天津大学

本科生毕业论文

题目：动态环境下稳定目标识别研究

学　　院　智能与计算学院

专　　业　人工智能专业

年　　级　2019 级

姓　　名　李盈盈

学　　号　3019244109

指导教师　刘若楠

独创性声明

本人声明：所呈交的毕业论文，是本人在指导教师指导下，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，本毕业论文中不包含任何他人已经发表或撰写过的研究成果。对本毕业论文所涉及的研究工作出贡献的其他个人和集体，均已在论文中作了明确的说明。本毕业论文原创性声明的法律责任由本人承担。

论文作者签名：

年　　月　　日

本人声明：本毕业论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容。

论文指导教师签名：刘若楠

年　　月　　日

摘　要

近年来，越来越多的研究者开始关注动态环境下的稳定目标识别问题，即如何在变化的环境中准确地识别和跟踪目标。这是一个具有挑战性的问题，因为在动态环境中，目标通常会发生形变、遮挡、位移等情况，同时还可能存在光照变化、天气变化等外部干扰因素。为了解决这些问题，我们将基于因果推断的稳定学习方法引入到目标检测领域中。传统的目标检测模型通常基于独立同分布假设，与传统的模型不同，我们的研究提出了一种可以适应不同域之间分布差异的模型，从而提高了模型的泛化能力和鲁棒性。特别地，本研究将该方法应用于单阶段目标检测框架中，在具有域偏移性质的数据集上取得了较好的实验结果。通过本研究，我们期望能够为动态环境下的稳定目标识别问题提供一种新的解决思路，并促进深度学习在实际应用场景中的推广和发展。

关键词：稳定学习，因果推断，目标检测，域适应，域泛化

ABSTRACT

In recent years, more and more researchers have begun to pay attention to the problem of stable object recognition in dynamic environment, that is, how to accurately identify and track objects in a changing environment. This is a challenging problem because in a dynamic environment, objects often undergo deformation, occlusion, displacement, etc. , and there may also be external interference factors such as lighting changes and weather changes. In order to solve these problems, we introduce stable learning methods based on causal inference into the field of object detection. The traditional model for object detection usually based on the Independent Identity Distribution (I.I.D.) hypothesis. Different from them, our study proposes a model that can adapt to the distribution differences between different domains, which achieves better generalization ability and robustness. In particular, this method is applied to the one stage object detection model, and good experimental results are achieved on datasets with domain shift properties. Through this study, we hope to provide a new solution to the problem of stable object recognition in dynamic environment, and promote the promotion and development of deep learning in practical application scenarios.

KEY WORDS: Stable Learning, Causal Inference, Domain Adaption, Domain Generalization

目 录

第一章　绪论

1.1　研究背景和意义

1.1.1　研究背景

目标识别技术是计算机视觉领域的一个热门方向，该技术广泛应用于机器人导航、智能视频监控、工业检测、航空航天等诸多领域。目前，将神经网络与深度学习技术应用到该领域以减少对人力资源的消耗已经具有十分重要的现实意义。比如，在机器人导航领域，[44]提出了一种基于 RGB-D 相机的实时语义地图构建方法，通过利用深度学习估计的场景语义流来提高动态物体的识别和追踪效果；在智能视频监控领域，[45]提出了一种基于深度学习和物联网技术的智能视频监控系统，该系统可以实现多目标跟踪、事件检测、异常行为分析等功能，具有高精度和高实时性，可以在复杂环境下应用于安防领域；在工业检测领域，[46]提出了一种基于深度学习的 X 射线图像缺陷检测方法，该方法可以自动检测图像中的缺陷区域，并对缺陷进行分类和定位；在航空航天领域，[47]提出了一种基于卷积神经网络的无人机航空目标检测方法，该方法可以实现在复杂环境下对目标的自动识别与跟踪。

然而，现有的目标识别技术大多数基于独立同分布假设，在该假设下训练的模型仅能在源域与目标域具有相似的图像空间数据分布的情况下取得较好的测试效果。但是在实际应用中，用于训练的源域数据往往与用于测试的目标域数据有着不同的分布，这种分布差异可能还会比较大，比如训练集图像中的目标所处的背景环境在测试集中发生了变化，这种情况下模型的稳定性将受到挑战。

1.1.2　研究意义

我们的研究主要把基于因果推断的稳定学习方法应用到现有的目标识别技术中，旨在提高模型的稳定性、鲁棒性与泛化能力。同时，通过对域适应和域泛化等经典问题进一步研究，我们的研究能在一定程度上缓解各种因素导致的目标识别困难程度提高的问题，这些因素包括待识别目标状态多变等内部因素以及目标所在背景环境多变等外部因素。具体来说，一方面，我们的研究有利于目标识别技术更好地投入到实际应用中；另一方面，我们的研究将对目标识别方向、稳定学习方向、域泛化及域适应方向的理论体系进行完善，一定程度上能弥补当前这些研究方向上的某些不足。

具体来说，目前解决域适应和域泛化问题主要从数据操作、表征学习和学习策略等角度切入（[1]），相比其他方法，基于因果推断的稳定学习的方法在目标识别领域中的应用是一个新的探索，可以为后世的理论发展与技术探索开创新的思路。

1.2 国内外研究现状

与我们的研究相关的已有的比较成熟的技术主要涉及到目标识别、域适应和域泛化，因此，我们的国内外研究现状将从这三个方面分别进行阐述。

1.2.1 目标识别

广义的目标识别即目标检测，目标检测（Object Detection）是计算机视觉中的一个重要问题，它的主要任务是在图像或视频中，识别出物体所属类并定位出物体的位置。目标检测的主要目的是从含有单个或者多个物体的图像中自动识别出感兴趣的物体。在目标检测任务中，需要定位每个检测到的物体并将其标记出来，通常使用边界框（Bounding Box）来表示物体的位置、大小和姿态等信息，在检测时对图像进行处理、分析和预测，最终输出所有检测到的目标的类别信息及其具体的位置坐标。

目标检测早期采用的是基于手工设计的特征和传统的机器学习方法。传统的手工设计特征包括 Haar 特征、HOG 特征和 SIFT 特征等，传统的目标检测器包括 Viola Jones 检测器、HOG 检测器、Deformable Part-based (DPM)检测器等。这些方法具有良好的鲁棒性和可解释性，但由于人工特征设计的局限性，限制了它们在复杂场景下的应用。

随着深度学习的兴起，目标检测领域发生了巨大变化，早期诞生的是一些多阶段的目标检测模型。2014 年 RCNN 模型（[6]）被提出，它的核心流程是将输入图像分成许多候选区域，并使用选择性搜索（SS, Selective Search）算法来生成这些候选区域，然后，它使用预训练的 CNN 模型（如 AlexNet 或 VGGNet）来提取每个候选区域的特征，并将这些特征送入一个支持向量机（SVM）进行分类。随后的 Fast R-CNN（[7]）、Faster R-CNN（[8]）、Mask R-CNN（[9]）等模型都是在 RCNN 模型的基础上进行改进的，这些改进了的多阶段目标检测模型加快了训练和推理速度，同时提高了检测和分割精度，但是这些模型仍然存在一些不足：

1、训练和推理速度慢：由于需要逐个处理每个候选区域，模型训练和推断速度较慢，这限制了多阶段目标检测器在实际应用中的实时性。

2、复杂的训练流程：由于需要分别训练 CNN 特征提取器、候选区域生成器和分类器等多个组件，多阶段目标检测器的训练流程较为复杂。

3、目标尺寸不同引起的性能下降：多阶段目标检测器通常只能预测一种尺寸的目标框，这意味着如果目标具有不同的尺寸或宽高比，则需要使用多个不同的模型来进行检测，从而增加了系统的复杂性。

在多阶段目标检测器诞生后不久又诞生了一系列的单阶段目标检测器，且目前的大多数主流的目标检测模型都属于单阶段目标检测器。与多阶段目标检测器截然不同的是，这一类检测器通常只需要一次前向传播就可以同时实现定位和分类，因此具有训练和推断速度快、易于实现端到端训练等优点。常见的单级目标检测器包括 YOLO 系列（[10]，[11]，[12]，[13]，[14]，[15]）、SSD（[16]）和 RetinaNet（[17]）。

具体来说，YOLO（You Only Look Once）是一种基于卷积神经网络（CNN）的单阶段目标检测器，它将图像分成网格，并在每个网格上预测目标框和类别得分，通过减少候选框的数量来提高检测速度和精度，并使用多尺度训练策略来提高检测性能。

SSD 使用不同尺度的特征图来检测不同大小的目标，并且利用多个锚框（anchor）来进行检测，它通过使用多个尺度和多个最大池化层来保留更多的空间信息，从而提高检测精度。

RetinaNet 是一种使用特殊的损失函数——Focal Loss 来解决类别不平衡问题的单阶段目标检测器，它通过使用特定的卷积层 FCOS 来同时输出目标框位置和类别信息，并通过改变损失函数的权重分配方式，使得正样本的损失权重逐渐减少，从而有效地解决了类别不平衡问题，提高了检测精度。

当然，随着这三个系列不断更新出不同版本，各个系列的不同版本在实施细节上还是有所差异的，并且不断衍生出许多新技术来优化模型的性能，比如 YOLO 系列在 YOLOv3 上就已经学习 SSD 采用了多尺度技巧，等等。

总体来说，相对于多阶段目标检测器，单阶段目标检测器虽然速度快，但是定位精度略低，并且对小目标的检测效果不如多阶段目标检测器好。

1.2.2 域适应与域泛化

域适应（Domain Adaptation）和域泛化（Domain Generalization）是计算机视觉领域中两个重要的概念。它们都涉及到将一个训练集中的知识或经验应用于另一个测试集合，但两者的研究问题略有不同。

域适应可以认为是迁移学习的一种，在机器学习中，当源域和目标域数据分布不同，但是两者的任务相同时，可以使用域适应，将在有着大量样本源域数据

上训练得到的的精度较高的模型运用到分布存在差异的目标域中。目前已有的域适应学习方法主要分为五类（[19]）：

第一类是使用实例加权进行自适应。该方法的局限性是仅适用于源域与目标域分布差异较小的情况，其子类方法包括基于联合训练加权、基于核映射加权和直观加权等。

第二类是特征自适应，其子类方法包括特征子空间对齐、特征转换、特征重构、特征编码等。

第三类是分类器自适应，其子类方法主要包括基于核的分类器、基于流行正则器的分类器和基于贝叶斯的分类器，该类方法主要有三个缺点：1）目标数据不正确的伪标签会显著降低模型性能；2）在估计各种潜在变量时，不准确的分布假设会产生非常大的消极影响；3）域间流形假设在严重的域分布差异情况下不成立。

第四类是深度网络自适应，其子类主要包括基于边缘对齐的方法、基于条件对齐的方法和基于自编码器的方法，该类方法的主要缺点包括：1）需要大量有标记的源数据来训练(微调)深度网络；2）当域分布差异很大时，未标记目标样本预测到 k 类的置信度有时很低；3）深度适应的可解释性不容乐观，易导致负迁移。

第五类是对抗自适应，其子类主要包括基于梯度反转与极大极小优化的方法和基于生成对抗网络 GAN 的方法，比如[20]提出了一个新的无监督领域自适应方法——协作对抗网络（Collaborative and Adversarial Network，CAN），这个方法结合网络的域协作性和域对抗性来进行训练。该类方法的主要缺点有三点：1）域判别器容易过度训练；2）只最大化域混淆度容易导致类别偏见；3）特征生成器与判别器之间的博弈依赖于人。

总的来说，未来在域适应领域中，一个比较有发展前景的方向是将样本或者特征重加权、深度网络和对抗策略三大方法集成来解决域偏移问题。

相比域适应，域泛化则更进一步，它是分布外泛化（OOD，Out Of Distribution）问题的一种。域泛化假设模型的输入为来自多个源域的数据集，与域适应相同的是，域泛化的目标域与源域也有着不同的分布，但是与之不同的是，域泛化场景下目标域分布未知，而这在域适应场景下是已知的。域泛化希望模型能够学到域无关的特征，这种特征可以容易地泛化到新的测试数据域上。

目前已有的解决域泛化问题的方法已经很多。具体来说，领域对齐的方法通常是去对齐源域之间的特征分布，其设计的主要驱动力是：如果学到的特征对源域之间的分布偏移不敏感，那么该特征也应该对目标域的分布偏移具有较好的鲁棒性。目前绝大多数的领域泛化方法都是基于领域对齐这个思想开发的，比如[21]

