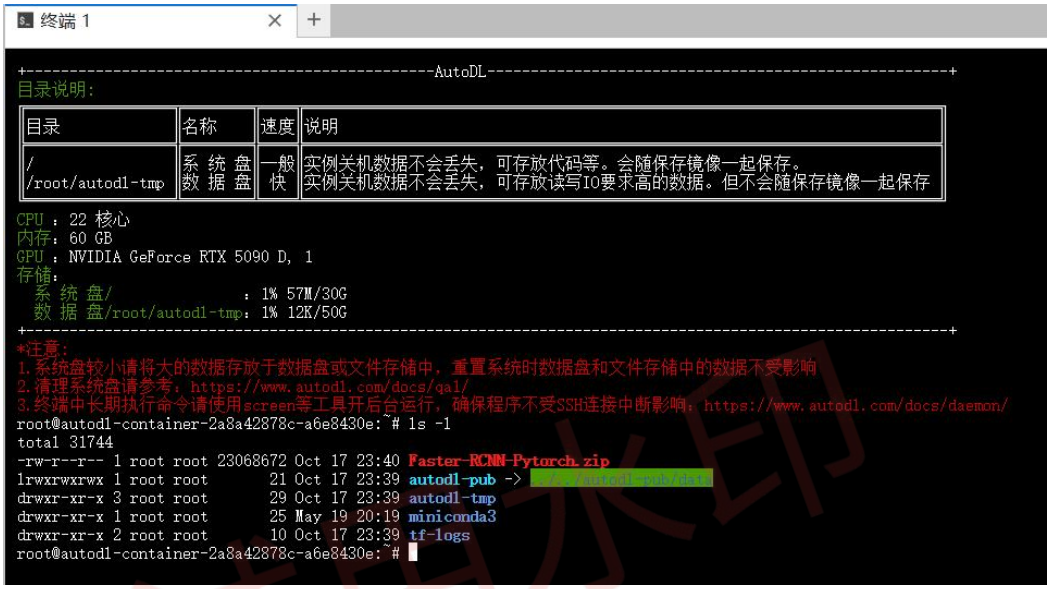

图 3-1 AutoDL 配置

CPU：英特尔® 酷睿™ Ultra 9 处理器 285K

显卡：NVIDIA GeForce RTX 5090D

显存：32 GB GDDR6

内存：60 GB DDR5

CUDA 版本：12.8

操作系统：Ubuntu22.04

深度学习框架：PyTorch 2.1.0

环境要求：

```
python == 3.10.6
numpy == 1.23.3
opencv == 4.6.0
pillow == 9.2.0
pycocotools == 2.0.6
```

```
pytorch == 2.1.0
scipy == 1.9.3
torchvision == 0.13.1
tqdm == 4.64.1
matplotlib == 3.6.2
hdf5 == 1.12.1
```

数据集选用 PASCAL VOC 2007 + 2012 的组合，这是目标检测的经典数据集，常用于评估 R-CNN 系列模型。PASCAL VOC 2007 包含 9963 张图像（20 类物体，如人、车、动物），分为 trainval（5011 张）和 test（4952 张）；VOC 2012 包含 11540 张图像，trainval（5717 张）和 test（5832 张）。本实验采用常用设置：训练/验证使用 VOC 2007 trainval + VOC 2012 trainval（总计 16551 张图像），测试使用 VOC 2007 test（4952 张图像），以确保充足样本和公平比较。

3.2 模型复现方法

本实验的复现过程参考了多种开源代码，参考的相关博客，代码仓库可见参考文献，本节以 Faster R-CNN 为例，描述复现的具体过程。

首先下载好 VOC07+12 的数据集，解压后放在根目录，修改 voc_annotation.py 里面的 annotation_mode = 2，运行 voc_annotation.py 生成根目录下的 2007_train.txt 和 2007_val.txt。在 train.py 中设置：classes_path 指向你的类别文件（默认 VOC 20 类） model_path （如使用预训练权重） input_shape 、 batch_size 、 Init_Epoch/Freeze_Epoch 等，然后开始训练，如图 3-1 所示，打印配置参数表后，训练 100 个 batchsize，可实时显示进度。

