

密级：_____

浙江大学

硕士 学位 论 文



论文题目 基于卷积神经网络的目标检测算法
其增量学习研究

作者姓名 王世豪

指导教师 钱云涛 教授

学科(专业) 计算机科学与技术

所在学院 计算机科学与技术学院

提交日期 2018/03/25

A Dissertation Submitted to Zhejiang University for the Degree of Master of Engineering



TITLE: Research on Incremental Learning of
Object Detection Algorithm Based on
Convolutional Neural Network

Author: Shihao Wang

Supervisor: Prof. Yuntao Qian

Subject: Computer Science and Technology

College: Computer Science and Technology

Submitted Date: 2018/03/25

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 浙江大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名: 王世豪 签字日期: 2019年3月28日

学位论文版权使用授权书

本学位论文作者完全了解 浙江大学 有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 浙江大学 可以将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、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名: 王世豪

导师签名: 张晓东

签字日期: 2019年3月28日

签字日期: 2019年3月28日

学位论文作者毕业后去向:

工作单位:

电话:

通讯地址:

邮编

摘要

图像中的目标检测是计算机视觉领域的核心任务之一，是计算机理解图像的基石，可用于智能视频监控、无人驾驶、机器人、智能家居、医疗图像辅助诊断等需求、产品中。

自 ImageNet 图像分类大赛 2012 年的冠军 AlexNet 诞生以来，卷积神经网络 (Convolutional Neural Network, CNN) 在计算机视觉领域得到了广泛的应用。以 CNN 为主的深度学习技术也大幅度提高了目标检测任务的准确率。凭借着 CNN 的优势和大量学者的不懈努力，如今计算机程序进行目标检测的效果几乎可以和人类视觉比肩。其中有代表性的工作如 R-CNN、Fast R-CNN、Faster R-CNN、YOLO、FPN、RetinaNet。在特定场景，目标检测算法的准确率也逐渐提升至快饱和的状态。但一些新兴的问题也逐渐进入人们的视野。计算机视觉中的增量学习，当属这些新兴问题中的一员。而目标检测是计算机视觉中的主要任务。所以，本文即是研究目标检测的增量学习。

目前，在目标检测领域准确率较高的方法均基于卷积神经网络。但联结类方法（如神经网络）存在灾难性忘记(Catastrophic Forgetting)的问题。该问题具体为：若在一个已有的神经网络模型上训练新任务时，会使该模型已有的能力急剧丧失。所以，基于卷积神经网络的目标检测算法也存在这样的问题，即：当往一个已有的目标检测模型中添加新的可检测类别时，若用新类别的数据集直接训练已有网络，那么训练出的模型会很大程度地丧失在旧类别上的检测能力。

本文的研究目标即是：如何更好地往已有的目标检测模型中添加新的可检测类别。本文研究了两种方法。

第一种方法是基于特征提取思想的方法。这是一种直观的，为新类别添加专有层的方法。本文的贡献是：提出了在使用该方法时，可提高准确率的细节。并通过实验，研究了该方法的一些性质。本文提出：(1) 前移新类别专有层的添加位置，能可观地提高网络在新类别上的检测准确率。(2) 为新类别添加浅层旁路，

能使网络在新类别上的检测准确率有一些提升。

第二种方法是基于保留损失函数的方法。该方法基于 LwF^[28] (Learning Without Forgetting)。LwF 是 Zhizhong Li 等人提出的在图像分类上进行增量学习的方法。LwF 借鉴了模型压缩中，广泛被称为模型蒸馏的方法。本文的贡献是：率先将 LwF 的思路应用到了具有区域建议网络(Region Proposal Network, RPN)模块的流行目标检测算法上，在目标检测的增量学习任务上得到了比现有方法更高的准确率。并通过大量实验研究了，在借鉴 LwF 的思路为目标检测模型添加新的可检测类别时，值得考虑的各种细节。本文对多个细节进行了实验，选出了在这些细节上最好的做法。这些细节的处理方式，将会对模型最终的准确率产生一定的影响。

关键词：卷积神经网络，增量学习，目标检测，灾难性忘记，模型蒸馏

Abstract

Object detection in images is one of the core tasks in the computer vision field and the cornerstone of computer understanding images. It can be applied to intelligent video surveillance, automatic drive, robots, smart home, medical image assistant diagnosis and other needs, products.

Since the birth of AlexNet, the champion of the ImageNet Image Classification Competition in 2012, the Convolutional Neural Network (CNN) has been widely used in the computer vision field. Deep learning technologies especially CNN also greatly improve the accuracy of object detection task. With the advantages of CNN and the unremitting efforts of a large number of scholars, nowadays computer programs can almost match human vision in object detection tasks. Some representative works were produced like R-CNN, Fast R-CNN, Faster R-CNN, YOLO, FPN and RetinaNet. In specific scenarios, the accuracy of object detection algorithms is gradually improved to the saturation nearly. But some new problems are gradually coming into people's vision. Incremental learning in computer vision is one of these emerging problems. Object detection is the main task of computer vision. Therefore, this thesis is to research the incremental learning of object detection.

At present, the most accurate methods in the object detection field are all based on CNN. However, the connection type methods such as neural networks have catastrophic forgetting problem. This problem is: if a new task is trained on an existing neural network model, the existing ability of this model will be lost sharply. Therefore, the CNN-based object detection algorithms also have such problem, that is, when adding new detectable categories to an existing object detection model, if the existing model is directly trained with the data of new categories, the final model will lose the detection ability on the old categories to a great extent.

The goal of this thesis is: how to better add new detectable categories to an existing object detection model. Two methods are researched in this thesis.

The first method is based on the idea of feature extraction. This is an intuitive

way to add a proprietary layer to new categories. The contribution of this thesis is: proposed the details which can improve accuracy when using this method, and researched some properties of this method through experiments. This thesis puts forward: (1) Moving forward the adding position of the new categories' proprietary layer can significantly improve the detection accuracy on the new categories. (2) Adding shallow bypass layers for the new categories' detection can improve the detection accuracy on the new categories.