提出了一种新型跨领域特征学习方法，可以通过对齐源域和目标域之间的特征分布来提高模型的泛化能力。

基于元学习的方法的核心是把多个源域随机分成伪源域（pseudo-source domain）和伪目标域（pseudo-target domain），然后利用元学习的算法去优化伪源域上的目标函数，使得模型在伪目标域上的性能有所提升，以此来激发模型学习泛化性强的特征。例如，[22]提出了一种基于元转移学习的少样本学习方法，可以通过在多个域之间共享知识来提高模型的泛化能力。

集成学习的方法学习多个模型或模块，然后在测试的时候做融合。例如，[23]提出了一种深度集成网络方法（DCN，Deformable Convolution Networks），可以通过将多个源域数据集组合起来进行联合训练来实现跨领域知识转移。

表征分解学习的方法通过学习如何分解出域分布相关特征和域分布不相关特征解决域泛化问题。比如，[24]提出了一种基于深度自编码器的表征分解学习方法，可以在多个源域和目标域之间实现跨领域知识转移，该方法通过学习一个映射函数，将每个样本表示成域无关特征和域相关特征，并利用这些特征来训练分类器。

不变风险最小化的方法主要是从 IRM 算法上衍生出来的，其设计目的是减少模型对非相关元素（如图像背景）的依赖。比如[31]提出了一种基于 IRM 算法的扩展方法，可以应对非平稳环境和多个分布之间的领域转移，该方法通过学习一个置信集合来识别可靠的特征。

除此以外，还有数据增强、网络架构设计、自监督学习、启发式训练等方法来解决域泛化问题。

但是，以上介绍的这些传统的域适应和域泛化的问题主要存在以下几点不足：

1、假设限制：传统的方法通常基于独立同分布假设，即训练集和测试集之间具有相同的数据分布。然而，在真实世界中，这种假设很难得到满足，因此传统方法的性能可能会下降。

2、特定问题的依赖：传统的方法通常对特定问题进行优化，无法处理广泛多样的应用场景，因此缺乏通用性。

3、数据偏置问题：由于数据的不平衡或采样错误，模型可能存在数据偏置问题，传统的方法不能很好地应对该问题。

相比而言，基于因果推断的稳定学习方法是一种用于解决域适应和域泛化问题的新方法，其相对于传统方法具有以下优势：

1、从因果关系出发：该方法通过建立因果图模型，将不同领域之间的差异量化为因果效应，并针对这些效应进行学习。由于因果关系是真实世界中最基本

的关系之一，因此这种方法可以更准确地描述数据之间的关系，并提高模型的泛化能力。

2、更好的鲁棒性：由于基于因果推断的稳定学习方法考虑了不同领域之间的因果效应，并在此基础上进行学习，因此可以更好地抵抗数据之间的扰动和噪声，提高模型的鲁棒性。

3、增强可解释性：因果图模型可以清晰地展示变量之间的因果关系，从而增强了模型的可解释性和可理解性。这对于一些需要解释和说明的应用场景非常重要。

因此，我们的研究将基于因果推断的稳定学习方法引入到目标检测框架中，用于解决目标检测任务中出现的域适应与域泛化问题，相比传统的方法比较具有进步性和创新性。

1.3 研究目的和内容

1.3.1 研究目的

目标检测是计算机视觉领域的一个重要研究方向，其主要任务是在图像或视频中自动识别和定位特定的物体。近年来，随着深度学习技术的发展，目标检测技术的精度和效率得到了大幅提升。然而，目前的目标检测方法仍然存在一些问题，如容易受到噪声、遮挡、光照等因素的影响，导致检测结果不稳定。

为了解决这些问题，我们的研究将基于因果推断的稳定学习方法引入到目标检测中。将稳定学习方法引入到深度神经网络中可以使模型在输入数据发生扰动时，输出结果尽可能不变，从而有效地降低模型对于数据变化的敏感性，提高模型的稳定性和鲁棒性。具体来说，我们的研究通过将稳定学习方法引入到目标检测框架中，可以达到以下几个目的：

1、增强鲁棒性：目标检测模型经常面临各种干扰和噪声，例如遮挡、光照变化、噪声等，这些因素会导致模型的输出结果不稳定。将稳定学习引入到目标检测中，可以有效地增强模型的鲁棒性，使其对噪声和扰动具有更好的适应性。

2、提高检测精度：现有的目标检测算法往往只适用于某些特定场景，而在其他场景下可能出现检测失败的情况。通过稳定学习，可以使目标检测模型适应更广泛的场景和复杂的背景，并提高检测的精度。

3、加快检测速度：目前的目标检测算法通常是基于深度学习的，计算量较大。通过引入稳定性学习，可以使模型专注于具有不变性的稳定特征，从而减少模型的计算量，提高检测速度，使得模型更加适用于实际场景的应用。

综上所述，我们的研究的主要目的是将基于因果推断的稳定学习方法引入到目标检测框架中，从而提高模型的稳定性、鲁棒性和精度，并加快检测速度，这有着广泛的研究和应用前景。

1.3.2 研究内容

一方面，从任务切入点和算法创新的角度来说，域泛化是比域适应更普遍的情况，并且可以认为域泛化囊括了域适应，因此，我们的研究把关注点放在解决域泛化问题上。针对域泛化问题，与已有方法不同，我们的研究创新性地把基于因果推断的稳定学习方法用在了目标检测任务上，并取得了不错的效果。

另一方面，从模型整体架构来说，我们的模型沿用了 YOLOv5 的模型主框架，使用单阶段目标检测方法进行目标分类和定位，保留了单阶段目标检测模型速度快、量级轻、参数相对较少的优势，同时加入稳定学习模块，提高了模型的精度、鲁棒性和泛化能力，在具有域偏移性质的目标检测数据集上取得了良好的测试效果。

第二章　相关工作

2.1 稳定学习

稳定学习的概念是在 2018 年的 KDD 会议上由清华大学副教授崔鹏团队提出的，它旨在解决这样一个问题：当我们用机器学习进行预测建模的时候，如果对测试数据集没有任何先验知识，如何保证模型在未知分布上作出稳定预测。

在稳定学习研究领域，由崔鹏教授等人在[25]中提出的 DGBR（Domain Generalization Balance Regression）算法首次解决了二元离散响应变量（特征）这一设定下的稳定预测问题，此后，一系列稳定学习方法被提出来，用于解决更多不同设定下的稳定预测问题。在这几年的发展中，稳定学习主要包括以下三种方法：

一是基于样本加权的变量去相关，包括况琨等人在[26]中提出的 DWR（Decorrelated Weighting Regression），它结合了变量去相关正则化与加权回归模型，解决了在连续的预测变量这一设定下模型的稳定预测问题；还有[27]提出了一种基于变量聚类的变量分解算法，该算法使用来自不同环境的未标注数据，被称为区分性变量去相关（Differentiated Variable Decorrelation，DVD），等等。

二是对抗稳定学习，包括崔鹏老师团队在[28]中提出的 SAL（Stable Adversarial Learning），该算法利用异构数据源构建更实用的不确定性集，并进行差异化鲁棒性优化，其中协变量根据其与目标相关性的稳定性进行区分，等等。

三是异构型风险最小化，包括崔鹏老师团队在[29]中提出的 HRM（Heterogeneous risk minimization），它是一种优化框架，可实现数据和不变预测器之间潜在异质性的联合学习，尽管数据的分布发生变化，模型在该框架下仍具有很好的泛化能力，等等。

总的来说，相比传统的迁移学习，稳定学习存在一定的区别：迁移学习的优化目标是在具有测试集分布先验知识的情况下，最大化模型的预测性能（准确率等），而稳定学习假设在测试数据中存在多个环境，因此对模型的泛化能力提出了更高的要求。因此，用稳定学习的方法解决域泛化问题是一个新的研究思路，这也是我们的研究中的工作重点。

2.2 基于因果推断的样本重加权策略

因果推断是一种与不变性相关的科学，我们在这里使用一个例子简单地解释一下因果推断：如图 2-1 所示，对于任何给定的特征 T，给样本附上不同的权重

使得带有特征 T 的样本与不带有特征 T 的样本具有相似的 X 分布，然后计算实验组 Y 与对照组 Y 的分布的差异，如果 Y 的分布变化明显，则说明 T 对 Y 有因果效应，此时可以估计出：平均而言， T 对 Y 的因果效应对于 X 的变化是具有不变性的。我们把这种使用某种手段消除或削弱不相关因素的影响，找到真正的因果关系的过程称为因果推断。

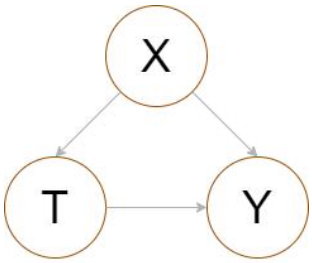

图 2-1 典型的因果推断框架

在因果推断中一个常用的方法是样本重加权。样本重加权可以使变量之间相互独立，而如果所有的变量在样本加权后相互独立，此时的相关性将等于因果性。为了将不变性适配到深度学习框架中，我们要研究多个输入变量对模型预测得到的输出变量的影响。稳定学习试图找到一组合适的样本权重，进行样本加权后再通过输入变量对输出变量进行回归，此时的回归系数即为满足因果关系的回归系数，通过上述方式训练出的模型将具有较好的泛化能力，在动态环境下有较强的稳定性。

第三章　基于稳定学习的动态环境下稳定目标识别

3.1　问题定义

3.1.1　目标识别

对于目标识别（Object Recognition），有狭义和广义两种理解形式。狭义的目标识别即图像分类任务，这种任务仅仅输出图像中物体所属类别，可以分为单标签识别和多标签识别，前者适用于一张图像中仅含有单个主要物体，每张图像仅仅对应一个类别标签；而后者适用于一张图像中存在多个主要物体，这种情况下一张图像对应着所有它包含的主要物体的类别标签。广义的目标识别即目标检测任务，除了对图像中的主要物体进行分类，还要对它们进行定位。我们的研究将目标识别定义为目标检测，这是因为目标检测是一个更高层面的任务，它在实际应用中也更多。

3.1.2　动态环境下的稳定性

深度学习是一种基于神经网络的机器学习算法，其在目标检测领域中也取得了重大进展。但是，由于实际应用中的数据通常是动态的，深度学习模型在动态环境下的稳定性成为了一个重要问题。

首先，动态环境指的是模型将在不断变化的数据流中进行推理，这些数据流的分布可能会随着时间发生变化。例如，在自动驾驶汽车中，道路上的交通情况随着时间而变化，这意味着模型必须能够持续适应新的数据流。

其次，为了提高深度学习模型在动态环境下的稳定性，有以下几个方面需要注意：

1、模型结构的选择：模型应该能够快速适应新的数据，并对数据流中的异常值具备鲁棒性。因此，在设计模型时需要考虑到动态环境的特点，例如可以采用增量式学习或增量式迁移学习等方法，来避免模型过度拟合新数据。使用增量式学习方法的工作在近几年也比较多，比如[2]提出了一种基于向量量化技术的增量式少样本学习方法，该方法可以快速地适应新类别，并利用已有的知识和经验进行推理和分类。

2、数据预处理：数据预处理是深度学习模型的重要组成部分，通过对数据进行筛选、清洗和归一化等预处理操作，可以有效提高模型在动态环境下的稳定性。

3、数据流的监控：监控数据流变化的趋势可以帮助我们及时发现潜在的问题。例如，监测输入数据的方差、对数据进行频率分析等，都可以有效检测到数据的异常行为或分布偏离。

4、模型的鲁棒性测试：在设计深度学习模型时，需要考虑到模型鲁棒性的因素。例如，通过对一些已知的干扰或攻击（如噪声、缺失数据、人为干扰等）进行测试，可以评估模型在动态环境中的稳定性，并在此基础上对模型加以改进。

最后，受以上四点的启发，我们的工作针对性地目标检测中的动态环境问题进行了深入研究。具体来说：

1、在模型结构的选择上，我们以 Yolov5 模型为主要框架，针对其速度快但精度有待提升的问题，我们将基于因果推断的稳定学习模块引入到主框架中，提高了模型的精度和鲁棒性。

2、在数据预处理上：首先，受[3]的启发，我们使用了自动数据增强技术，在进行数据增强时对图像进行随机裁剪、缩放和旋转等操作，来扩充训练数据集，从而提高模型的鲁棒性和泛化能力；其次，受[4]的启发，我们采用了标签平滑技术来减少模型过拟合的可能性，使得真实标签和假设标签之间的距离更加平滑，减少了分类预测时噪声的干扰；除此以外，受[5]的启发，我们还采用了数据归一化技术，将图像像素值从[0,255]的整数范围映射到[0,1]的浮点数范围内，以便更好地进行梯度计算和权重更新。

3、在数据流的监控方面，我们在探究预保存特征数量影响的消融实验中考察了样本权重方差，有效检测到：随着训练迭代次数的增加，学习到的样本权重的变化趋势是逐渐趋于平稳的，这说明此时基本已经通过样本加权去除了所有特征变量间的相关性。