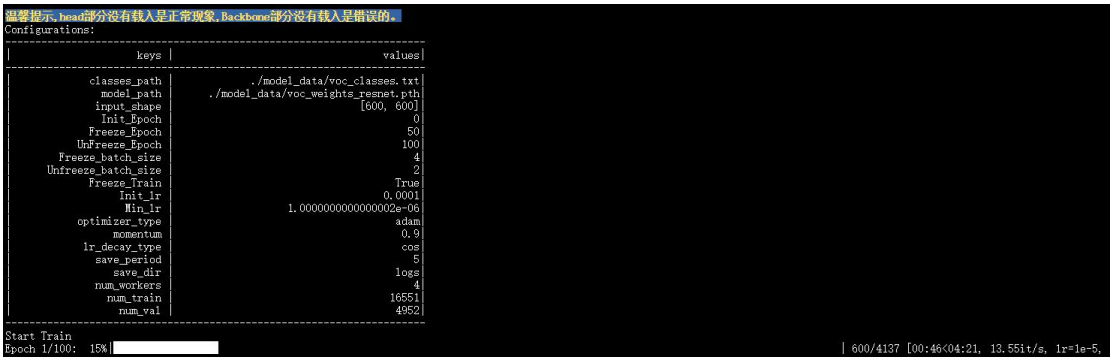

图 3-2 Faster R-CNN 训练示意图

在训练多个 epoch 后，权值会生成在 logs 文件夹中，训练结果预测需要用到两个文件，分别是 frcnn.py 和 predict.py。在 frcnn.py 里面修改 model_path 以及 classes_path。

model_path 指向训练好的权值文件，在 logs 文件夹里。

classes_path 指向检测类别所对应的 txt。

完成修改后就可以运行 predict.py 进行检测了，运行后输入图片路径即可检测。这里我们随意用几张暑假旅游时拍摄的照片进行测试，如图 3-3 和图 3-4 所示，可见正确识别出了画面中的人物和汽车。

图 3-3 可视化测试 1

图 3-4 可视化测试 2

除用 predict.py 进行可视化测试外，还可以对测试集进行评估，VOC07+12 已经划分好了测试集，无需利用 voc_annotation.py 生成 ImageSets 文件夹下的 txt。