The second method is retaining loss function method, and it can be regarded as proposed by this thesis. It is based on LwF^[28] (Learning Without Forgetting). LwF is an incremental learning method for image classification proposed by Zhizhong Li et al. LwF drew inspiration from model compression method, which is widely known as model distillation. The contribution of this thesis is to take the lead in applying LwF idea to popular object detection algorithms which have Regional Proposal Network (RPN) module. And this thesis got a higher accuracy baseline on this task than state of the art methods. Through a large number of experiments, it is worth considering various details when adding new detectable categories to the object detection model with the idea of LwF. In this thesis, many details are experimented and the best ways to deal with these details are selected. These details will have a certain impact on the final accuracy of the model.

Keywords: Convolutional neural network, incremental learning, object detection, catastrophic forgetting, model distillation

目录

摘要	i
Abstract.....	iii
第 1 章 绪论	1
1.1 目标检测及其增量学习简介	1
1.2 研究背景及意义	1
1.3 国内外研究现状	6
1.3.1 目标检测算法	6
1.3.2 基于卷积神经网络的增量学习	9
1.4 本文创新点	13
1.5 本文组织结构	13
第 2 章 理论基础及相关技术	15
2.1 目标检测算法的评价指标	15
2.2 卷积神经网络原理	16
2.2.1 卷积层	16
2.2.2 激活层	17
2.2.3 池化层	17
2.2.4 全连接层	18
2.2.5 Softmax 层	19
2.2.6 Batch Normalization 层	19
2.3 残差网络	21
2.4 Faster R-CNN 目标检测算法	23
2.5 特征金字塔网络目标检测算法	27
2.6 模型蒸馏	29
第 3 章 实验数据的准备	31

3.1 本文使用的数据集	31
3.1.1 Pascal VOC 数据集.....	31
3.1.2 MS-COCO 数据集	32
3.1.3 Open Image Challenge 2018 数据集	33
3.2 数据集的使用与处理	34
第 4 章 基于特征提取思想的研究	36
4.1 具体方法	36
4.2 实验结果	38
4.3 微调层数与新类别准确率的关系	39
4.4 已有网络可检测类数与新类别准确率的关系	41
4.5 新类别专有层的添加位置	42
4.6 添加浅层旁路	45
4.7 本章小结	47
第 5 章 基于保留损失函数方法的研究	49
5.1 具体方法	49
5.2 实现细节	52
5.3 实验结果	53
5.4 回归目标的选取	56
5.5 物体候选框的选取	58
5.6 预训练的重要性	59
5.7 保留误差和新类别检测误差的赋权	60
5.8 为新类别添加第二阶段专有层	61
5.9 本章小结	62
第 6 章 总结与展望	64
参考文献	66
攻读硕士学位期间主要的研究成果	71
致谢	72

图目录

图 1.1 目标检测任务图示	1
图 1.2 MS-COCO 数据集的验证集中，被漏标的图片	4
图 1.3 使用 ResNet-50-FPN 目标检测模型，对该图片检测的结果	4
图 1.4 目标检测算法将船的舵轮识别为椅子	5
图 1.5 目标检测算法将笔筒中的笔识别为牙刷	5
图 1.6 四种为分类网络添加新的可识别类别的方法	10
图 2.1 残差网络中的第一类残差模块	21
图 2.2 残差网络中的第二类残差模块	22
图 2.3 Faster R-CNN 网络结构图	23
图 2.4 RPN 模块	24
图 2.5 anchor 的示意图	24
图 2.6 CLS 模块	26
图 2.7 FPN 网络结构图	28
图 3.1 Pascal VOC 数据集中的图片	32
图 3.2 MS-COCO 数据集中的图片	33
图 3.3 Open Image Challenge 2018 数据集中的图片	34
图 4.1 改动后的 RPN 模块和 CLS 模块	37
图 4.2 为新类别添专有的 RPN 隐层图示	40
图 4.3 三种新类别专有层的添加位置	44
图 4.4 添加浅层旁路后的 FPN 网络结构	46
图 5.1 基于保留损失函数方法的训练过程	52
图 5.2 用训练好的被训练网络进行检测的结果展示(a).....	55
图 5.3 用训练好的被训练网络进行检测的结果展示(b).....	56

表目录

表 4.1 本章方法的实验结果	39
表 4.2 微调层数与新类别准确率的关系	40
表 4.3 已有网络可检测类数与新类别准确率的关系	42
表 4.4 三种新类别专有层添加位置的实验结果	44
表 4.5 添加浅层旁路后的结果	47
表 5.1 不添加保留误差的结果	53
表 5.2 本章方法的准确率	53
表 5.3 与 Shmalkov 等人的方法的对比	54
表 5.4 回归目标的选取对准确率的影响	57
表 5.5 使用不同的物体建议框训练对结果的影响	59
表 5.6 无预训练的结果	59
表 5.7 保留误差和新类检测误差赋权对结果的影响	61
表 5.8 为新类别添加第二阶段专有层的结果	62

第1章 绪论

1.1 目标检测及其增量学习简介

典型的目标检测算法任务是：给定一幅图像和一组类别名（例如：猫、狗、汽车），要求算法输出一组平行于图像边界的矩形框的坐标。这些矩形框要能很好地框住既定类别的物体。且算法要能正确输出每个矩形框框住的物体的类别名。目标检测任务的图示，如图 1.1 所示。



图 1.1 目标检测任务图示

对于目标检测算法的增量学习，该任务是：当存在一个目标检测程序后，为其添加新的可检测类别。并且使得，进行检测时增加的时间开销尽可能少，新、旧类别的检测准确率尽可能高。

1.2 研究背景及意义

2012 年，第 3 届 ImageNet 图像分类大赛^{[1][2]}被举办。加拿大多伦多大学的 Alex Krizhevsky、Ilya Sutskever、以及 Geoffrey Hinton 三人组成的团队获得了冠军。且他们的方法 AlexNet^[3]高出第二名 10.8% 的准确率^[4]。他们的方法：深度卷积神经网络从此备受关注，且成功应用在了计算机视觉的诸多方面。

深度卷积神经网络的普及，也离不开并行计算的发展。21 世纪以来，人们逐渐产生使用显卡(Graphics Processing Unit, GPU)进行并行计算的想法，这种计算方式被称为 GPGPU(General-purpose computing on graphics processing units)^[5]。