4、在模型的鲁棒性测试方面，我们选用 Cityspaces 作为源域数据集，Foggy Cityspaces 作为目标域数据集，由于后者相当于在前者基础上增加干扰特征，这种训练集与测试集具有分布偏移的设置可以验证模型的鲁棒性。

3.2 研究创新

为了把基于因果推断的稳定学习方法适配到深度学习框架中，[30]中提出了两种方法：1）提出了一种基于随机傅里叶特征的非线性特征去相关方法，该方法具有线性计算复杂度。2）针对分批次训练问题，提出了一种有效的优化机制，通过迭代保存和重新加载模型的特征和权重，来全局感知和去除任意两两特征间相关性。

一方面，[30]所提出的模型 StableNet 仅仅适用于图像分类任务，而在真实的应用场景中，我们往往不仅仅需希望模型识别出图像中的单个主要物体，而且希望它能对图像中的多个物体进行定位；另一方面，现有的目标识别模型，无论是多阶段的还是单阶段，基本都以独立同分布假设为前提，并没有把基于因果推断的稳定学习方法适配到单阶段或者多阶段目标识别模型中相关工作。因此，我们的研究创新性的把[30]中提出的两种方法适配到单阶段目标检测框架中。

具体来说，在我们进行应用创新设计得到的模型中，基于因果推断的稳定学习模块（LSWD，Learning Sample Weights for Decorrelation）使用了基于随机傅里叶特征（RFF，Random Fourier Features）的样本重加权方法去除所有特征之间的线性和非线性依赖关系，使变量之间两两相互独立，从而在训练的过程中能很大程度上去除不相关特征与标签之间的协相关关系，使模型能够找到相关特征与标签之间的强因果关系。这种强因果关系具有不变性，这样做可以提高模型在动态环境下的稳定性。

在本章节中，3.2.1 小节首先对相关符号给出定义与说明；3.2.2 与 3.2.3 两小节将重点回顾一下[30]中提出的两种方法并介绍如何在我们的模型中使用它们；3.2.4 小节将详细介绍一下我们的模型的损失函数，这部分基本沿用的是 Yolov5 模型的损失函数（Yolov5 没有对应的论文，我们是根据官方代码理解它的损失函数的实现并在这一小节详细介绍的），但是我们在分类损失这一块儿做出了修改，主要是把前面学习到的样本权重加在分类损失上，具体如何实现以及为什么这样实现，我们将会在这一小节详细介绍。3.2.5 小节将简单的介绍一下我们提出的模型的训练与优化过程。

3.2.1 符号定义

我们首先这样定义我们研究的问题：使用 $X\in R^{mx}$表示原始的像素空间，$Y\in R^{my}$表示输出空间，$Z\in R^{mz}$表示表征空间，其中 mx、my、mz 分别是 X、Y、Z 三个空间的维度。使用 $f: X\rightarrow Z$ 表示模型的特征提取器完成的特征提取功能，这在我们的模型中主要由 Backbone 以及 Neck 结构完成；同时使用 $g: Y\rightarrow Z$ 表示模型的分类器完成的类别预测功能，这在我们的模型中主要由 Head 结构中的类别预测部分完成。这里仅仅考虑由类别预测部分是因为：定位回归过程与特征之间的相关性比较弱，如果这里对位置预测也使用样本加权策略将会大大增加工作量，而成效甚微，所以我们的基于因果推断的稳定学习方法仅仅用在分类预测上，而不考虑位置预测。

假设我们有 n 个图像样本 $X\in R^{n*mx}$, 它们所对应的类别标签为 $Y\in R^{n'*my}$，注意这里是 n'而不是 n，是因为一张图像中不一定仅仅包含单个感兴趣目标，也就

是说一张图像不一定仅仅对应一个类别标签。同时我们用X_i表示第 i 个图像样本，用y_{ij}表示第 i 个图像样本中第 j 个感兴趣目标的类别标签。另外，对于第 i 个样本，其通过深度神经网络的特征提取部分最终得到的表征用$Z_{:,i}$表示。使用$\omega \in R^n$表示学习到的样本权重，μ和 v 是随机傅里叶特征映射函数。

3.2.2 基于随机傅里叶特征的样本加权去相关

（一）独立测试统计

为了消除表征空间中任意两组特征$Z_{:,i}$和$Z_{:,j}$之间的相关性，使任意两组特征之间相互独立，我们需要在学习样本权重的过程中使用假设测试统计（Hypothesis Testing Statistics）来实时地检验当下任意两两变量间的独立性。受[32]的启发，我们的研究使用基于核的条件独立测试（Kernel-based Conditional Independence Test，KCIT）来完成假设测试统计。

具体的来说，假设用变量 A 和 B 分别简记$Z_{:,i}$和$Z_{:,j}$，（A1, A2, . . . An）和(B1, B2, . . . Bn)分别是 A 和 B 这两个变量上的一组样本集合，同时假设变量 A 满足的分布对应的一个正定的核为 k_A，这个正定的核对应的再生希伯特空间（Reproducing Kernel Hilbert Space, RKHS）为 H_A。类似的，在变量 B 上的正定核和对应的再生希伯特空间分别用 k_B 和 H_B 表示，则从 H_B 到 H_A 的交叉协方差算子按公式(3-1)进行计算：

$$< h_A, \sum_{AB} h_B >= E_{AB}[h_A(A)h_B(B)] - E_A[h_A(A)]E_B[h_B(B)] \tag{3-1}$$

上式中的 h_A和h_B分别表示 H_A空间和H_B空间中的任意函数，即满足 $h_A \in H_A$ 且 $h_B \in H_B$。[33]中提出，有了任意两个变量之间的交叉协方差算子，它们之间的独立性可以根据定理 3.1 来衡量：

定理 3.1：如果 k_A 与 k_B 的内积是独特（characteristic）的，且 $E[k_A(A,A)] < \infty$，$E[k_B(B,B)] < \infty$，那么有：

$$\sum_{AB}=0 \Leftrightarrow A \perp B \tag{3-2}$$

[34]中提出了希伯特-施密特独立判据（Hilbert-Schmidt Independence Criterion，HSIC），该判据可以通过判断交叉协方差算子Σ_{AB}是否为 0 或者近似为 0 来检验特征间独立性。然而，HSIC 的计算开销会随着训练数据的 batch-size 的增加而显著的增加，针对这个问题，[35]提出了随机化的条件相关性测试（Randomized conditional Correlation Test，RCoT）和随机化的条件独立测试（Randomized conditional Independence Test，RCIT），这两种方法是对 HSIC 方法的近似，可以在大大降低时间开销的同时保证独立性测试的准确性与 HSIC 相当。我们的独立测试没有先验条件的约束，可以看做是条件独立测试中条件为空

的特殊情况，因此也可以使用[35]中的相关结论和方法。具体来说，[35]中证明了：1、可以使用某些特殊空间中的函数近似连续平移不变核，并在此基础上使用近似的交叉协方差算子来进行后续运算与判断；2、Frobenius 范数对应于欧氏空间（Euclidean space）的 Hilbert-Schmidt 范数，因此条件独立测试统计可以基于 Frobenius 范数。受这两条结论的启发，我们的研究按照公式(3-3)计算 A 和 B 这两个变量之间的近似部分交叉协方差矩阵$\sum_{AB}^{\wedge}$：

$$\sum_{AB}^{\wedge} = \frac{1}{\mathrm{n}-1}\sum_{i=1}^{n}\left[\left(\mathrm{u}(A_i)-\frac{1}{n}\sum_{j=1}^{n}\mathrm{u}(B_j)\right)^{T}\bullet\left(v(B_i)-\frac{1}{n}\sum_{j=1}^{n}v(B_j)\right)\right] \tag{3-3}$$

上式中，

$$\begin{aligned}&\mathrm{u}(A)=(\mu_1(A),\mu_2(A),\ldots\mu_{nA}(A)),\mu_j(A)\in \mathrm{H}_{RFF},\forall j,\\&v(A)=(v_1(B),v_2(B),\ldots v_{nB}(B)),v_j(B)\in \mathrm{H}_{RFF},\forall j.\end{aligned} \tag{3-4}$$

这里的 RFF（Randomized Fourier Features）表示在原始图像空间基础上提取的随机傅里叶特征。我们将分别从 H_{RFF}中采样n_A和$\mathrm{n_B}$个函数，$\mathrm{H_{RFF}}$表示我们自定义的一个随机傅里叶特征函数空间，其形式如下：

$$\mathrm{H}_{RFF}=\left\{\mathrm{h}:x\to\sqrt{2}\cos(\omega x+\phi)\,|\,\omega\sim N(0,1),\phi\sim U\mathrm{niform}(0,2\pi)\right\} \tag{3-5}$$

这里之所以采用公式(3-5)的定义方式，是因为在[35]中证明了满足这种映射的函数对应的空间可以近似连续平移不变核。到这里，独立性测试统计量 I_{AB} 就可以定义为近似部分交叉协方差矩阵的 Frobenius 范数，如公式(3-6)所示:

$$I_{AB}=\left\|\hat{\sum}AB\right\|_F^2 \tag{3-6}$$

这里值得注意的是，I_{AB} 总是非负的。当 I_{AB} 逐渐趋于零时，两个变量 A 和 B 逐渐趋于独立。因此，I_{AB} 可以有效地测量任意变量之间的独立性。

一般而言，独立性测试的准确性随着样本数增加而增加，也即随着$\mathrm{n_A}$和$\mathrm{n_B}$的增加而增加，但是一味的增加样本数将会大大增加计算的复杂度，因此需要选取适中的$\mathrm{n_A}$和$\mathrm{n_B}$值，使用尽可能少的样本得到尽可能高的独立测试准确率，受[35]的启发，将$\mathrm{n_A}$和$\mathrm{n_B}$都设置为 5，就足以判断随机变量的独立性。

（二）学习样本权重去除相关性

受[36]的启发，我们在研究中使用样本加权来消除表征空间中不同特征间的相关性，并且通过 3.2.1 节中（一）这部分介绍的方法来衡量独立性。具体来说，我们使用了一个维度为(n,1)的向量ω来存储样本权重，其中 n 为样本数量。在初始化的时候，先将ω的每一个元素都用 1 进行初始化，即每一个样本的权重都为 1，然后使用一个 softmax 处理使得ω所有元素的总和为 1，这样做是因为我们在

实际操作中，学习到的样本权重仅仅在分类损失上使用，如果不进行 softmax 处理，会加大分类损失在总损失中的占比。在学习到样本权重后，加权后的近似部分交叉协方差矩阵按公式(3-7)计算：

$$\hat{\sum} AB;\ \omega=\frac{1}{n-1}\sum_{i=1}^{n}\left[\left(\omega_i\bullet u(A_i)-\frac{1}{n}\sum_{j=1}^{n}\omega_j\bullet u(A_j)\right)^T\left(\omega_i\bullet v(B_i)-\frac{1}{n}\sum_{j=1}^{n}\omega_j\bullet v(B_j)\right)\right] \tag{3-7}$$

上式中μ和 v 是公式(3-4)中阐述的 RFF 映射函数。在我们的研究中，学习样本权重去除相关性的目标是去除任意一对特征之间的相关性，使所有特征两两相互独立，因此，我们在训练的过程中按照优化公式(3-8)学习样本权重：

$$\begin{aligned}\omega^{*}&=\arg\min_{\omega\in\Delta_n}\sum_{1<i<j<mZ}\left\|\hat{\sum}\left((Z:,i)\bullet(Z:,j)\right);\omega\right\|_F^2,\\ \Delta_n&=\left\{\omega\in R_{+}^{n}\mid\sum_{i=1}^{n}\omega_i=1\right\}\end{aligned} \tag{3-8}$$

总的来说，在训练阶段，我们的算法迭代性地优化样本权重ω、特征提取函数 f 和类别预测函数 g，整个迭代过程如公式(3-9)所示：

$$\begin{aligned}f^{(t+1)},g^{(t+1)}&=\arg\min_{f,g}\sum_{i=1}^{n}\omega_i^{(t)}loss_{BCE}\left(g\left(f\left(X_i\right)\right),y_i\right),\\ \omega^{(t+1)}&=\arg\min_{\omega\in\Delta_n}\sum_{1\le i<j\le m_Z}\left\|\hat{\sum}Z_{:,i}^{(t+1)}Z_{:,j}^{(t+1)};\omega\right\|_F^2\end{aligned} \tag{3-9}$$

其中$Z^{(t+1)}=f^{(t+1)}(X)$,t 表示时间戳，$loss_{BCE}(\cdot;\cdot)$表示二进制交叉熵逻辑回归损失函数（[37]，BCE with logits loss，Binary Cross Entropy With LogitsLoss）。