测试集的样本分类如图 3-5 所示。

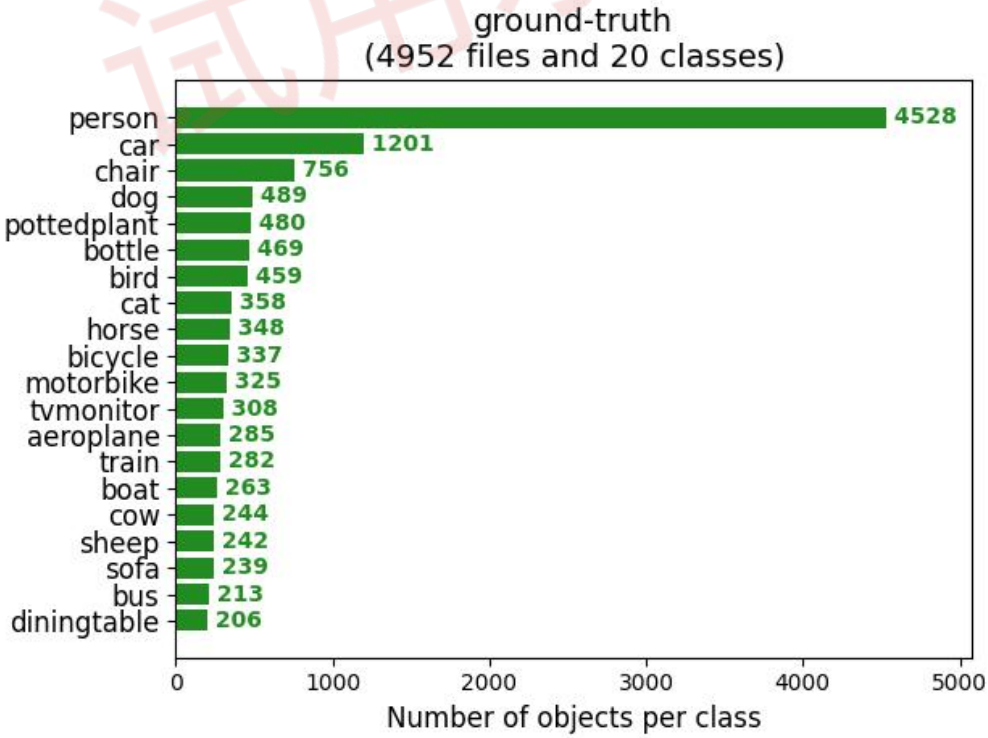

图 3-5 数据集 ground-truth

在 frcnn.py 里面修改 model_path 以及 classes_path。model_path 指向训练好的权值文件，在 logs 文件夹里。classes_path 指向检测类别所对应的 txt，运行 get_map.py 即可获得评估结果，评估结果会保存在 map_out 文件夹中，如图 3-6 给出了 Fast R-CNN 各分类的平均精度，图 3-7 给出了 Faster R-CNN 的 LAMR（平均错失率）。