于是，英伟达公司在 2006 年，推出了 CUDA。这是一个并行计算库，用户使用其中规定的语法和函数库，就可以将自己的代码并行地运行在英伟达的 GPU 上。AlexNet 正是使用了 CUDA 进行并行计算。随后微软公司推出了跨显卡平台的 DirectCompute 接口，苹果公司和 Khronos 组织推出了跨显卡平台的 OpenCL 接口。使用 GPU 进行并行计算使神经网络的训练和测试的速度提升了数十甚至上百倍，使得神经网络的应用和研究变得普及。

在 AlexNet 出现之后，众多学者利用卷积神经网络进行目标检测研究，取得了很好的结果。基于卷积神经网络的目标检测算法，比手工设计特征，然后再用机器学习算法进行训练的方法好很多。

目前，目标检测技术已应用到生活的许多方面，将来技术的改进还会持续在这些领域产生价值。这些领域包括：智能视频监控、无人驾驶、智能化交通系统、无人机、机器人、智能家居、军事、医疗图像辅助诊断、工业自动化生产、人机交互、基于内容的图像检索、娱乐。

但目前的目标检测模型都是进行一次训练后即投入使用。在使用过程中模型不能变更任务。若要更改模型的任务，如添加新的可检测类别，需要将模型同时在新、旧类别数据集上联合进行训练，哪怕要添加的只是一个类别。而旧类别数据集可能很大，使得训练的时间很长，资源占用较多，甚至旧类别数据集可能无法再被获取。所以，以这种方式去添加新的可检测类别有很大的弊端。

本文即是研究，如何更好地为已有的目标检测模型添加新的可检测类别，并在只使用新类别数据集的情况下。本文研究的目标检测增量学习，新添加的类别可以是某个类别的子类，甚至是一个个体，而不仅限于一个个相对独立的类别，如：猫、狗。

目标检测算法的增量学习，可增强目标检测算法在上述诸多应用中的能力，使其更加灵活，方便人们的使用。除此之外，下面具体着重地介绍其三点应用或意义：

视频监控领域的应用。随着深度学习、计算机视觉、目标检测算法的发展，智能视频监控已得到了广泛普及。智能视频监控程序若可添加新的可检测类别，

那么势必会在使用上带来极大的便利。因为会出现使用者想要添加新的可检测类别情况。例如某地公安机关想用他们的视频监控程序，检测出所有戴鸭舌帽的人，但已有的程序不能检测这个类别。这时，可增量的目标检测算法即可发挥它的优势。

机器人领域的应用。对于一个机器人来讲，不能添加新的可检测类别，相当于其视觉上的目标检测功能在其出厂时就已定型，不能根据后期的需要添加新的类别，这显然会对人们的使用产生限制。批量生产的机器人产品适用于其被训练的环境，但客户所处的环境可能还需要机器人再进行学习，才能发挥它的作用。这样的情况例如：家用机器人，智能服务机器人。

提高现有目标检测算法的准确率。通过可视化现有目标检测算法在公开数据集的验证集/测试集上预测的物体框。本文发现：目前先进的目标检测算法在公开数据集的验证集/测试集上进行测试时，存在将背景错判为某类目标的情况，这些情况拉低了目标检测算法的准确率。

本文发现，不少这类情况其实是验证集/测试集漏标导致的，即目标检测算法的判断是正确的。验证集/测试集漏标在 Pascal VOC^[6] 和 MS-COCO^[7] 数据集中均有出现。如在 MS-COCO 2014 数据集的 mini 验证集（等价于 MS-COCO 2017 验证集）中，名称为 000000451144.jpg 的图片（如图 1.2），图中左侧人的双肩背包被漏标，没有被标记为双肩背包(backpack)。



图 1.2 MS-COCO 数据集的验证集中，被漏标的图片

(图片选自 MS-COCO 验证集，图片名称：000000451144.jpg)

而 Detectron 开源代码库^[8]提供的基于 ResNet-50 的 FPN 模型，检测出了左边人背的双肩包，并且给出了较高的概率 97.6%。该模型对这张图片，关于双肩包类别的检测结果如图 1.3 所示。



图 1.3 使用 ResNet-50-FPN 目标检测模型，对该图片检测的结果

除了验证集/测试集漏标外，更多的情况确实是模型将背景错判为了某类目标。目标检测算法将不在数据集标注类别内的其他类物体，判断为了某类目标。例如，在图 1.4 中，将船的舵轮识别成椅子，给出的概率为 81%。



图 1.4 目标检测算法将船的舵轮识别为椅子

(图片选自 Pascal VOC 2007 测试集, 图片名称 003012.jpg)

在图 1.5 中, 将笔筒中的笔识别成牙刷, 给出的概率为 84%。

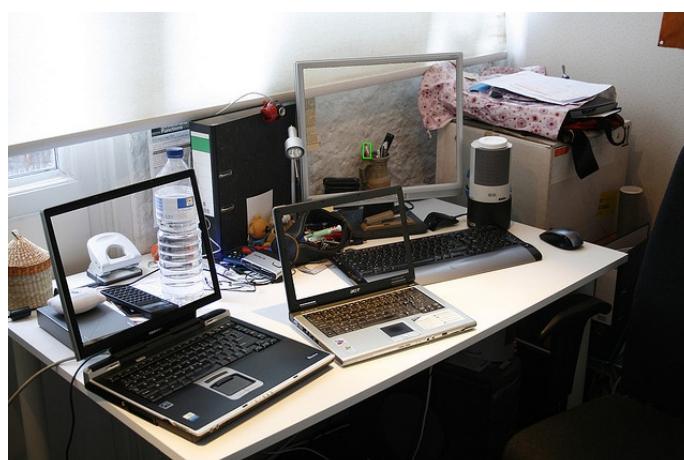


图 1.5 目标检测算法将笔筒中的笔识别为牙刷

(图片选自 MS-COCO 验证集, 图片名称 000000570782.jpg)

这些物体虽然是和被误识的类别中的物体很像。但本文认为, 如果将这些物体作为新的类别去训练神经网络, 则这些物体真正类别的预测概率, 应该能高过被误识类别的预测概率。从而可以减少这种误识情况的发生。