3.2.3 全局性地学习样本权重

上面的公式(3-9)需要的ω是我们训练过程中所有样本的权重，但是在真实的深度学习任务的训练场景下，总样本数(这里我们用 n_sample 表示)往往庞大，如果一次性地把所有样本输入到我们的神经网络之中，将需要巨大的存储和计算开销，因此我们通常采用分批次训练的方法来训练我们的深度模型：在完整的训练过程中，我们需要训练多次来不断地优化我们的模型参数，一次训练会用到所有训练样本，我们把这样的一次训练称为一个 epoch，假设我们的模型总共训练 epoch_n 次，在每一个 epoch 中，我们都会用到分批次训练的方法，现在假设我们的一个小批次的样本数为 batch-size，那么完成一个 epoch 的训练需要迭代 batch_n 次(这里应满足 batch_n =⌈n_sample/batch-size⌉)。这样的话，我们的训练在一次迭代时学习到的样本权重仅仅是这一个批次的样本的权重，而不是所有样

本的权重；同时，我们的特征提取器提取到的最终特征也只是这一个批次的样本的特征，而不是所有样本的特征。为了解决上述问题，我们需要一个策略，在保留分批次训练方法的前提下，做到全局性地学习所有样本的权重。

针对这一个问题，我们提出一种“保存与重加载”策略。受[38]的启发，我们的模型使用三个预测层来检测不同尺度的物体。这样，具体的来说：

首先，我们给模型的每个预测层均添加了两个属性：model. pre_features 和 model. pre_weight，并设置两个全局变量 pre_features 和 pre_weight。其中，model. pre_features 初始化后的维度为（batch-size，feature_dim），pre_features 用 model. pre_features 初始化，用于保存当前迭代之前的所有迭代对应的样本特征信息，我们称之为“从前局部样本特征信息”。这里的 feature_dim 表示每个预测层进行预测之前特征提取器最终提取到的、对应于一张图像的展平特征的维度。由于这个维度较高，在实际操作中，我们在展平特征之前使用平均池化（Average Pooling）（[37]）进行了特征降维，在我们的模型中，三个预测层对应的降维后的 feature_dim 分别 128，256，512。三个预测层对应的 model. pre_weight 初始化后的维度为（batch-size，1），pre_ weight 用 model. pre_weight 初始化，用于保存当前迭代之前的所有迭代对应的样本权重信息，我们称之为“从前局部样本权重信息”。在第一个 epoch 的第一次迭代训练之前，model. pre_features 用全 0 进行初始化，这是因为在开始第一次迭代训练之前我们没有提取过任何样本的特征信息；model. pre_weight 用全 1 进行初始化，这是因为在开始学习样本权重之前，我们有理由认为最初的样本权重都应该是相等的。值得注意的是，由于我们的工作在进行实验时是基于 Pytorch 实现的，在这里 model. pre_features 和 model. pre_weight 均使用 Pytorch 的 register_buffer( ) 函数进行定义和初始化，这样做是为了保证这两个模型属性值在训练与优化的过程中不会自动更新。

其次，我们在每一个批次的训练过程（即每一次迭代过程）中给每一个预测层均添加了两个局部变量 weight 和 cfeatures。其中，cfeatures 保存当前迭代对应的这个批次的样本特征信息，我们称之为“当前局部样本特征信息”，它的维度与对应的 pre_features 是相同的，这也正是保证后面的张量拼接操作（concat）的前提。weight 用于保存当前迭代对应的样本权重信息，我们称之为“当前局部样本权重信息”，它的维度以及初始化的方式与对应的 pre_weight 都是相同的，这里要注意的是，weight 在每一次迭代过程中都要初始化，而 pre_weight 仅仅在第一个 epoch 的第一次迭代训练前初始化。

接着，我们采用 concat 操作将 pre_features 与 cfeatures 进行拼接并保存在局部变量 all_feature 中，将 pre_weight 与 weight 进行拼接并保存在局部变量

all_weight 中，all_feature 与 all_weight 分别表示“全局样本特征信息”与“全局样本权重信息”。但这里值得注意的是，即使我们称之为“全局”，这两个变量保存的仍然不是所有样本的相关信息，而是包括当下这个 epoch 下当前批次及其之前的所有批次的样本的相关信息。接下来，我们使用公式(3-8)来更新 weight。

然后，我们在一个 epoch 的每一次迭代将要结束之前，将 pre_features 与 pre_weight 分别按照公式(3-10)进行更新：

$$\begin{aligned} Z'_{\text{before}\_local} &= \partial_i * Z_{\text{before}\_local} + (1-\partial_i) * Z_L, \\ w'_{\text{before}\_local} &= \partial_i * w_{\text{before}\_local} + (1-\partial_i) * w_L. \end{aligned} \tag{3-10}$$

上式中，Z'_{before_local}与Z_{before_local}分别代表更新后与更新前的“从前局部样本特征信息”（pre_features），ω'_{before_local}与ω_{before_local}分别代表更新后与更新前的“从前局部样本权重信息”（pre_weight），Z_L与ω_L分别代表“当前局部样本特征信息”（cfeatures）与“当前局部样本权重信息”（更新后的 weight），α_i是一个平滑超参数，在具体设置时，我们使用 batch_n - 1 个不同的平滑参数 α_i，在全局信息中同时考虑长期记忆（α_i大）和短期记忆（α_i小）。注意这里是 batch_n - 1 而不是 batch_n，是因为最后一个批次的样本数可能小于 batch-size，在进行拼接与更新的过程中会出现变量维度不一致而导致的计算错误，考虑到这一点，一个简单的做法是在每一个 epoch 中进行最后一次迭代训练时不进行样本权重的学习。这样虽然会对特征去相关产生一定的影响，但由于最后一个批次的样本数只占总样本数的极小一部分，这种操作带来的影响基本可以忽略不计。

最后，在更新了 pre_features 与 pre_weight 之后，我们将 model.pre_features 和 model.pre_weight 这两个属性值以 copy 的方式同步更新，这样做保证了模型在一次训练完成后接着进行下一次训练时，不用再用全 0 值和全 1 值分别初始化这两个变量，而是用上一次训练的结果进行初始化，这样可以加快模型收敛的速度。

3.2.4 损失函数

目标检测主要有三个目的：(1)检测出图像中目标的位置；(2)检测出目标的大小，通常为恰好包围目标的矩形框；(3)对检测到的目标进行识别分类。针对这三个目的，我们的模型的损失函数在设计的时候也由三大部分组成：边界框损失、置信度损失和分类损失。

我们的模型沿用单阶段模型“使用固定网格上的检测器”的主要思想。具体来说，受 YOLO 系列模型的启发，我们的模型将原始图像进行 resize 后得到尺寸为 640 * 640 的图像作为输入，然后将输入图像划分为 N*N 的网格，网格中的每个单元格称为 grid。我们的模型选择了 80*80、40*40、20*20 这三种尺寸的网格

划分，分别对应着的三个预测层（predict layers），每一个 grid 对应着三个不同形状的锚（anchor），然后我们对每个 grid 的每个 anchor 都预测三个指标：矩形框、置信度和分类概率。其中，矩形框表征目标的大小以及精确位置；置信度表征预测框（Predicted Box）的可信程度，取值范围为 0-1，值越大说明 Predicted Box 中越可能存在目标；分类概率表征目标的类别。我们最终得到的损失函数为这三种损失的加权和。

（一）mask 掩码矩阵

在计算这三个损失之前，我们首先要对上面提到的 grid 进行 mask 掩码处理，这是因为大多数 grid 的大多数 anchor 并不对应真实目标，引入 mask 掩码处理可以大大减少模型的计算开销。实际上，并非所有的 Predicted Box 都需要计算所有这三种类别的损失函数值，而是根据 mask 矩阵来决定：(1)仅 mask 矩阵中对应位置为 true 的预测框需要计算矩形框损失和分类损失；(2)所有预测框都需要计算置信度损失，但是 mask 为 true 的预测框与 mask 为 false 的预测框的置信度标签值不一样。

具体来说，在初始化 mask 矩阵时，我们将其设置为值全部为 false 的 bool 类型的矩阵。假设我们从输入的数据集获取的第 i 个图像样本中的一个目标对应的标签是一个五元组（c, x, y, w, h），其中 c 表示类别，x 与 y 表示真实框（Ground Truth Box）的中心点横纵坐标，w 与 h 表示 Ground Truth Box 的宽和高，假设第 i 个图像样本 resize 之前的宽与高分别为w_i和h_i，首先要利用公式(3-11)进行坐标转换，得到的x'，y'，w_{gt}，h_{gt}分别是 Ground Truth Box 对应于 resize 之后的图像的中心点横纵坐标和宽高值：

$$\begin{aligned} x' &= \frac{x}{w_i} * 640, 0 \le x' < 640, \\ y' &= \frac{y}{h_i} * 640, 0 \le y' < 640, \\ w_{gt} &= \frac{w}{w_i} * 640, 0 \le w_{gt} < 640, \\ h_{gt} &= \frac{h}{h_i} * 640, 0 \le h_{gt} < 640, \end{aligned} \tag{3-11}$$

然后根据公式(3-12)计算出 Ground Truth Box 在 N*N 的 grid 中的网格坐标(x_g, y_g)，注意这里的x_g与y_g均为浮点数：

$$x_g = \frac{x'}{(640/N)}, 0 \le x_g < N,$$
$$y_g = \frac{y'}{(640/N)}, 0 \le y_g < N. \tag{3-12}$$

接着对x_g与y_g向下取整，得到整型网格坐标$(\mathrm{x}_0, \mathrm{y}_0)$。同时，为了加快训练的收敛速度，我们在研究中使用一定的方法增加正样本的数量：对网格$(\mathrm{x}_0, \mathrm{y}_0)$的左右、上下再各取一个邻近的网格$(\mathrm{x}_1, \mathrm{y}_0)$和$(\mathrm{x}_0, \mathrm{y}_1)$。这里具体的做法为：1）如果点$(x_g, y_g)$在格子的左上角，则取左边、上方的两个格子；2）如果点(x_g, y_g)在格子的右上角，则取右边、上方的两个格子；3）如果点(x_g, y_g)在格子的左下角，则取左边、下方的两个格子；4）如果点(x_g, y_g)在格子的右下角，则取右边、上下方的两个格子。根据以上原则，我们将得到三个互相邻近的网格$(\mathrm{x}_0, \mathrm{y}_0)$、$(\mathrm{x}_1, \mathrm{y}_0)$和$(\mathrm{x}_0, \mathrm{y}_1)$，这时我们认为 Ground Truth Box 位于这三个格子附近。

前面已经提到过，我们的模型有三个 predict layers，这三个 predict layers 对应着不同的网格划分，分别为 80*80、40*40、20*20 这三种尺寸，对于每种尺寸的网格，我们给它们分配 3 个预置的 anchor, 具体做法为：1）宽、高最小的 anchor0、anchor1、anchor2 分配给 80*80 网格的每个格子；2）宽、高次小的 anchor3、anchor4、anchor5 分配给 40*40 网格的每个格子；3）宽、高最大的 anchor6、anchor7、anchor8 分配给 20*20 网格的每个格子。按照这种规则，以 80*80 的网格为例，$(\mathrm{x}_0, \mathrm{y}_0)$、$(\mathrm{x}_1, \mathrm{y}_0)$和$(\mathrm{x}_0, \mathrm{y}_1)$这三个相邻的格子每一个都对应 anchor0、anchor1、anchor2 这三个 anchor 框，假设 anchor0、anchor1、anchor2 这三个 anchor 框的宽高分别为(w_0, h_0)、(w_1, h_1)、(w_2, h_2)，在公式(3-11)已经求得真实目标框在 resize 之后的图像中的宽高为(w_{gt}, h_{gt})，有了这些变量，下一步将按照公式(3-13)分别计算(w_{gt}, h_{gt})与(w_0, h_0)、(w_1, h_1)、(w_2, h_2)的比例，再根据这些比例剔除不满足要求的 anchor 框：

$$\mathrm{t}_{\mathrm{wi}} = \max\left(\frac{w_{gt}}{wi}, \frac{wi}{w_{gt}}\right),$$
$$\mathrm{t}_{\mathrm{hi}} = \max\left(\frac{h_{gt}}{hi}, \frac{hi}{h_{gt}}\right), \tag{3-13}$$
$$i = 0,1,2.$$