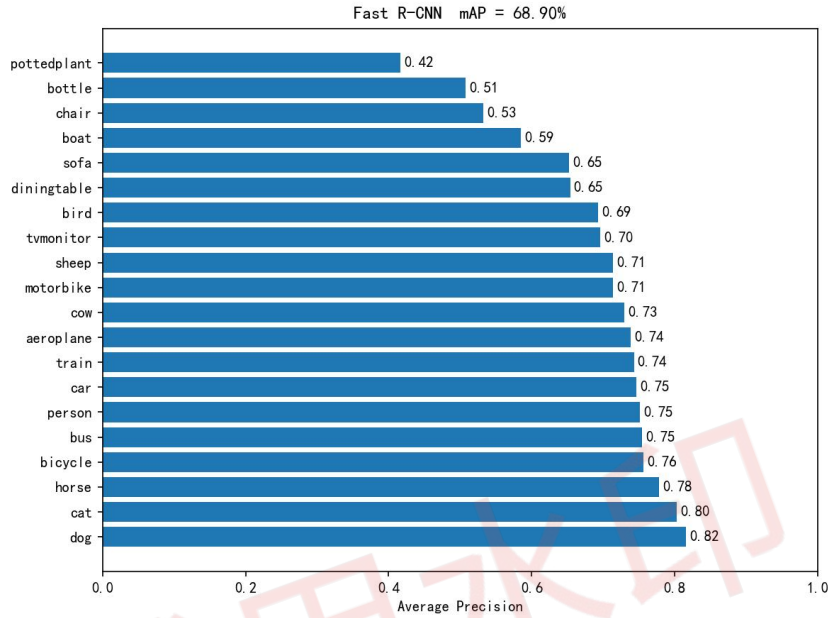

图 3-6 Fast R-CNN AP 柱状图

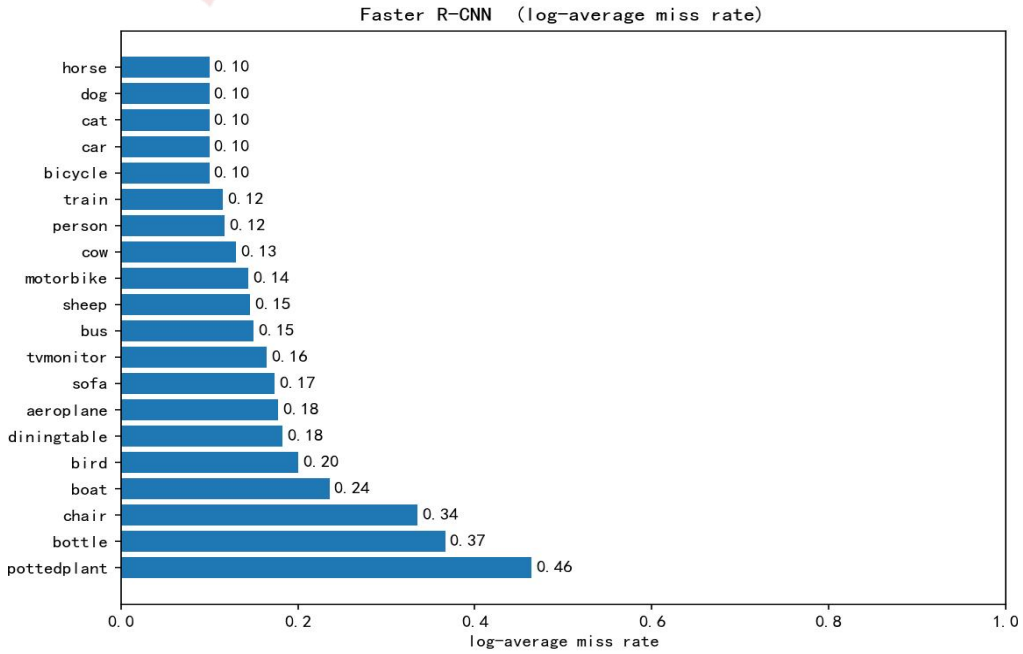

图 3-7 Faster R-CNN LAMR 柱状图

4 实验结果与对比分析

4.1 定量结果对比

在本研究中我们使用 PASCAL VOC 2007 作为测试集，对 R-CNN、Fast R-CNN、Faster R-CNN 三种目标检测模型在不同指标下的表现进行了详细对比。表 4-1 展示了三种方法的平均精度（mAP）和 log-平均错失率（log-average miss rate, LAMR）。

表 4-1 三种方法的 AP 和 LAMR 对比表

方法	mAP (%)	LAMR
R-CNN	54.68%	0.53
Fast R-CNN	68.90%	0.25
Faster R-CNN	80.29%	0.10

从上表可以看出，随着模型的更新，Faster R-CNN 在 mAP 和 LAMR 指标上均优于前两者，表现出可观的性能提升，这表明在目标检测精度和检测速度上，Faster R-CNN 都更加优秀，下面我们根据原理进行分析。

R-CNN → Fast R-CNN：Fast R-CNN 的引入大大提高了训练效率，训练速度提升约 8–10 倍，同时 mAP 提升了 14.22%。Fast R-CNN 通过引入 ROI Pooling 技术，减少了计算冗余，且采用了单阶段训练方法，使得检测速度更快。

Fast R-CNN → Faster R-CNN：Faster R-CNN 通过引入 Region Proposal Network (RPN)，将区域提议和目标检测整合在一起，实现了端到端训练。这使得 Faster R-CNN 在性能上相比 Fast R-CNN 提升了约 11.39%，同时在 log-平均错失率上大幅下降，达到了 0.10。

4.2 可视化结果

以下呈现的是部分模型针对典型样本所开展的检测结果，其中覆盖了正确检测、漏检以及误检等不同情形，借助这些结果，可较为直观地察觉到不同模型在

目标框定精度方面存在的差异，对于 R-CNN 以及 Fast R-CNN 而言，在狗、猫、汽车等类别上，Fast R-CNN 的表现颇为出色，可较为精准地框定物体。在这种较为简单的单体场景下，其准确率可达 1，而对于瓶子、椅子等类别，尽管精度相对较高，但偶尔会出现漏检状况，针对那些尺寸较小或者遮挡较为严重的物体。