而目标检测算法的增量学习, 是研究怎么向已有的模型添加新的可检测类别。它最理想的目标, 是使目标检测算法可以不断地添加新的可检测类别。所以,

又根据如上论述，它具有提高现有目标检测算法准确率的潜力。

1.3 国内外研究现状

1.3.1 目标检测算法

图像中的目标检测一直是计算机视觉中较被关注的问题。在深度学习兴起之前，目标检测的过程一般是：在训练阶段，对物体区域和非物体区域，用人工设计的特征提取方法提取特征。得到这些特征后，将这些特征作为训练样本，使用一种机器学习分类算法，训练得到一个分类器。在测试阶段，使用一个滑动窗口在图像上滑动。在滑动窗口停留的每个位置，使用人工设计的特征提取方法从图像中提取特征。将每个位置的特征送入训练好的分类器进行分类，即完成了目标检测。比较典型的方法有：Viola–Jones 目标检测算法^[9]（该算法以在人脸检测上的应用而著称）、HoG+SVM^[10]、可形变部件模型^[11](Deformable Part Model, DPM)。

自 ImageNet 图像分类大赛 2012 年的冠军 AlexNet 诞生以来，以卷积神经网络为主的深度学习逐渐渗透到各学科当中。

开始有研究者将用于图像分类的 CNN 应用于目标检测，较有名的早期论文如 Ross Girshick 等人的 R-CNN^[12]。在 R-CNN 中，先用非深度学习的方法(Selective Search^[13]或 Edge Boxes^[14])将图像中可能是物体的区域提取出来（感兴趣区域，Region of Interest, RoI），而后将这些 RoIs 调整成相同大小，堆叠起来输入一个 CNN 中，再将该 CNN 最终输出的特征用 SVM 进行分类，来判断每个 RoI 具体是什么类别的物体。同时，将 CNN 最终输出的特征输入一个线性回归器，进行物体矩形框（下文简称为物体框）变换参数的回归，用来调整 RoI 的坐标，以便 RoI 的坐标能更好地与真实物体框重合。后来，Girshick 又提出了 Fast R-CNN^[15]。这篇文章相较 R-CNN 主要的改进是：(1) 原先 R-CNN 是将 RoIs 堆叠起来输入 CNN，但该文提出先将整个图像输入 CNN，然后在特征图上取 RoIs 对应的特征，再将 RoIs 对应的特征子图堆叠起来输入后续的全连接层。由于全连接层需要输入数据的维度是固定的，这里作者受文献[16]的启发，提出了 RoIPooling。即像文献[16]那样，平分特征图来做池化，以将特征池化成想得到的维度，而不是以固

定的核大小（例如 2x2）来做池化操作。（2）不再使用 SVM 和线性分类器来做最后的分类和回归，而直接使用神经网络输出这些结果。然而，无论是 R-CNN，还是 Fast R-CNN，均先使用了 Selective Search 进行 RoI 提取。后来 Shaoqing Ren 等人考虑到该过程耗时较长，且该过程的计算和后续计算没有共享过程，故主要针对此处进行了改进，进而提出了 Faster R-CNN^[17]。提出使用 CNN 进行 RoI 的提取。他们观察到，CNN 中卷积层得到的特征图上，每个像素位置，实际就是由原图一个区域的像素计算得来的。若对该特征图做一个核为 1x1 的卷积，实际上可用作判断每个像素点对应的原图区域里有无物体的过程。且用作判断每个位置有无物体的网络，和 RoI 要输入的网络可以共享几乎所有卷积层。于是，原先提 RoI 的时间开销，基本被缩减为 0，很大地加快的检测的速度。

之后 Liu Wei 等人基于 Faster R-CNN 提出了 SSD(Single Shot MultiBox Detector)^[18]。这篇文章主要阐述了两个观点：（1）可以不用提取 RoI，而是直接判断每个位置是什么类别的物体。（2）可以在多个特征图上，进行判断。因在 CNN 中，不同位置的特征图上，每个像素点有不同的感受野。即浅层的特征图适合发现小目标，深层的特征图适合发现大目标。实验证明，SSD 在多个特征图上检测目标，确实提高了小目标的检测准确率。

与此同时，Redmon Joseph 等人提出了 YOLO(You Only Look Once)^[19]，它和 Faster R-CNN 有较大不同。YOLO 中同样也没有提取 RoI 的过程，同样是直接判断图像中每个位置是什么类别的物体。但它还有较特别的地方，Faster R-CNN 和 SSD 中，都采用在特征图上执行卷积去对图像中各个位置做判断。而 YOLO 直接将 CNN 得到的特征图变形(Flatten)成一列向量后，输入全连接层去判断每个位置是什么类别的物体。这样也可以得到比传统检测方法好的结果，且速度比 Faster R-CNN 和 SSD 都要快，但准确率不如 Faster R-CNN 和 SSD。

再之后，Jifeng Dai 等人提出 R-FCN^[20]，该文主要基于这样一个思路：深度卷积网络对一个图像内物体的位置是不敏感的，即不论物体具体在哪个位置，只要物体具有一定大小，具有的特征足够明显，深度卷积网络就由很大能力对图像做正确的分类^[20]。但在目标检测任务中，想要得到目标的精确位置，故该任务可

能与深度卷积网络的前述特性相悖。于是这篇文章提出，判断一个区域有没有物体，需通过 9 个该区域的子区域投票决定。例如，若一物体只有一半在某区域内，则该区域另一半的子区域就会对该区域有这个物体这件事，给出较低的预测概率，从而使该区域包含该物体的预测概率降低。再后来，Tsung-Yi Lin 等人为了提高小目标的检测准确率，提出了 FPN^[21]，即本文主要基于的目标检测算法。该方法和 SSD 有相似的思路，即为了检测小目标，想在较浅层的特征图上对图像区域做判别。但 FPN 将浅层特征图和深层特征图进行了融合，以此来克服浅层特征图的特征，可辨识性不足的缺陷，从而提高浅层特征图上小物体的识别精度。Zhiqiang Shen 等人提出了 DSOD^[22]，该文和前面这些文章的目的稍有不同，前面文章的网络都是先在 ImageNet 上进行预训练，然后再在目标检测的数据集上进行微调(Fine-tune)得到的。但这篇文章描述的模型是直接在目标检测数据集上训练得到的，例如 Pascal VOC 数据集。这篇文章仅利用 Pascal VOC 上三万张图片，就在测试集上得到了比 Faster R-CNN 好不少的结果。