有了上面计算的比例，具体的剔除规则为：如果$\mathrm{t}_{\mathrm{wi}} < threshold$且$\mathrm{t}_{\mathrm{hi}} < threshold$，则保留 anchor i，否则剔除 anchor i。我们的研究之中将$threshold$设置为 4，这个阈值用来限制的是 Ground Truth Box 与 anchor 之间的宽高比。最后，我们将保留的 anchor 框标记为 1(true)，剔除的 anchor 框标记为 0(false)，且把 anchor0、anchor1、anchor2 对应的标记记为(m0, m1, m2)，那么根据 80*80 网格中$(\mathrm{x}_0, \mathrm{y}_0)$、$(\mathrm{x}_1, \mathrm{y}_0)$和$(\mathrm{x}_0, \mathrm{y}_1)$这三个坐标位置，我们分别对 mask 矩阵按照公式(3-14)赋值，最后对

一张图像中的所有 Ground Truth Box 都作以上判断，并对 mask 矩阵赋值，即可得到该图像的 mask 矩阵。

$$\begin{aligned} &\text{mask}(i, x_0, y_0) = m_i, \\ &mask(i, x_1, y_0) = m_i, \\ &mask(i, x_0, y_1) = m_i, \\ &i = 0,1,2. \end{aligned} \tag{3-14}$$

（二）矩形框损失

我们的模型使用 CIOU（[39]）来计算矩形框损失。假设用符号loss_{box}表示矩形框损失，则其具体计算如公式(3-15)所示:

$$\begin{aligned} v &= \frac{4}{\pi^2}\left(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\frac{w_p}{h_p}\right)^2 \\ \partial &= \frac{v}{1 - IOU + v} \\ \text{loss}_{\text{box}} &= IOU - \frac{\rho^2}{c^2} - \partial v \end{aligned} \tag{3-15}$$

上式中，w_{gt}与h_{gt}表示真实框的宽和高，w_p与h_p表示预测框的宽和高，ρ为真实框与预测框的中心点距离，c 为真实框与目标框的最小包围矩形的对角线长度，v为真实框与目标框的宽高比相似度，α为ρ的影响因子，IOU 表示真实框与目标框的交并比。

（三）置信度损失

对于一张分割成 N*N 尺寸网格的图像,我们的模型对其中每个 grid 都预测三个位于该 grid 附近的 Predicted Box，每个 Predicted Box 的预测信息包括框的中心点坐标、宽高、置信度、分类概率，因此模型总共输出 3*N*N 个值为 0 到 1 之间的预测置信度，它们与 3*N*N 个 Predicted Box 一一对应。每个预测框的置信度表征这个 Predicted Box 含有目标物体的可信程度，值越大表示该 Predicted Box 越可信，也即越接近真实目标的最小包围框。在这里我们用到了 3.2.3 小节第（一）部分讲到的 mask 掩码矩阵：以维度同样为 3*N*N 的 mask 矩阵为标记，对置信度标签矩阵进行赋值。在这里我们使用了一个小技巧：对 mask 为 true 的位置不直接赋 1，而是计算该位置处 Predicted Box 与 Ground Truth Box 的 CIOU，使用 CIOU 作为该预测框的置信度标签，对 mask 为 false 的位置还是直接赋 0。这样一来，标签值的大小与 Predicted Box 和 Ground Truth Box 的重合度有关，两框重合度越高则标签值越大。同时，由于 CIOU 的取值范围是-1.5 至 1，而置信度标签的取值范围是 0 至 1，所以需要对 CIOU 做一个截断处理：当 CIOU 小于 0 时直接取 0 值作为标签。

我们的模型在进行前向传播时会得到预测置信度矩阵，这里我们记为P_{obj}，同时我们按照上面介绍的规则可以得到置信度标签矩阵L_{obj}。有了预测置信度矩阵P_{obj}和置信度标签矩阵L_{obj}，可以按照公式(3-16)计算 N*N 的网格的置信度损失值：

$$loss_{BCE}(i,x,y) = -L_{obj}(i,x,y) * \log P_{obj}(i,x,y) - (1 - L_{obj}(i,x,y)) * \log(1 - P_{obj}(i,x,y)),$$

$$loss_{\text{obj}} = \frac{1}{mm(mask = true)} \sum_{\substack{mask=true \\ 0\le i<3 \\ 0\le x<N \\ 0\le y<N}} loss_{BCE}(i,x,y),$$

$$loss_{\text{no\_obj}} = \frac{1}{mm(mask = false)} \sum_{\substack{mask=false \\ 0\le i<3 \\ 0\le x<N \\ 0\le y<N}} loss_{BCE}(i,x,y), \tag{3-16}$$

$$loss_{obj\_N} = a * loss_{obj} + (1-a) * loss_{no\_obj}.$$

其中$loss_{BCE}$为 BCE with logits loss，$loss_{obj}$表示 mask 为 true 的 grid 对应的损失之和，我们称之为正样本损失和；$loss_{no_obj}$表示 mask 为 false 的 grid 对应的损失之和，我们称之为负样本损失和；a 是一个平衡因子，最后得到的总的置信度损失为正样本损失与负样本损失的加权和。

（四）分类损失

上面已经提到，我们的模型对三种不同尺寸的 N*N 的网格的每个 grid 都预测三个 Predicted Box。每个 Predicted Box 的预测信息都包含了 c 个分类概率，其中 c 为总类别数，对于一个有 c 个类别的数据集来说，我们的模型的每个 Predicted Box 在分类任务上有 c 个 0 到 1 之间的分类概率，因此我们的模型的一种尺寸的网格总共预测 3*N*N*c 个分类概率，它们共同组成类别预测概率矩阵。N*N 的网格的类别标签概率矩阵与类别预测概率矩阵的维度相同，这里值得注意的是输入的类别标签是一个范围在 0 到 c - 1 之间的整数，因此这里要用到独热编码。同时，为了减少过拟合，增加训练的稳定性，我们还对独热码标签按照公式(3-17)做了平滑操作:

$$label_{smooth} = label * (1-\lambda) + \lambda / c \tag{3-17}$$

其中 *label* 为原始的类别标签独热编码，*label* 中的所有数值均为 0 或 1，λ为平滑系数，c 为上面说到的总类别数，得到的 $label_{smooth}$ 中的数值不仅仅为 0 或 1。假设由 $label_{smooth}$ 组成的类别标签概率矩阵为L_{smooth}，类别预测概率矩阵为P_{cls}，同样使用 BCE with logits loss 作为分类损失的原则，我们可以按照公式(3-18)得到了分类损失矩阵中的每个数值：

$$\begin{aligned}&\text{loss}_{\text{BCE}}(i,x,y,t)=-L_{smooth}(i,x,y,t)*\log P_{cls}(i,x,y,t)-(1-L_{smooth}(i,x,y,t))*\log(1-P_{cls}(i,x,y,t)),\\&0\le i<3,\\&0\le x<N,\\&0\le y<N,\\&0\le t<c.\end{aligned}\tag{3-18}$$

由于我们的样本加权是用在分类损失上的，而每一张图像对应的真实目标数往往是不同的，我们还需要通过一定的操作得到具体每一张图像对应的所有目标的分类损失。具体来说，我们在计算每个 Ground Truth Box 与 Predicted Box 的 BCE with logits loss 时，将 torch. nn. BCEWithLogitsLoss( )函数的 reduction 参数设置为‘none’，这样做是为了保证计算分类损失时不一次性地返回所有目标分类损失的平均值，而是以矩阵的形式返回，这个矩阵的维度为(n_target,*c*)，其中 n_target 为一个 batch 的所有图像包含的所有正样本的数量，*c* 为上文提到的总类别数量，我们还将这个矩阵按列求和，再除上 *c*，这样，得到的新矩阵中每个横向量代表一个目标的平均分类损失，我们称之为“单目标平均分类损失”。然后，我们以图像样本的 id 为索引，将属于同一个 id 的单目标分类损失求和，得到每张图像所有目标的分类损失和，然后按照 id 升序进行排序，返回一个维度为(batch_size,1)的矩阵，这个矩阵的每个分量就代表了一个 batch 中具体每一张图像的所有目标的分类损失，我们把它称之为“图像分类损失矩阵”

最后，我们将前面介绍的学习到的样本权重向量的每个分量与这里的图像分类损失矩阵的每个分量相乘，就得到了一个形状为(batch_size,1)的“加权图像分类损失矩阵”，这样做就通过样本加权去除了特征间的相关性。最后，把加权图像分类损失矩阵的所有分量相加再除以 batch_size 就可以得到一个 batch 的样本的平均分类损失。

3.2.5 模型训练与优化

因为我们的模型是受 YOLO 系列以及 StableNet（[40]）的共同启发而设计的，我们把它命名为 Stable_Yolo。在训练阶段，如 3.2.1 和 3.2.2 两小节所介绍的，Stable_Yolo 先利用之前迭代所保存的与特征间相关性有关的全局知识，学习每个 batch 的样本的权重，然后进行前向传播，利用公式(3-8)和 3.2.3 小节介绍的损失函数对模型的参数和样本权值进行了迭代优化。Stable_Yolo 的训练流程大致如算法 1 中的伪代码所示：

算法 1：Stable_Yolo 的训练流程

输入：epoch_n, balancing_epoch_n

输出：学习到的模型

1．**for** epoch **in range** (epoch_n) **do**:

2. **for** batch **in range** (batch_n) **do**:
3. 前向传播；
4. 根据公式(2-10)和 concat 操作重加载全局特征；
5. **for** balancing_epoch **in range** (balancing_epoch_n) **do**:
6. 根据公式(2-8)优化样本权重;
7. **end for**;
8. 根据 2.2.3 节介绍的损失函数进行反向传播；
9. **end for**;
10. **end for**.

在算法 1 中，epoch_n 表示模型总的训练次数，balancing_epoch_n 表示模型在每个批次的训练中优化样本权重的迭代次数。

在推理测试阶段，由于不存在反向传播，Stable_Yolo 跳过了样本加权阶段，直接使用训练好的固定的模型参数进行预测。

3.3 模型设计

如图 3-2 所示，Stable_Yolo 主要分为 Backbone、Neck、Chin、Head 四大模块。与 YOLO 系列的模型不同的是，我们的模型引入了 Chin 模块，并对 Head 模块进行必要的更改。

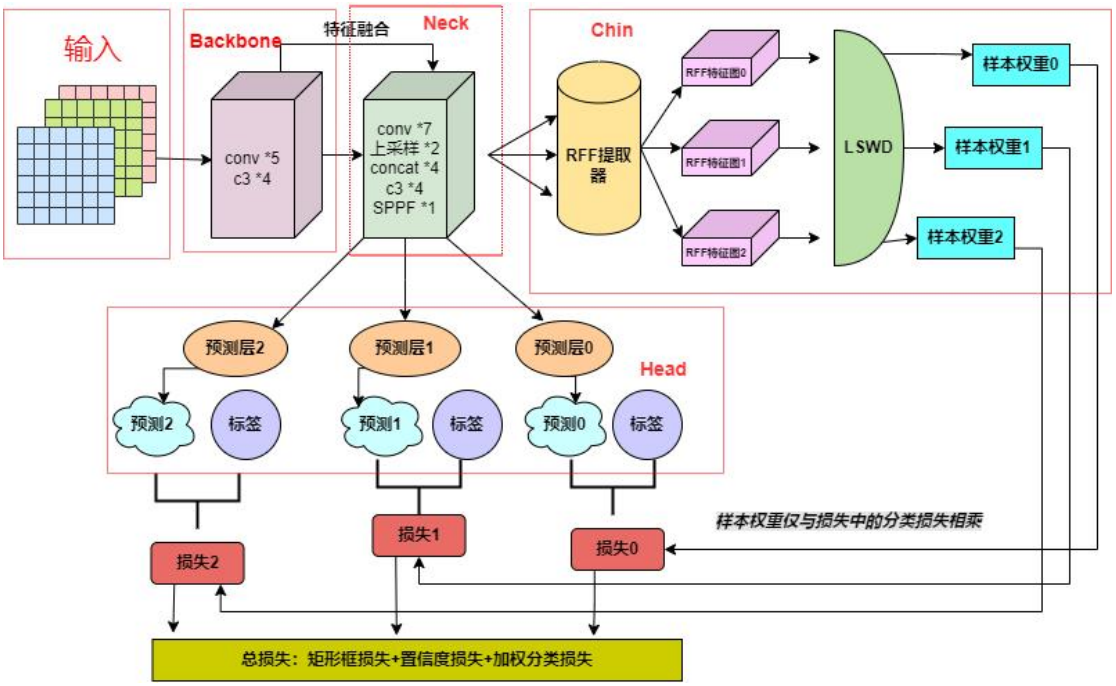

图 3-2　Stable_Yolo 模型结构

Chin 这个模块主要用于全局性地学习样本权重去除特征间的相关性。具体来说，Chin 模块分别接收来自 Neck 部分的三个特征图，这三个特征图分别对应于三种尺寸的网格，然后按照公式(3-5)分别提取这三个特征图的 RFF 特征，这由图 3-2 中的 RFF 提取器完成，由此可以得到三张 RFF 特征图，接着将它们送到图 3-2 中的 LSWD 结构中，这个结构的全称为 Learning Sample Weight for Decorrelation，主要作用是按照 3.2.1 和 3.2.2 两小节介绍的方法学习样本权重以去除任意两组特征间的相关性，这样，我们的模型在三张对应着不同 predicted layers 的 RFF 特征图上就分别学到了一组样本权重，这三组样本权重是相互独立、互不干扰的。