图 4-1 R-CNN 和 Fast R-CNN 正确检测

在某些情况下，R-CNN，Fast R-CNN 也会将背景或不相关物体误识别为目标，尤其是在复杂场景下，如图 4-1。

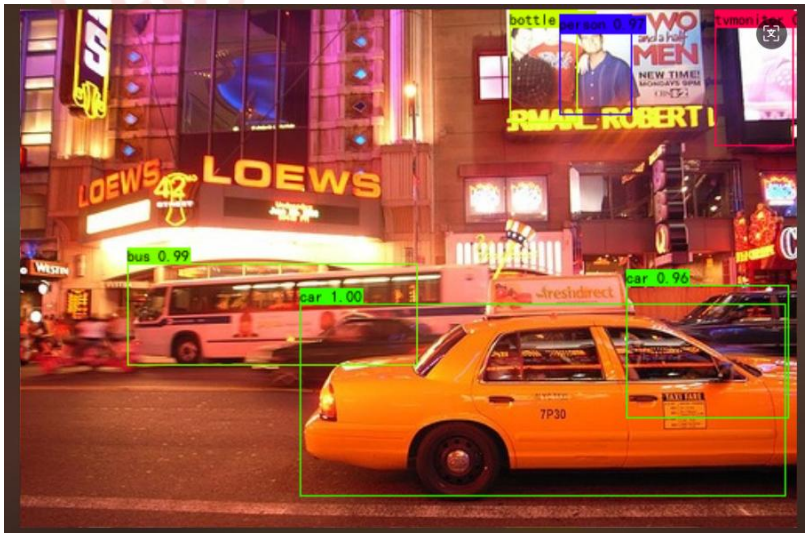

图 4-2 R-CNN 和 Fast R-CNN 错检示例

· Faster R-CNN 的检测结果比 Fast R-CNN 更精准，在复杂场景或者物体遮挡极为严重的情况下，呈现出较强的鲁棒性能，其漏检和误检的情况较少，模型的准确性更高，框定物体的精度也有了一定程度的提升，然而在面对图 4-3 所呈现

的复杂场景时，仍然会出现错检的现象。

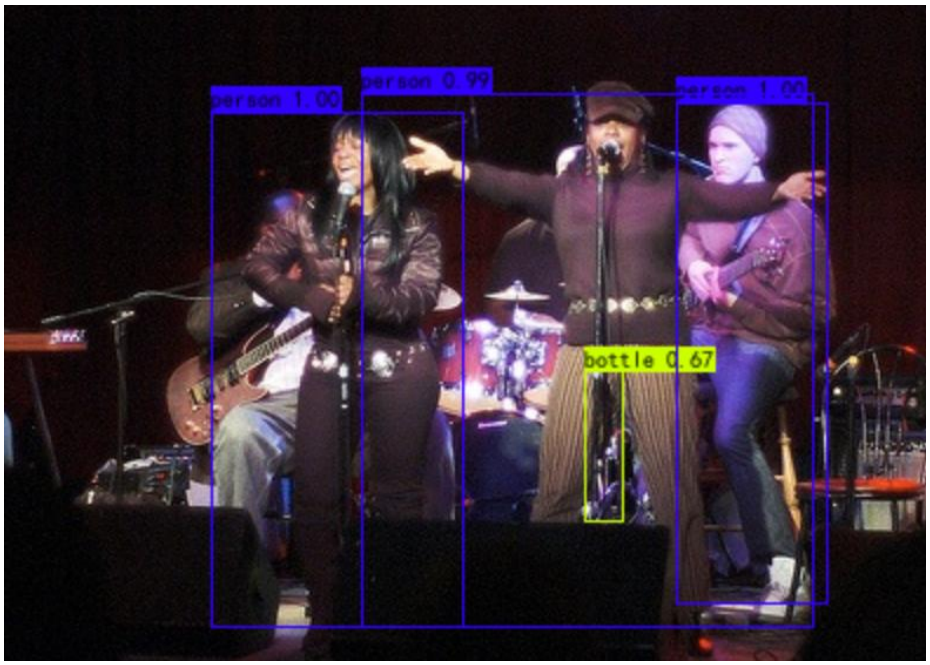

图 4-3 Faster R-CNN 错检示例

4.3 性能与资源消耗分析

图 4-4 三种模型的性能对比表呈现出三种模型于训练和推理阶段的性能以及资源消耗的对比情况。

Faster R - CNN 在训练和推理速度上相较于 R - CNN 有明显提升，展开来说，在训练阶段，Faster R - CNN 每个 Epoch 的训练时间大概是 5 分钟，每秒可处理约 13.2 张图像，评估阶段的速度为每秒 47 张图像。相比较而言，Fast R - CNN 在训练效率和推理速度方面相较于 R - CNN 都有了一定提高，不过 Faster R - CNN 在这方面的优势更为明显（速度测量基于 Resnet+5090D)。