接下来，Navaneeth Bodla 等人提出的 Soft-NMS^[23]和 Tsung-Yi Lin 等人又提出的 Focal-Loss^[24]都不是在网络结构上做改动。Soft-NMS 改动了目标检测领域一直常用的非极大值抑制(Non-Maximum Suppression, NMS)过程。传统 NMS 过程在删除疑似重复的邻近框时，采用一个比较直接的做法。即，将预测框按置信度从高到低排序后，若有两个框的交并比(Intersection over Union, IoU)大于一个阈值，如 0.5。那么就删除置信度较低的框。删除置信度较低的框，可看作将该预测框的置信度置为 0 的过程。而在 Soft-NMS 中，其置信度就不是被置为 0，而是按某种计算降低其的置信度。比如较低分框原先置信度为 0.8，若它和得分较高框的 IoU 为 0.6，高过之前设的阈值 0.5，那么它的置信度就更新为： $0.8*(1-0.6)=0.32$ ，而不是直接被删除。Focal-Loss 则是在网络的损失函数(Loss)上做改动。在训练目标检测任务时，有大量的样本，即图像区域是能被网络简单判别的。这些样本产生的误差很小，但由于数量多，使得它们支配着训练，使误差较大的难样本没有受到足够重视。在网络收敛后，还是会对一些难训练样本有较大误差。故后来有研究者提出了在线难样本挖掘(Online Hard Example Mining, OHEM)^[25]，这是在网

络训练收敛后，进行微调网络，在这期间计算误差函数时，对一个 Batch 中，所有样本的误差进行排序，只用误差较大的难样本进行训练。而 Focal-Loss，是将这种策略做的更好了。Focal-Loss 是对一个 Batch 中每个样本的误差加权，误差越低的样本，加的权重越低；误差越高的样本，加的权重越高。这样就避免了大量简单的样本支配训练，使得误差高的样本难以得到充分的训练。在他们的论文中，OHEM 可以使准确率提升 2%-3%，Focal-Loss 在 OHEM 的基础上准确率还有提升。

1.3.2 基于卷积神经网络的增量学习

如摘要中所述，神经网络具有灾难性忘记的特点^{[26][27]}。所以不能直观地进行增量学习，本节将介绍前人基于卷积神经网络所做的增量学习工作。

Zhizhong Li 等人^[28]最先借鉴 Caruana 等人^[29]和 Hinton 等人^[30]做模型压缩的方法，为图像分类网络添加新的可识别类别。他们将提出的方法称为 LwF(Learning Without Forgetting)。该方法的内容是：

首先构造一个被训练网络，该网络是已有网络的一份拷贝，但为使该网络能分类新的类别，在该网络的最后一个隐层后面，添加了一个全连接层。该层和其原有的最后一个全连接层呈并行的关系。

在测试阶段，用被训练网络对图像进行分类。其原有的最后一层全连接层负责输出关于旧类别的分类结果，新添加的全连接层负责输出关于新类别的分类结果。

在训练阶段，存在两个网络，一个是已有网络，一个是被训练网络。被训练网络在被训练时，有两个损失函数。一个是分类新类别的损失函数，一个是保留旧类别分类能力的损失函数。后者是让被训练网络在被输入图片时，关于旧类别的输出和已有网络的输出一致。而已有网络没有损失函数，所以其在训练过程中不会被改变。且训练过程只使用新类别的训练图片。

在这样的训练下，被训练网络可在损失很少旧类别准确率的情况下，在新类别上有很高的准确率。

在他们的论文中，同时还列出了另外4种为一个分类网络添加新的可识别类别的方法。这4种方法中的前3种比较直观，如图1.6所示。

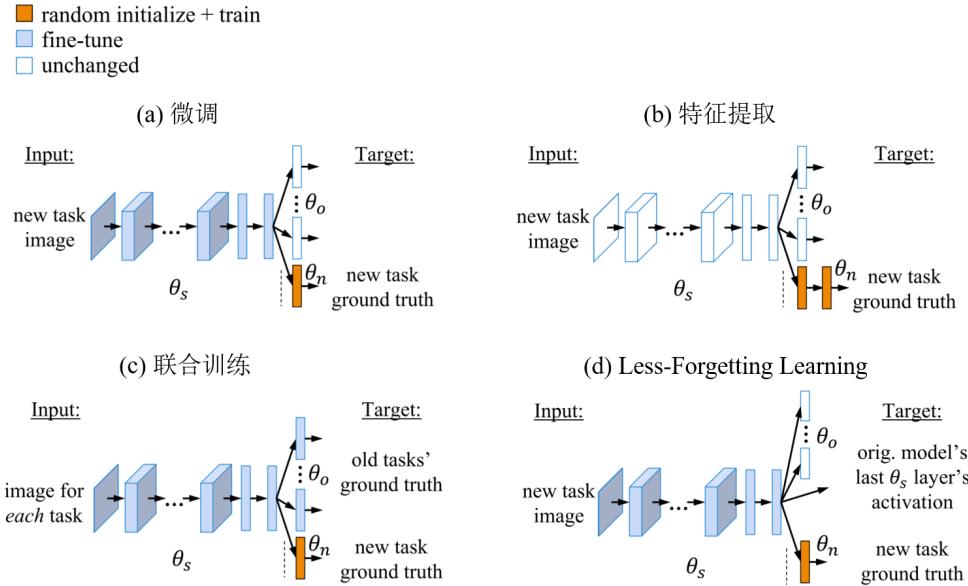


图1.6 四种为分类网络添加新的可识别类别的方法

网络结构上，这4种方法都为新类别添加了专有的输出层（图中橙色的层）。下面分别对这4种方法进行介绍：

(a) 方法是用新类别训练集去微调整个网络。其优点是：新类别分类准确率高。其缺点是：网络关于旧类别的分类能力急剧丧失，即发生了灾难性忘记。

(b) 方法是用新类别训练集只训练新添加的输出层，将网络前部的层看作通用的特征提取层。其优点是：网络关于旧类别的分类能力被完整保留。其缺点是：新类别分类准确率不高。文献[31]即是使用这种方法。