紧接着，在 Head 模块，每个预测层的损失的计算是独立进行的，predicted layers 损失计算的方法按照 3.2.3 小节介绍的进行。特别地，对于一个预测层，其未加权的分类损失要与 Chin 模块学到的对应的样本权重相乘，得到加权分类损失。最后，我们把每个预测层的三种损失按照重要性权重进行加权求和，得到每个预测层的加权总损失，并计算出三个预测层的加权总损失的和，得到模型的总损失，最后使用模型的总损失进行反向传播来优化模型参数，达到模型训练的目的。

Backbone 与 Head 两个模块基本沿用了 Yolov5 模型结构中的相关模块。其中，Backbone 模块的主要作用为提取特征、构建特征金字塔和加速计算，其主要采用了 CSP (Cross-Stage Partial Network) （[41]）Darknet53 和 Conv 作为基础架构，CSP Darknet53 对应于图 3-2 中的“C3”，其中的“C3*4”表示 Backbone

部分一共使用到 4 个 C3 架构，但注意的是这几个 C3 架构是与 Conv 交错排列的，后面的“*n”也都表示类似的含义。Conv 的具体结构如图 3-3 所示，其中的 Conv2d 表示 2D 卷积层，BN 表示批归一化层（Batch Normalization Layer），SiLU 表示激活函数；C3 中用到 Bottleneck，Bottleneck 的具体结构如图 3-4 所示，它起到加深网络的作用。C3 的具体结构如图 3-5 所示。

图 3-3　Conv 层的具体结构

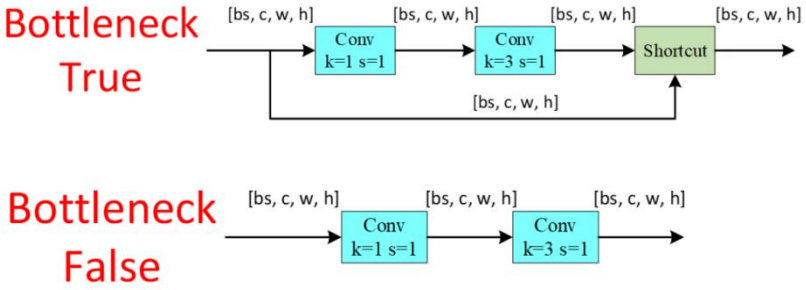

图 3-4　Bottleneck 层的具体结构

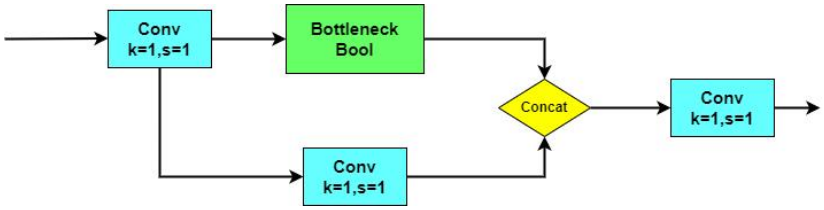

图 3-5　C3 的具体结构

Neck 部分的主要作用为特征提取与融合，其主要采用了 SPPF (Spatial Pyramid Pooling-Fast)、Conv 和 C3 作为基础架构。原始的 SPP 层[43]可以解决单尺度输入图像常出现的物体大小不一、位置不固定等问题，它会通过分别池化多个尺度的特征图，使网络能够处理多种尺度的物体，并且降低了计算量。与原始的 SPP 层不同，SPPF 层采用了更加高效的实现方式，从而可以提高运行速度并减少内存占用,其具体结构如图 3-6 所示，其中的 Max Pooling 表示最大池化层。

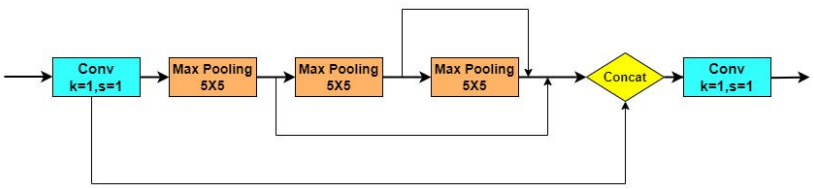

图 3-6　SPPF 的具体结构

第四章　实验与分析

4.1　实验设置

为了评估模型的稳定性，我们同时使用了一般数据集、带有域偏移性质的公开数据集以及自己标注的具有域偏移性质的数据集，在我们的模型和 yolov5 模型上分别进行训练和测试。

对于不同的数据集上的超参数的设置，我们在 coco128 数据集上将最大训练次数 epoch 设置为 300 次，其它几个数据集的 epoch 设置为 100 次，我们发现这样设置能使模型在训练时损失基本收敛；所有数据集上 batch-size 均设置为 16；学习样本权重时的 epoch_balancing 设置为 30 次；预训练的模型均使用 yolov5s，它在 yolov5 系列中是轻量最级的一个网络，甚至可以部署到安卓系统的移动端，速度最快但精度最低；初始的学习率都设置为 0.01，且使用 SGD 优化器；置信度损失、目标框损失、分类损失三种损失的权重比例设置为 1：0.05：0.5，这是 yolov5 的创作团队经过大量实验验证得到的一个经验比值，因此我们直接沿用。

除了在不同数据集上与 baseline 的对比试验，我们还通过改变相关超参数进行了两个简单的消融实验，以此来进一步验证我们提出的方法的有效性：

1、Stable_Yolo 依赖于从高斯分布中采样的随机傅里叶特征来平衡训练数据，因此，理论上讲，采样的特征越多，最终的表征就越独立。然而，在实践中，生成更多的特征需要更多的计算成本。在第一种消融研究中，我们验证了随机傅里叶特征采样数量大小的影响。受[44]的启发，我们调整采样中随机傅里叶特征的维度来完成这项消融对比。

2、在第 3.2.2 小节中我们提出了“预保存特征及权重”的策略，为了探究这一策略对模型性能的影响，我们通过调节 presave_ratio 这一超参数来调整预保存特征及权重所占比例，从而完成这项消融实验。

4.2　数据集

coco128：coco128 数据集是一个仅包含 128 张真实世界图像样本的目标检测数据集。虽然图像数量有限，整个数据集大小也仅仅为 7MB，但是该数据集包含 80 个常见的类别，涵盖了各种动物和交通工具、日常生活用品等。由于 coco128 数据集的训练集和验证集以及测试集选用的是同一批图像样本（即这里的 128 张图像样本），因此可以认为这个数据集不存在域偏移情况，我们把这种情况称为“一般数据集”。

Cityspaces / Foggy Cityscapes：Cityspaces 数据集是一个语义分割数据集，包括 2975 张训练图片、500 张验证图和 1525 张测试图，每张图片大小都是 1024*2048，都有像素级的标注，都是正常天气下不同城市的市区场景，目标物体主要是行人、车辆等。Foggy Cityspaces 是在 Cityspaces 数据集基础上添加上人工合成的雾制作而成的，因而标注信息和原 Cityspaces 数据集完全相同。在实验中，为了适配我们的模型要求的输入标签格式，我们对原始的 Cityspaces 和 Foggy Cityscapes 数据集先进行了一系列的标签转换：原始的数据集的标签是 json 格式的，我们通过一定的操作将图像的像素级标注 json 文件转换为 txt 格式标签，具体来说，转换后的每个图像对应的标签中的每一行表示该图像中的一个 Ground Truth Box，每一行包含 5 个分量：类别标签（0-c 之间的整型数字）、Ground Truth Box 的中心点横纵坐标 x 和 y 及其宽高 w 和 h（后面 4 个分量均为浮点数）。为了利用这两个数据集构造具有域偏移性质的数据集，我们在划分数据集时，把 Cityspaces 数据集中的 2975 张训练图片和 500 张验证图片作为我们的源域，这些图片为正常天气下的市区场景，我们把它们分别作为我们的训练集和验证集。同时，我们把 Foggy Cityspaces 数据集中的 1525 张测试图片作为我们的目标域，这些图片为大雾场景下的市区场景，识别难度大大提升，我们把它们作为我们的测试集。通过上述处理，我们就得到了具有域偏移性质的数据集。

NICO_Detection：上面介绍到，将 Cityspaces 和 Foggy Cityspaces 进行组合可以构造具有域偏移性质的数据集，但是因为源域和目标域都只有一个，这种域偏移程度比较弱，导致其验证效果具有一定的局限性。为了更好地验证我们模型的稳定性、鲁棒性以及泛化能力，我们除了使用一些公开的图像数据集进行组合，还自己手工标注了一些图像。具体来说，我们主要在 NICO 数据集上进行标注，该数据集在[42]中被提出，它是一个为分布转移问题而设计的图像识别数据集，包括 19 个类别和 10 个域，不同类别的域是不同的，域分割的标准也因不同的类别而异。例如，一些类别的域按照图像的背景划分，如“在水上”或“在草上”，而另一些类别的域则按照物体的姿势划分，如“跑步”或“站立”。来自原始 NICO 数据集的图像示例如图 4-1 所示。由于 NICO 仅仅是一个用于图像分类任务的数据集，且该数据集规模庞大，所以我们先在该数据集的子集上使用 lableme 工具进行手工标注，然后再进行一系列标签格式转换，得到了具有较大的域偏移性质的新的目标检测数据集，我们将这个新的数据集称为 NICO_Detection。具体来说，我们在构建 NICO_Detection 数据集时做了如下工作：1、仍然保持旧类别不变，但每个类别仅保留 6 个偏差较大的域，其中 4 个域作为训练集，另外 2 个域则分别作为验证集和测试集；2、每个类的每个域仅仅保留 50 至 60 张比较典型的图像样本，我们的一个抽样原则是：尽量使有的图

像样本中为大目标而有的图像样本中为小目标（不一定要多），这样做的目的是为了验证模型的泛化能力，判断模型能否很好地检测出各种不同尺度的目标。另一个抽样原则是：适量选取一些“难例”样本，如包含存在遮档、重叠、不完整等情况的目标的图像，这样做的目的是验证模型的鲁棒性以及模型对“难例”样本的检测能力。3、使用 lableme 工具在图像样本上进行手工标注时，除了保留原有的 19 个类别，我们还新添加了类别“person”，这是因为：在标注的过程中，我们发现“person”这一目标在很多图像样本中有出现，而且图像中的“person”类目标具有自身特征多样（姿态差异、肤色差异等）、所处背景特征变化大的特点，且“person”类目标往往与其它类目标有重叠，准确地对“person”类目标进行定位和分类具有一定的难度，添加这一类别将对验证模型的性能具有很大的意义。4、lableme 标注得到的标签类似于上面介绍的 Cityspaces 数据集的原始 json 格式标签，我们按照类似的方法进行标签格式转换，最终得到了带有适配模型输入标签格式的 NICO_Detection 数据集。