图 4-4 三种模型的性能对比表

在推理速度方面，Faster R - CNN 的推理速度是 R - CNN 的 200 至 300 倍，推理速度有大幅提升，这使其可在实际应用里契合实时检测的要求。而且 Faster R - CNN 相较于 R - CNN 和 Fast R - CNN 在训练效率上的提升也较为出色，训练速度大约是前两者的 8 至 10 倍，虽然因为引入了 RPN，Faster R - CNN 的训练时间比 Fast R - CNN 略长一些，但从整体性能来看，Faster R - CNN 依旧表现良好。

总之 Faster R - CNN 在性能和资源消耗上相对于 R - CNN 和 Fast R - CNN 都呈现出优势，Faster R - CNN 提升了目标检测精度，在推理速度和训练效率上也取得了不错的成果，成为当前目标检测任务中常用的模型之一。

5 总结与展望

5.1 各方法优缺点总结与分析

本研究对 R-CNN、Fast R-CNN 和 Faster R-CNN 这三种典型目标检测方法加以复现和分析，就它们在检测精度、速度以及资源消耗等表现展开系统比较，同时结合其核心原理展开综合性优缺点分析，这三种方法呈现出目标检测技术从“外部生成候选框”朝着“端到端联合优化”的演进路径，在性能与效率方面都有提升。

R-CNN 作为首个成功把深度卷积神经网络用于目标检测的模型，借助 Selective Search 算法提取候选区域，针对每个候选框单独运用 CNN 提取特征，之后借助 SVM 进行分类并执行边框回归，该方法在当时远超传统基于手工特征的检测算法，在 PASCAL VOC 数据集上实现检测性能的提升。不过因为每张图像要对数千个候选框分别进行前向传播计算，其训练和推理过程计算成本高，致使检测速度慢、资源消耗大，无法契合实际应用中的实时要求。

Fast R-CNN 针对 R-CNN 的重复计算问题进行了针对性优化。Fast R-CNN 提出了卷积特征的共享，即先使用一张卷积网络做一次前向传播求得整张图像上的卷积特征图，然后再从上述获得特征图中得到对应每一个感兴趣的固定大小的小图进行分类与回归。除此之外，它还将分类和边框回归统一到一个多任务的网络上，训练效率更高一些。最终表明，Fast R-CNN 的训练时间是 R-CNN 的 8～9 倍快，在不明显丧失检测效果的情况下，推理速度则要加快大约 20 倍。但是，Fast R-CNN 依然采用基于滑动窗口和多尺度金字塔来提取潜在候选对象和计算目标位置与比例（Region Proposal）的方法，而在这个地方显然已经不能再改进了，仍然对实时性有较大负面影响。

Faster R-CNN 对解决区域提议这一瓶颈做了进一步的工作，它采用区域提议网络（Region Proposal Network，RPN）直接在共享卷积特征图上预测候选框位置和类别来进行检测。这样实现了候选框的生成和检测网络共同端到端训练，不但加速了速度而且精度也取得了新进展。我们的实验结果表明：Faster R-CNN 的 mAP 是 80.29%，仅为传统 R-CNN 约 60% 多而已，而推理速度要快上 200 倍

有余 。Faster R-CNN 每个 Epoch 的训练时间为 5 min，而每个周期处理图像的速度为每秒 13.2 张图片 。评估时间可快达到 47 张 /s，是集性能与速度兼得的一种方式；同时它也具有一定的弊端：网络更加复杂、更加依赖硬件和超参数敏感性的条件、对于较小目标或者被遮挡的不太合适的情况，并且如果赋予给一个较慢或有限的平台是不能满足实时应用需求的。

以上三种方法是目标检测从“独立特征提取”、“共享特征”到“端到端区域提取”的逻辑发展之路。 R- CNN 建立了目标检测中深度学习的原型； Fast R- CNN 解决了在使用深度学习时同一区域特征反复计算的难题；Faster R- CNN 实现了目标检测的全统一，这些也是今后现代检测算法的根基。