(c) 方法是同时使用新、旧类别的训练集去训练整个网络。其优点是：新、旧类别分类准确率都高。其缺点是：需使用旧类别训练集、训练时间长。

(d) 方法是文献[32]提出的方法。它和LwF不同之处在于，它是在训练时想保持旧类输出层的输入和已有网络一样。根据LwF论文中的比较，它没有LwF在新类别分类任务上的准确率高。

Arun Mallya等人提出了Piggyback方法^[33]。该方法的内容是：为已有网络中

每个参数都训练一个掩模(Mask)，在测试时阶段，掩模的值会是 0 或者 1。使用该网络的每一个任务都有关于所有参数的一套掩模。当在进行一个任务时，若某个参数的掩模值为 1，那么该参数保留为原值。若某个参数的掩模值为 0，那么该参数的值被置为 0。每个任务的一套掩模是反向传播算法进行学习得来的，且在训练时，掩模的值是连续的，在测试时，将掩模的值做了二值化，只使其有 0, 1 两个点的值域。

Francisco M. Castro 等人提出的方法^[34]也是利用蒸馏。但是它会用一种选择机制，在训练旧类别时，从旧类别的训练样本中选择一些典型的代表。在训练新类别时，也会用到旧类别的代表。也就是说，该方法在训练新类别时用到了旧类别的图片。且旧类别的样本代表池有一定的容量大小，当旧类别样本代表池已满的时候，它会按照典型性的排名将排名靠后的旧类别样本剔除出去。也就是说，在后面的继续添加新类别的训练过程中，不会再使用被剔除了的旧类别样本代表。

iCaRL^[35]是 Sylvestre-Alvise Rebuffi 等人提出的用于图像分类的增量学习方法。在训练分类旧类别时，它将 CNN 作为特征提取器，用最近范例均值(Nearest-Mean-of-Exemplars)分类方法和特征向量进行分类。同时，借鉴 herding 算法^[36]从训练样本中挑选范例，将范例存储下来。在添加新的类别时，和训练旧类别不同的地方在于：采用新类分类损失函数和蒸馏损失函数同时对 CNN 进行训练，训练数据是旧类的范例和新类的训练样本。同时，再从新类中挑选出范例，加入范例池。范例池中的范例有优先级权重。且范例池有一定的大小限制，当范例池满的时候，按照优先级移除权重低的范例。

Spyros Gidaris 等人也提出了一种用于图像分类的增量学习方法^[37]。该方法是针对新类别只有很少的样本的情况。在训练分类旧类别时，它将 CNN 最后一层隐层和输出层之间的点乘操作改为了计算余弦相似度，并去掉了该隐层后的 ReLU 层。该方法拥有一个新类别输出层生成器。该生成器拥有可训练的参数。在添加新类别时，将新类别的特征向量、旧类别的输出层输入该生成器，该生成器利用一种 Attention 机制计算得出新类的输出层。有了关于新类别的输出层参数

后，网络即可对新类别进行分类。注意，在添加新类别时，网络的特征提取阶段的参数是不变的。

此外，James Kirkpatrick 等人提出了弹性权重固化 EWC(Elastic Weight Consolidation)方法^{[38][39]}。该方法在训练旧类别时，估计每个参数的重要性权重。在训练新类别时，根据每个参数的重要性权重，有选择地对每个参数进行不同的正则化，来降低重要参数的学习速率，以此来保证旧类别的识别率。

Konstantin Shmelkov 等人的工作^[40]和本文工作的目的一样，即基于卷积神经网络的目标检测，做其增量学习的实现。它是基于 Fast R-CNN 和 LwF 方法做的。它的具体做法是：训练时也是有两个网络，即，将已有网络（训练好的 Fast R-CNN 网络）复制一份作为被训练网络。但被训练网络在最后部，添加了用于识别新类别的层，该层和其原来的最后一层为并行关系。

对于一张训练图像，先用 Edge Boxes^[14]（在 Pascal VOC 数据集上使用此方法）或 MCG 方法^[41]（在 MS-COCO 数据集上使用此方法）得出物体建议框(Object Proposals)。然后将物体建议框、该训练图片同时输入已有网络（训练好的 Fast R-CNN 网络）和被训练网络。被训练网络关于新类别的输出用 Fast R-CNN 定义的方式去训练。同时，将已有网络给出的，背景概率最低的一些物体建议框挑出。将被训练网络关于这些框的输出，以已有网络关于这些框的输出作为回归目标，添加损失函数进行训练，文中默认用的是 L2 损失函数。训练过程也只使用新类别的训练图片。类似于 LwF，这样训练，就使得被训练网络在训练结束时，既拥有了检测新类别物体的能力，同时保留了对旧类别的检测能力。

Linting Guan 等人也提出过关于增量的目标检测的方法^[42]。他们的方法是：不接触训练已有网络用的图片，但是训练新类的图片里面有旧类别。所以它的方法是，将已有网络在新类别训练集中的检测结果当成旧类的真实标注，和训练新类的图片中新类的真实标注，一起在已有网络的拷贝上进行训练（该网络的初始值是已有网络的）。若存在检测得到的高分框和新类别的真实框之间有较大的 IoU，那么即认为这个预测框是错误的，将其从训练新网络用的标注中剔除。该论文的好处是：没有用到训练旧类别使用的图片。它的缺点同样也比较明显，那

就是若已有网络产生的高分框是错误的，且没有被上述的方法成功剔除掉，那么训练新类别用的标注数据就是错误的。同时，若已有网络漏检，那么新网络的训练也将受到不小的影响。

1.4 本文创新点

本文使用两种方法，为一个已有的目标检测网络添加新的可检测类别。

在第一种基于特征提取的方法中，本文的创新点在于：

(1) 实验探究了为新类别添加的专有层数和新类别准确率的关系。发现：为新类别添加的专有层数越多，网络在新类别上的准确率越高，且准确率的增益随添加层数的增多而减少。(2) 实验探究了已有网络可检测类数和其增量能力之间的关系。发现：已有网络可检测类数越多，其增量能力越强，新类别准确率越高。(3) 提出将新类别专有层的添加位置前移，能可观地提升网络在新类别上的准确率。(4) 提出为新类别添加浅层旁路，可提升网络在新类别上的准确率。