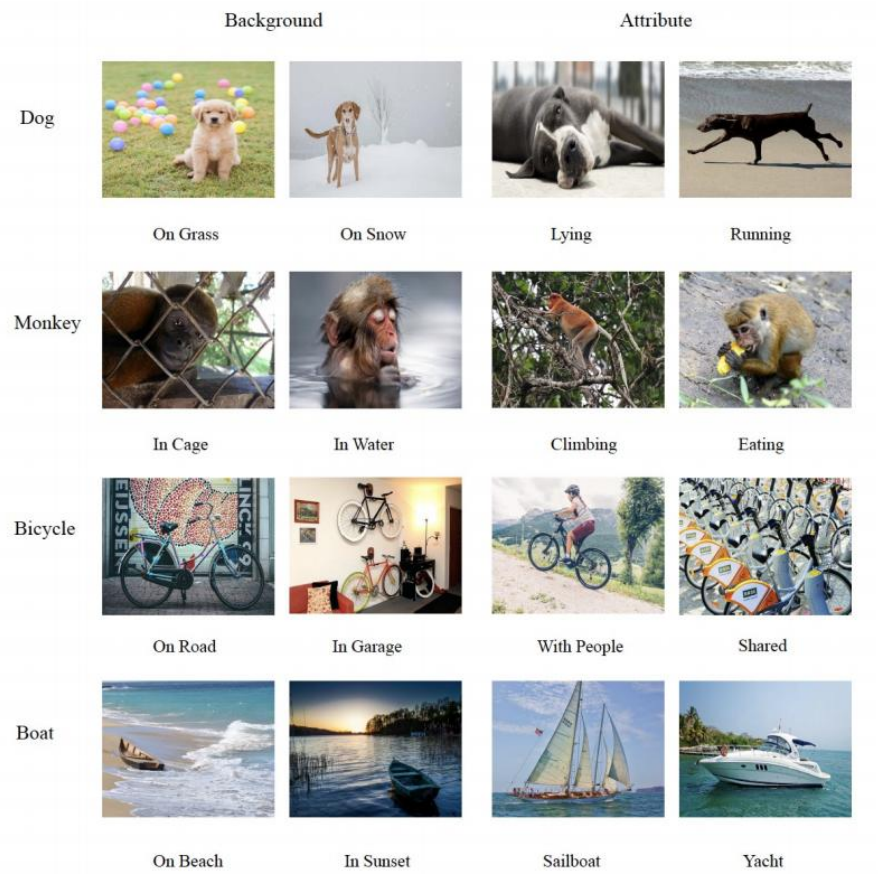

图 4-1　原始 NICO 数据集中的示例图像

4.3 实验分析指标

我们的评估主要使用了四种常见的目标检测分析指标，并进行一些可视化以便更直观的分析结果。

首先，我们对几个变量做出相关定义，如表 4-1 所示：

表 4-1 相关变量定义

	正样本	负样本
预测为正	True Positive (TP)	False Positive (FP)
预测为负	False Negative (FN)	True Negative (TN)

（一）Precision

Precision 表示精确率，它代表所有预测为正样本的结果中，预测正确的比率，其计算如公式(4-1)所示：

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

（二）Recall

Recall 表示召回率，它代表了所有正样本中被正确预测的比率，其计算如公式(4-2)所示：

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

（三）MAP50 与 MAP50-95

MAP (Mean Average Precision)表示平均精度均值。介绍这个指标之前，我们需要先认识 AP (Average Precision)。简单来说，AP 就是对 PR 曲线上的 Precision 值求均值，假设 PR 曲线对应的函数为 p(r)，同时假设 PR 曲线上的 Precision 值和 Recall 值的取值范围都在 0 至 1 之间，那么 AP 的计算如公式(4-3)所示：

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

上面介绍的 AP 是针对一个类别而言的，而在目标检测任务中涉及多个类别，在这里我们把所有类别的 AP 都计算出来后，再对它们求平均值，就得到了 MAP。需要注意的是，我们使用的评估指标是 MAP50 和 MAP50-95，这两个指标是在 IOU 阈值分别设置为大于 50、50 到 90 之间时计算得到的 MAP 值。

（四）F1-score

F1-score 是分类问题的一个衡量指标。一些多分类问题的机器学习竞赛，常常将 F1-score 作为最终测评的方法。它是精确率和召回率的调和平均数，最大为 1，最小为 0。对于某个类别，它是综合了 Precision 和 Recall 的一个判断指标，F1-Score 的值是从 0 到 1 的，1 最好，0 最差。F1-score 的计算如公式(4-4)所示：

$$\mathrm{F1-score}=\frac{2*\mathrm{Precision}*\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{4-4}$$

4.4 结果与分析

4.4.1 与 baseline 的对比

由于我们的模型在主框架上与 Yolov5 模型是类似的，我们选取 Yolov5 模型作为基准（baseline）进行对比，这里的对比我们主要在 4.2 小节介绍的三种数据集上展开，并通过一些可视化结果的定性分析和一些评估指标的定量分析对我们提出的 Stable_Yolo 模型的鲁棒性、稳定性和泛化能力进行了验证。

首先，总的来说，表 4-2 展示了我们的模型与 baseline 在这三个数据集上的几个主要性能指标的评估结果，表格中的 P、R、map50 及 map50-95 分别为 4.3 小节中介绍的 Precision、Recall、MAP50 和 MAP50-95。图 4-3 展示了三张 Stable_Yolo 的检测效果图。图 4-3 展示了我们的模型在 Cityspaces/Foggy Cityspaces 组合数据集上仅选取部分类别进行训练与测试得到的 F1-score 曲线。其中横轴表示 IOU 置信度阈值，纵轴表示 F1-score。

表 4-2 模型主要性能指标结果

		coco128	Cityspaces/Foggy Cityspaces			Nico_Detection
			all	car	some	
Yolov5	P	94.5	35.3	82.8	48.6	55.38
	R	91.3	25.6	63.7	31.6	45.25
	map50	96.0	27.1	73.4	31.8	55.73
	map50-95	83.0	13.6	48.3	15.0	50.32
Stable_ Yolo	P	94.6	35.9	83.7	49.5	59.25
	R	91.3	29.1	65.5	32.6	49.30
	map50	96.8	28.5	74.1	32.5	60.85
	map50-95	83.3	14.5	49.2	16.8	55.64

图 4-2 Stable_Yolo 检测效果样例

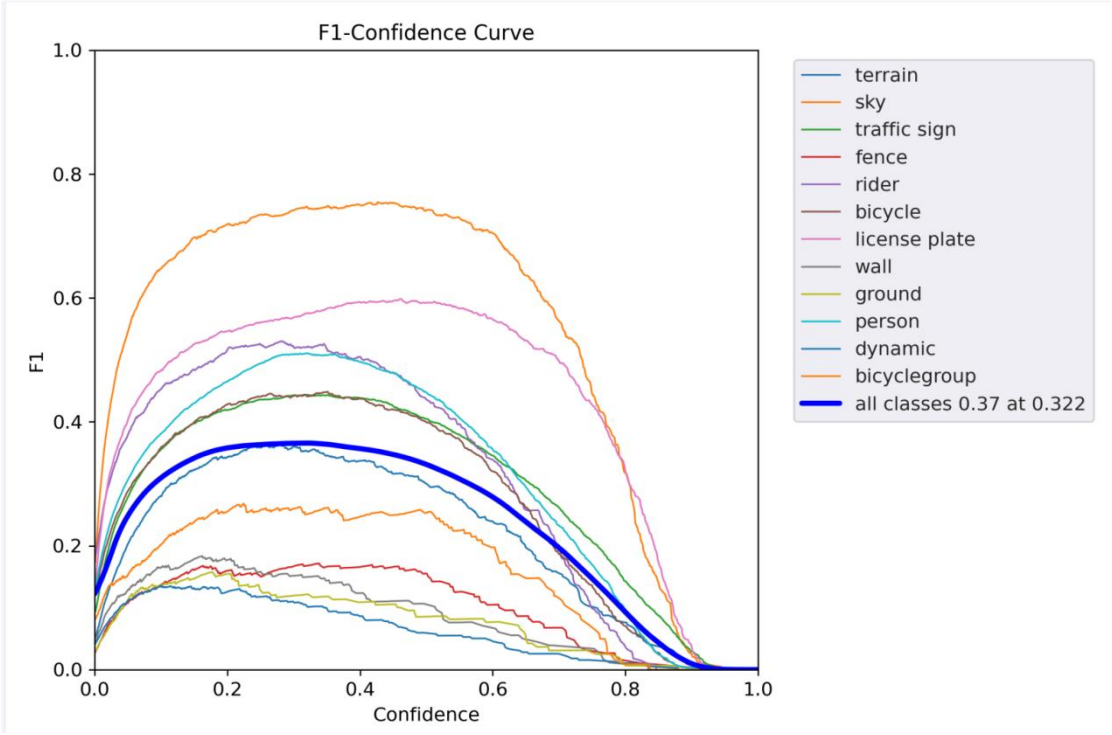

图 4-3 F1-score 曲线

其次，具体地对比与讨论每个数据集上的结果，我们做出如下分析：

1、coco128：在这个一般数据集上，我们的模型所表现的性能与 baseline 相当。这可能是因为我们的模型主要改进点是将基于因果推断的稳定学习引入到单级目标检测框架中，从而从一定程度上解决域泛化问题（域偏移问题也适用），提高模型的稳定性。而一般的数据集不存在明显的域泛化或域偏移，所以即使稳定性一般的模型（这些模型往往基于 I.I.D）也能表现出很好的性能，因此在像 coco128 这样的一般数据集上的对比不能很好地体现我们的模型的优势。

2、Cityspaces / Foggy Cityscapes：在这个组合数据集上，我们进一步通过在数据加载部分添加类别过滤功能对原组合数据集的类别进行了几种不同的选择。图 4-4 的(a)、(b)、(c)分别展示了三种类别选择的标签分布，其中横轴表示不同类别对应的编号，纵轴表示类别的实例（instance）数。具体来说，第一

种不进行过滤，选取 Cityspaces 数据集中的所有类别，一共包含 38 个类别，我们可以观察到这些类别的实例数大致呈现出正态分布；第二种类别过滤仅仅选取 car 类别，这相当于进行单类别目标检测，选取 car 类别是因为这个类别的实例数最多；第三种类别过滤仅选取 38 个类别中特征比较相似的 12 个类别，这样做出选择是因为具有相似特征的类别在进行分类时难度更大。表 3-1 中的 all、car、some 分别对应这三种不同的类别设置，根据对应的实验结果可以得到的结论是：1）相比一般数据集（如上文提到的 coco128 数据集），普通模型在具有域偏移性质的数据集上的预测性能会差很多，我们在这里称这种数据集为“特殊数据集”，这种数据集不满足独立同分布假设；2）与 baseline 模型相比，我们的模型在域偏移程度不大的“特殊数据集”上的测试性能稍微有所提升，这说明我们的工作有效地在一定程度上提高模型的稳定性。3）当图像中的类别数增加、同时实例数也随着增加的时候，不管是 baseline，还是我们提出的 Stable_Yolo, 性能都有所下降，这符合客观规律，同时也启发我们可以在 Stable_Yolo 的特征提取模块——即 Backbone 和 Neck 模块进行改进，使模型提取到的特征更具不变性和代表性，更有利于后面的分类任务。

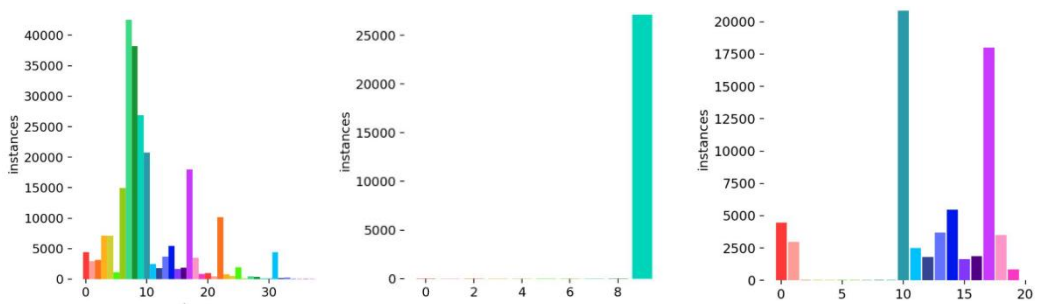

图 4-4 Cityspaces/Foggy Cityspaces 数据集三种类别选择的标签分布
(a) all：选择所有类别；(b) car：仅仅选择 car 类别；(c) some：选择 12 个典型类别

3、NICO_Detection：从表 4-1 的结果可以看出来。相比 Cityspaces/Foggy Cityspaces 组合数据集，我们的模型在 NICO_Detection 数据集上的性能提高的更为明显，这是因为这个数据集域偏移程度更大，同时域泛化程度也更大，这进一步验证了我们所做工作的有效性。

4.4.2 消融实验

两个消融实验都仅仅使用 Stable_Yolo，并仅仅在 NICO_Detection 数据集上进行，实验的超参数设置同在该数据集上 Stable_Yolo 与 baseline 的对比实验中超参数的设置一致，图 4-5 的(a)和(b)分别展示了两个消融实验的结果，其中纵轴都

表示精度 P（precision）,(a)的横轴表示随机傅里叶特征的维度相对默认值（默认值为 1）的倍数，(b)的横轴表示预保存的特征所占比例相对默认值(默认值为 0.9)的倍数。

具体来说，针对在随机傅里叶特征维度变化上的消融实验，我们得到如下分析结果：如果我们去掉所有的随机傅里叶特征，我们在公式(3-8)中的 Frobenius 范数正则化就会退化，这样会导致只能去除特征之间的线性相关性，而不能有效地去除特征之间的非线性相关性。图 4-5 的(a)证明了我们提出的基于 RFF 的样本加权策略用来消除特征之间的非线性依赖关系的有效性。