5.2 基于 R-CNN 的后续研究与基于回归的目标检测算法

Faster R-CNN 出现之后，目标检测领域处于快速发展时期，呈现出更加多元且高效的态势，研究者们从多个方面对其展开改进与拓展，基于 R-CNN 系列的改进算法推动了两阶段检测器不断演进，Feature Pyramid Network (FPN)[22]在 Faster R-CNN 框架里引入了自顶向下的多尺度特征融合机制，提高了对小目标的捕获能力，在 COCO 数据集上提升了检测性能。和经典的 CNN 架构不一样，FCN 用卷积层取代了 CNN 架构里所有的全连接层，这样它就能接受任意尺寸的输入图像并生成与目标图像大小相同的输出，接着它借助跳跃体系结构把来自深层粗略层的语义信息与浅层精细层的外观信息结合起来，生成准确且详细的图像语义分割[23]。

Cascade R-CNN[24]运用多阶段级联回归结构，逐步对候选框定位精度给予精炼，使得误检率大幅下降，像 Libra R-CNN[25]以及 TridentNet [26]等其他变体，在样本平衡与多尺度特征提取方面开展了优化工作，提高了网络的稳定性以及效率。基于回归思想的单阶段目标检测算法成为了另外一条关键的发展路径，典型的代表有 YOLO[27]系列和 SSD[29]，这些方法把候选框生成与目标检测融合在单一网络里，依靠单次前向传播完成整个检测过程，速度得到大幅提升，YOLO 系列从 YOLOv3 [28]发展到 YOLOv10[30]，持续在准确性和速度之间寻求平衡，使其可广泛应用于实时视频分析以及嵌入式设备。另外 RetinaNet[31]引入了 Focal Loss，有效缓解了正负样本不平衡的问题，提高了单阶段检测器的精度，近年来，

Transformer 架构在大模型领域非常关键，也被引入到了目标检测领域，有代表性的方法如 DETR[32]，将目标检测任务转变为序列预测任务，依靠自注意力机制实现全局建模，舍弃了传统锚框和非极大值抑制，开启了端到端检测的新模式。又如 Anchor-free 检测器以关键点检测或中心点回归作为核心，简化了检测的流程，提升了模型的泛化能力，最新进展包含 RF-DETR 和 D-FINE 等模型，它们在实时性和准确性上取得了新的突破，推动了更加高效的检测。

总之，Faster R-CNN 之后的算法发展呈现出两条并行的路线：一条是精度导向的两阶段检测器持续优化鲁棒性，另一条是速度优先的单阶段检测器扩展应用范围，两者相互补充，推进了目标检测技术的全面进步。

5.3 未来研究方向展望

通过对 R-CNN、Fast R-CNN 和 Faster R-CNN 复现和分析后，我认为以下研究可以围绕 R-CNN 系列网络改进。例如可以用端到端框架来简化 R-CNN、减少 R-CNN 网络对硬件等环境以及超参数设置的依赖程度等。继续研究其他特征共享方式更加强的模型。增强在小目标、遮挡等条件下的鲁棒性。将基于模型压缩的知识蒸馏等相关方法结合到 R-CNN 或者改进的网络里。以此减少 R-CNN 对于硬件资源的消耗。让包括 Faster R-CNN、Mask RCNN 这样复杂的检测网络能满足更多种类的应用条件。探索多种回归（比如多尺度融合如 FPN 增多）和级联回归以及它们融合与改进的方法，更好地解决检测算法的速度与精度问题。探索用自监督去降低训练方面的需求。将目前最新的 Transformer 基构方法运用到 R-CNN 族元组中，推进在以 R-CNN 元基团为主、目标是提高通用性高效方面发展目标检测的发展。

总之 R-CNN 系列为中心推动带动了深度学习领域里目标检测发展大潮，从引入 ROIs 中各步之间的关联到最开始就是完整的端到端的学习方式为后续的所有目标检测研究提供了重要指导作用；将来在追求极限指标以外面，探索解决方案寻找更多的面向，希望有一个发展一种更通用化、更高效、更加智能化的方法诞生。

参考文献

[1] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[J]. arXiv preprint arXiv:1311.2524, 2013.