在第二种基于保留损失函数的方法中，本文的创新点在于：

(1) 率先实验并证明了，将 LwF 的思路（保留损失函数的方法）引入具有 RPN 模块的目标检测算法中，是有效的。该方法很好地保留了目标检测网络对旧类别的检测能力，同时添加了新的可检测类别。(2) 本文进行了大量实验，来讨论该方法中值得注意的各种细节。本文为每个细节，都挑出了最好的做法。

1.5 本文组织结构

全文共分为六个章节，各章节内容安排如下：

第一章介绍目标检测及其增量学习的概念、研究意义、以及应用领域，并介绍该课题的国内外相关研究现状，最后说明本文的主要工作和创新点。

第二章首先介绍目标检测算法的评价指标。随后，介绍了卷积神经网络、残差网络，以及 Faster R-CNN 算法、FPN 算法、模型蒸馏的原理。

第三章介绍本文使用的数据集，以及对数据集所做的处理。本文共使用了三个数据集，分别为 Pascal VOC、MS-COCO、Open Image Challenge 2018 数据集。本章分别对三个数据集进行介绍，并对本文使用的训练数据、测试数据做详细的

说明。

第四章介绍本文研究的第一个方法，基于特征提取思想的方法。这是一种直观的方法。本文首先在默认设置下得到了该方法的结果，然后设计了四个实验研究该方法的性质，这些实验的目的都是为了提高该方法的准确率。其中后两个实验内容是本文提出的对该方法的改进，从实验结果可看出，这两个改进成功地提高了该方法的准确率。

第五章介绍本文研究的第二个方法（本文提出），基于保留损失函数的方法。本文将 Zhizhong Li 等人在图像分类上提出的 LwF 方法应用到目标检测上，从而得出了该方法。本章首先介绍该方法的内容，其次介绍该方法的实现细节，最后本文对该方法内的五个细节进行了详细实验，为每个细节选出了最好的做法。

第六章总结与展望。该章对本文内容进行归纳总结，并且提出了未来的可研究方向。

第2章 理论基础及相关技术

2.1 目标检测算法的评价指标

多类物体目标检测算法的评价指标是 mAP(mean Average Precision)。它是所有类别 AP(Average Precision)的平均值。对于其中一类物体，AP 是该类 PR 曲线(Precision Recall Curve)下的面积。对于一类物体的 AP，它的计算过程如下：

将所有测试图片中的预测框按被判断为该类的概率排序，形成一个队列。从队首（第一个框），向后遍历。若一个预测框，与一个该类真实目标的交并比(Intersection over Union, IoU)大于一定阈值（如 0.5），且该真实目标未被前面更高分的预测框召回过，那么判定该预测框是正确的，它将该真实目标召回了。这时，需计算在此召回率下的准确率。该准确率的计算为：设在该框前面（队首方向）加上该预测框，有 N 个预测框，假设正确的框的数量是 N_true，那么该准确率就是 N_{true}/N 。就这样，每出现一个正确的预测框（等价于被召回的真实目标增加一个），就计算一下此时的召回率、准确率，直到遍历完队列，或成功召回了该类所有的真目标。这之后，先对 PR 曲线进行变形，将 PR 曲线上每个点的准确率值，更改为其右边所有点最大的准确率值。在 Pascal VOC 第一版的评价算法中，在变形后的 PR 曲线上取召回率为[0:0.1:1]这 11 点的准确率做平均，得出的结果作为 AP 值。在 Pascal VOC 第二版的评价算法中，将变形后的 PR 曲线下的面积作为 AP 值。在 MS-COCO 的评价算法中，在变形后的 PR 曲线上取召回率为[0:0.01:1]这 101 点的准确率做平均，得出的结果作为 AP 值。

值得注意的是，在目标检测算法的评价算法中，会限制单张图片预测框的数量，以使算法间进行公平地比较。在 MS-COCO 数据集的评价算法中，对每张图片，目标检测算法最多只能给出 100 个预测框。

另外值得注意的是，在 MS-COCO 数据集中，它的评价指标有特殊的地方。它存在一个 mAP@0.5:0.95 的评价指标。它将本文之前叙述的算法称作 mAP@0.5，因为之前算法评判预测框正确与否，是使用 IoU 为 0.5 的阈值。推理可知，mAP@0.95，是使用 IoU 为 0.95 的阈值。mAP@0.5:0.95 的意思是在 0.5 到 0.95

的闭区间内，每隔 0.05 就作为一个 IoU 阈值，计算一次 mAP。将这 10 个 mAP 取平均后，记作 mAP@0.5:0.95。它相较单一的 mAP@0.5 评价指标，更鼓励算法将真实目标框得准确。框得越准确的算法，将会有越高的 mAP@0.5:0.95。

2.2 卷积神经网络原理

2.2.1 卷积层

在一般的卷积神经网络中，一个卷积层由多个卷积核组成。每个卷积核的空间维度相等（一般取 1×1 到 11×11 之间的某个配置），通道数也相等且等于输入数据的通道数。

若将卷积层计算过程写成一种比较简单的表达，其中 a^{l-1} 为该卷积层的输入。 W^l 为该层卷积核的权重， b^l 为偏置， σ 为激活函数， a^l 为该卷积层的输出，* 代表卷积层中的卷积过程。则该卷积层的前向传播计算可表示为：

$$a^l = \sigma(z^l) = \sigma(a^{l-1} * W^l + b^l) \quad (2.1)$$

对于反向传播，若 δ^l 是误差对该层输出，且未经过激活函数的结果 z^l 的导数，即：

$$\delta^l = \frac{\partial L(W, b, x)}{\partial z^l} \quad (2.2)$$

则，误差对该层卷积核的权重 W^l 的导数为：

$$\frac{\partial L(W, b, x)}{\partial W^l} = \frac{\partial L(W, b, x)}{\partial z^l} \frac{\partial z^l}{\partial W^l} = a^{l-1} * \delta^l \quad (2.3)$$

误差对该层偏置 b^l 的导数为：

$$\frac{\partial L(W, b, x)}{\partial b^l} = \sum_{u,v} (\delta^l)_{u,v} \quad (2.4)$$