针对在预保存特征所占比例变化上的消融实验，我们发现：当预保存特征的大小减小到 0 时，在每个 batch 内部学习到的样本权值将会产生明显的方差；总的来说，随着预保存特征的数量的增加，一开始模型精度会显著增加，但到后面将会趋近一种收敛状态，精度仅仅略有提高，但样本权值的方差将显著下降，这表明预保存特征有助于全局学习样本权重，使得模型的泛化能力更加稳定。

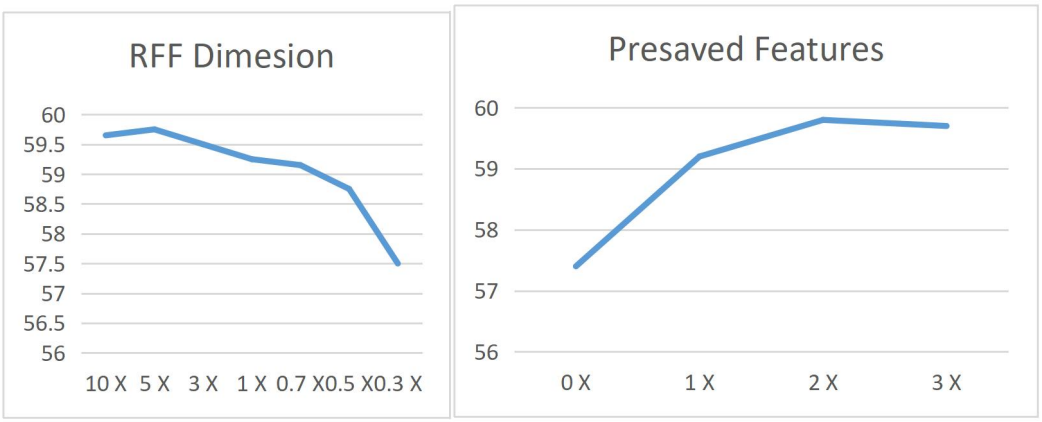

图 4-5 两个消融实验结果图

(a) 探究 RFF 的维度的影响；(b)探究预保存特征数量的影响

第五章　总结

为了提高用于目标识别的深度学习模型在域偏移或者域泛化情况下的稳定性，我们将基于因果推断的稳定学习的方法引入到单阶段目标检测框架中，通过全局性地学习样本权重去除特征间的相关性，使得模型能够学习到具有不变性的稳定特征与标签之间的强因果关系。在具体实现方面，我们按照这种思路设计并实现了一种新的称为 Stable_Yolo 的模型，并自己标注了一个适用于域泛化任务的目标检测数据集 NICO_Detection，然后在公开数据集 coco128、Cityspaces/Foggy Cityspaces 和自标注数据集 NICO_Detection 上进行了充分的实验验证。实验结果表明，该模型在准确率、召回率、置信度为 50 的 MAP（Mean Average Precision）以及置信度为 50 至 95 的 MAP 等多个目标检测常用的评估指标上均优于改进前的模型。同时，我们进一步通过两个消融实验进行深入分析和探讨，证明了我们提出基于 RFF 特征的全局样本重加权方法可以提高模型的稳定性和泛化能力。总之，本文提出了一种创新性的解决方案，以解决目标识别领域中存在的域泛化和域偏移等重要问题，并通过对算法思路的分析和实验验证，证明了这个方案的可行性和有效性，为未来的研究工作提供了新的思路和方向。

参考文献

[1]Shen Z, Liu J, He Y, et al. Towards out-of-distribution generalization: A s urvey[J]. arXiv preprint arXiv:2108.13624, 2021.

[2]Chen K, Lee C G. Incremental few-shot learning via vector quantization in deep embedded space[C]//International Conference on Learning Representations. 2021.

[3]Cubuk E D, Zoph B, Mane D, et al. Autoaugment: Learning augmentation policies from data[J]. arXiv preprint arXiv:1805.09501, 2018.

[4]Müller R, Kornblith S, Hinton G E. When does label smoothing help?[J]. Advances in neural information processing systems, 2019, 32.

[5]Li X, Chen S, Hu X, et al. Understanding the disharmony between dropout and batch normalization by variance shift[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 2682-2690.

[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]Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[8]Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.

[9]He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.

[10]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.

[11]Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

[12]Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[13]Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[14]Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications[J]. arXiv preprint arXiv:2209.02976, 2022.

[15]Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[J]. arXiv preprint arXiv:2207.02696, 2022.

[16]Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.

[17]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.

[18]Zou Z, Shi Z, Guo Y, et al. Object detection in 20 years: A survey[J]. arXiv preprint arXiv:1905.05055, 2019.

[19]Zhang L, Gao X. Transfer adaptation learning: A decade survey[J]. arXiv preprint arXiv:1903.04687, 2019.

[20]Zhang W, Ouyang W, Li W, et al. Collaborative and adversarial network for unsupervised domain adaptation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 3801-3809.

[21]Chen D, Wang Y, Huang J, et al. Cross-Domain Feature Learning for Large-Scale Social Media Analysis[J]. ACM Transactions on Intelligent Systems and Technology, 2018, 9(4): 1-20.

[22]Sun Q, Liu Y, Chua T S, et al. Meta-transfer learning for few-shot learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 403-412.

[23]Xu R, Chen Z, Zuo W, et al. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 3964-3973.

[24]Long M, Cao Y, Wang J, et al. Learning transferable features with deep adaptation networks[C]//International conference on machine learning. PMLR, 2015: 97-105.

[25]Kuang K, Cui P, Athey S, et al. Stable prediction across unknown environments[C]//proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 1617-1626.

[26]Kuang K, Xiong R, Cui P, et al. Stable prediction with model misspecification and agnostic distribution shift[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(04): 4485-4492.

[27]Shen Z, Cui P, Liu J, et al. Stable learning via differentiated variable decorrelation[C]//Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining. 2020: 2185-2193.

[28]Liu J, Shen Z, Cui P, et al. Distributionally robust learning with stable adversarial training[J]. IEEE Transactions on Knowledge and Data Engineering, 2022.

[29]Liu J, Hu Z, Cui P, et al. Heterogeneous risk minimization[C]//International Conference on Machine Learning. PMLR, 2021: 6804-6814.

[30]Zhang X, Cui P, Xu R, et al. Deep stable learning for out-of-distribution generalization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 5372-5382.

[31]Li M, Liu J, Xu Y, et al. Invariant risk minimization for non-stationary behaviors[C]//International Conference on Machine Learning. PMLR, 2019: 4113-4122.

[32]Zhang K, Peters J, Janzing D, et al. Kernel-based conditional independence test and application in causal discovery[J]. arXiv preprint arXiv:1202.3775, 2012.

[33]Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. Journal of Machine Learning Research, 5(Jan):73–99, 2004.

[34]Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Scholkopf. Kernel measures of conditional dependence. In Advances in neural information processing systems, pages 489–496, 2008.

[35]Eric V Strobl, Kun Zhang, and Shyam Visweswaran. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery.

[36]Kun Kuang, Ruoxuan Xiong, Peng Cui, Susan Athey, and Bo Li. Stable prediction with model misspecifification and agnostic distribution shift. In AAAI, pages 4485–4492, 2020.

[37]LeCun Y, Boser B, Denker J, et al. Handwritten digit recognition with a back-propagation network[J]. Advances in neural information processing systems, 1989, 2.

[38]Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.

[39]Choi J, Chun D, Kim H, et al. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 502-511.

[40]Zhang X, Cui P, Xu R, et al. Deep stable learning for out-of-distribution generalization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 5372-5382.

[41]Wang C Y, Liao H Y M, Wu Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020: 390-391.

[42]He Y, Shen Z, Cui P. Towards non-iid image classification: A dataset and baselines[J]. Pattern Recognition, 2021, 110: 107383.

[43]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.

[44]Chang Y, Liu C Y, Cheng W H, et al. Real-Time RGB-D Semantic Mapping with Depth-Estimated Semantic Flow[J]. IEEE Robotics and Automation Letters, 2019, 4(2): 1197-1204.

[45]Liu M, Yang J, Zhu Y, et al. Intelligent Video Surveillance System Based on Deep Learning and Internet of Things[J]. IEEE Access, 2020, 8: 127615-127628.

[46]Kang J, Kim S H, Lee K M. Deep Learning-based Defect Detection in Industrial X-ray Images[J]. Journal of Mechanical Science and Technology, 2020, 34(1): 1-15.

[47]Kwon H, Lee J Y, Kim Y M. Aerial Object Detection Using Convolutional Neural Networks[J]. Journal of Sensors, 2020, 2020: 1-9.

致　谢

行文于此，落笔为终，已经写到了论文的最后一章节，我的本科生涯也即将结束，四年的时光仿佛弹指一挥间，仿佛仍在梦的昨天。四年来，“天大”两个字对于我来说已经变得更加亲近、更加充满韵味。四年来，我也经历了许多，不知不觉间已经慢慢变成了独当一面的大人。置身天大，目光所及之处，皆是各种回忆，四年来，我度过了人生中最青春的年华，纵有万般不舍，但仍然心怀感激。

首先，我要特别感谢我的论文指导教师刘若楠老师。在做学术上，在工作中，身为一名学者，身为一名导师，刘老师有着严谨的教学态度、严密的逻辑思维、丰富的学科知识，她认真负责的工作态度让我在学习和做人方面都受益匪浅，在整个论文的定题、修改过程中也少不了刘老师指明方向、启发思路、细心审查、点正错误。在生活中，在日常交流中，刘老师又像一位温柔的大姐姐，我丝毫没有在她身上看到任何“架子”，她总是给我一种很亲近的感觉，这也鼓励了我有什么问题敢于主动与她交流，有什么想法敢于主动与她分享，有什么思考敢于主动与她探讨。最后，我想在这里对刘老师说的是，我将牢记您的教诲，不管对于这个课题，还是对于做学术乃至做人的态度，我都将不断奋力拼搏，不断深入反思，不断超越自我。

其次，我要感谢我的家人，感谢家人对我二十多年的悉心照顾。虽然我的家庭条件相当一般，甚至有些困难，但是我的父母从来没有给我任何经济上的压力，他们把这些压力都扛在自己的肩上，放在自己的心头，只为让我无忧无虑的长大。在我一路的成长过程中，是家人在背后一直默默地关心着我、鼓励着我，对我今后的路，他们也义无反顾的支持着我、信任着我。我的坚强、我的勇气、我的自信不是与生俱来的，是他们的爱催生了我前进的动力，成为我奋斗的最大精神支柱。这些爱我会永远牢记于心。以前，还有现在，都是他们把爱无限度的给了我；现在，还有将来，我会尽我所能把爱最大限度的给他们。负重前行，脚下才会更加铿锵有力，我终将用我的行动让他们过上更好的生活。

再者，我要感谢我的朋友们，感谢每一次相遇。其实，大学四年我一直在奔跑，在逐梦，几乎从未停下脚步。变强的过程中难免需要更多的独处与思考，难免要忍受一些孤独，但我不想让自己成为一座孤岛，也感谢遇到了你们，让我没有成为一座孤岛。是你们的出现，使得我心灵的小岛焕发出勃勃生机，使得我成为一个有血有肉的人，而不是在无尽的忙碌之中变得麻木。谢谢彤彤、小元、颖慧、子淇等等小仙女们出现在我的生命线里，给了我温暖，给了我鼓励。这里还

想额外感谢一下小徐童鞋，虽然我们这个学期才认识，但是真的有种相见恨晚的感觉，虽然你平时总是哈哈哈哈哈的，但是本质上是个十分关心朋友、十分靠谱、十分值得信赖的人，和你一起泡图书馆、泡自习室比一个人学习效率高吖！真的也很喜欢看见你认真的样子，现在你看着我毕业，看着我上岸读研，几年后我也希望能看着你毕业，看着你保研成功，还是那句话：“一起加油吖！”

最后，感谢智算学院的所有老师，特别是我的辅导员涛哥，也感谢一路上帮助过我的学长学姐，甚至是学弟学妹（哈哈哈我比较笨，偶尔有些问题向学弟学妹请教也正常）。我记得之前在上《计算机系统基础》这门课程的时候，由于疫情，我们在家里没有返校，课程都是居家线上进行的。当时初学的我不太会配环境，仍记得授课老师在课后还专门为我录了一个视频指导我，对此我其实是相当感动的。诸如此类的还有很多，我都记得，我记得辅导员涛哥、露姐对我的鼓励、我记得宿管阿姨对我的关心，我记得很熟的一位食堂阿姨给过我的帮助，我记得每一个人对我的好，这些我都会珍藏在心里一辈子，哪怕那只是很微不足道的小事，对我来说都是弥足珍贵的感动。

时光荏苒，但是过去不会遗忘；终有一别，但是我们来日方长。6 月里，天大，再见；8 月里，天大，我们还会再见。愿再相见，我会是更优秀的我！