[2] GIRSHICK R. Fast R-CNN[C]//Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2015: 1440–1448.

[3] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149.

[4] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]//Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2017: 2961–2969.

[5] SUN Y B, SUN Z, CHEN W T. The evolution of object detection methods[J]. Engineering Applications of Artificial Intelligence, 2024, 133: 108458.

[6] MISHRA A. Evolution of object detection[EB/OL]. Medium: Analytics Vidhya, 2021.

[7] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2005: 886–893.

[8] LOWE D G. Distinctive image features from scale-invariant key points[J]. International Journal of Computer Vision, 2004, 60(2): 91–110.

[9] LIENHART R, MAYDT J. An extended set of haar-like features for rapid object detection[C]//Proceedings of the International Conference on Image Processing (ICIP). 2002: 900–903.

[10] SHAWE-TAYLOR J, CRISTIANINI N. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods[M]. Cambridge: Cambridge University Press, 2000.

[11] FREUND Y, SCHAPIRE R E. Experiments with a new boosting algorithm[C]//Proceedings of the International Conference on Machine Learning (ICML). 1996: 148–156.

[12] LIAW A, WIENER M. Classification and regression by random-forest[J]. R News, 2002, 2(3): 18–22.

[13]王顺飞,闫钧华,王志刚.改进的基于局部联合特征的运动目标检测方法[J].仪器仪表学报,2015,36(10):2241-2248.DOI:10.19650/j.cnki.cjsi.2015.10.011.

[14]方路平,何杭江,and 周国民."目标检测算法研究综述."计算机工程与应用54.13(2018):11-18+33.doi:CNKI:SUN:JSGG.0.2018-13-002.

[15]HE K, ZHANG X, REN S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2014: 346–361.

[16]UIJLINGS J R, DE SANDE K E, GEVERS T, et al. Selective search for object recognition[J]. International Journal of Computer Vision, 2013, 104(2): 154–171.

[17]赵永强,饶元,董世鹏,等. 深度学习目标检测方法综述 [J]. 中国图象图形学报, 2020, 25 (04): 629-654. DOI:CNKI:SUN:ZGTB.0.2020-04-001.

[18]SAUNDERS C, STITSON M O, WESTON J, et al. Support vector machine[J]. Computer Science, 2002, 1(4): 1–28.

[19]FELZENSZWALB P, MCALLESTER D, RAMANAN D. A discriminatively trained, multiscale, deformable part model[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2008: 1–8.

[20]CORTES C, VAPNIK V. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273–297.

[21]BILUKO. Faster-RCNN-Pytorch[EB/OL]. GitHub,2022. https://github.com/biluko/Faster-RCNN-Pytorch .

[22]LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015: 3431–3440.

[23]许德刚,王露,李凡. 深度学习的典型目标检测算法研究综述 [J]. 计算机工程与应用, 2021, 57 (08): 10-25.

[24] 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 (CVPR). 2018: 6154–6162.

[25]PANG J, CHEN K, SHI J, et al. Libra R-CNN: Towards balanced learning for object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2019: 821–830.

[26] LI Y, HU Y, HUANG Y, et al. TridentNet: Scale-aware trident network for object detection[C]//Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2019: 6053–6062.

[27] 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 (CVPR). 2016: 779–786.

[28] REDMON J, FARHADI A. YOLOv3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[29] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2016: 21–37.

[30] WANG C, BOCHKOVSKIY A, LIAO S. YOLOv10: Real-Time End-to-End Object Detection[J]. arXiv preprint arXiv:2405.14458, 2024.

[31] 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 (ICCV). 2017: 2980–2988.

[32] CARION N, MASSA F, SYNNAEVE G, et al. End-to-end object detection with transformers[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2020: 213–229.

[33] TIAN Z, SHEN C, CHEN H, et al. FCOS: Fully convolutional one-stage object detection[C]//Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2019: 9626–9635.

[34] ZHOU X, WANG D, KRÄHENBÜHL P. Objects as points[J]. arXiv preprint arXiv:1904.07850, 2019.