为了求误差对该层之前层的参数的导数，需计算出误差对该层前一层未经激活函数时的数据的导数，该过程计算公式为：

$$\delta^{l-1} = \delta^l \frac{\partial z^l}{\partial z^{l-1}} = \delta^l * \text{rot180}(W^l) \odot \sigma'(z^{l-1}) \quad (2.5)$$

其中 \odot 表示 Hadamard 积，表示将两个同维矩阵或两个同维向量，对应的元素进行相乘的运算。 $\text{rot180}()$ 表示将括号中的矩阵顺时针或逆时针旋转 180 度。 $*$ 仍代表卷积层中的卷积过程。

2.2.2 激活层

激活层是将数据进行非线性变换的过程。常用的激活函数有 Sigmoid 函数、双曲正切函数、线性整流函数(ReLU)^[43]。下面分别进行介绍。

(1) Sigmoid 函数

$$\delta(z) = \frac{1}{1 + \exp(-z)} \quad (2.6)$$

因变量关于自变量求导：

$$\delta'(z) = \delta(z)(1 - \delta(z)) \quad (2.7)$$

(2) 双曲正切函数

$$\delta(z) = \tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \quad (2.8)$$

因变量关于自变量求导：

$$\delta'(z) = 1 - \tanh^2(z) \quad (2.9)$$

(3) 线性整流函数

$$\delta(z) = \max(0, z) \quad (2.10)$$

因变量关于自变量求导：

$$\delta'(z) = \begin{cases} 0, & z \leq 0 \\ 1, & z > 0 \end{cases} \quad (2.11)$$

2.2.3 池化层

池化层是降低数据空间维度的过程。池化层一般分为 Max Pooling 和 Average Pooling。Max Pooling 是将一个特征图中临近的几个像素的最大值输出。例如在 2×2 的 Max Pooling 中，是将一个 2×2 区域的像素的最大值作为该区域的下采样

值。而 Average Pooling 是将邻近区域的平均值输出，作为该区域的下采样值。池化层的前向传播用公式可表示为：

$$a^l = \text{subsample}(a^{l-1}) \quad (2.12)$$

其中 subsample 表示 Max Pooling 的下采样逻辑，或 Average Pooling 的下采样逻辑。

对于反向传播，池化层的反向传播可用如下公式表示：

$$\delta^{l-1} = \text{upsample}(\delta^l) \odot \sigma'(z^{l-1}) \quad (2.13)$$

其中 δ^l 为误差对该池化层激活后的输出值的导数， σ 为激活函数， z^{l-1} 为经过激活层前，该池化层的输出。 upsample 表示 Max Pooling 的反向逻辑，或 Average Pooling 的反向逻辑。

2.2.4 全连接层

卷积神经网络的全连接层和多层感知机中的定义相同，用公式可以如下表示：

$$a^l = \sigma(z^l) = \sigma(W^l a^{l-1} + b^l) \quad (2.14)$$

其中 a^{l-1} 为该全连接层的输入， W^l 为该全连接层的权重， b^l 为偏置， σ 为激活函数， a^l 为该全连接层的输出。 $W^l a^{l-1}$ 间的乘积过程为矩阵和向量相乘。

对于反向传播，若 δ^l 是误差对该全连接层输出，且还未经过激活层的结果 z^l 的导数，即：

$$\delta^l = \frac{\partial L(W, b, x)}{\partial z^l} \quad (2.15)$$

则，误差对该层权重 W^l 的导数为：

$$\frac{\partial L(W, b, x)}{\partial W^l} = \frac{\partial L(W, b, x)}{\partial z^l} \frac{\partial z^l}{\partial W^l} = \delta^l (a^{l-1})^T \quad (2.16)$$

误差对该层偏置的导数为：

$$\frac{\partial L(W, b, x)}{\partial b^l} = \frac{\partial L(W, b, x)}{\partial z^l} \frac{\partial z^l}{\partial b^l} = \delta^l \quad (2.17)$$

为了求误差对该层之前层的参数的导数，需计算出误差对该层前一层未经激活函

数时的数据的导数，该过程计算公式为：

$$\delta^{l-1} = \delta^l \frac{\partial z^l}{\partial z^{l-1}} = (W^l)^T \delta^l \odot \sigma'(z^{l-1}) \quad (2.18)$$

2.2.5 Softmax 层

Softmax 层实现了对一列向量的归一化，同时使得较大的值，相对于较小的值差距拉的更大。Softmax 层的计算公式如下。

$$a_i^l = \frac{e^{a_i^{l-1}}}{\sum_i e^{a_i^{l-1}}} \quad (2.19)$$

2.2.6 Batch Normalization 层

Batch Normalization 层^[44]是将特征图上，通道序号相同的元素作为一组，对同组的数据做归一化。通常 Batch Normalization 层的输入数据有两种情况。一种是卷积层输出的特征图，记为 $a^{l-1} \in \mathbb{R}^{N \times C \times H \times W}$ 。另一种是全连接层输出的特征图，记为 $a^{l-1} \in \mathbb{R}^{N \times C}$ 。

首先对其输入为 $a^{l-1} \in \mathbb{R}^{N \times C}$ 的情况进行说明，在该情况下，它的前向传播过程为：

首先求 μ 和 δ ：

$$\mu_j = \frac{1}{N} \sum_{i=1}^N a_{i,j}^{l-1} \quad (j=1, \dots, C) \quad (2.20)$$

$$\delta_j^2 = \frac{1}{N} \sum_{i=1}^N (a_{i,j}^{l-1} - \mu_j)^2 \quad (j=1, \dots, C) \quad (2.21)$$

在得到 μ 和 δ 后，对特征图中的数值进行归一化：

$$\hat{x}_{i,j} = \frac{a_{i,j}^{l-1} - \mu_j}{\sqrt{\delta_j^2 + \epsilon}} \quad (j=1, \dots, C, i=1, \dots, N) \quad (2.22)$$

$$a_{i,j}^l = \gamma_j \hat{x}_{i,j} + \beta_j \quad (j=1, \dots, C, i=1, \dots, N) \quad (2.23)$$

得到的 $a^l \in \mathbb{R}^{N \times C}$ ，即为 Batch Normalization 层输出的特征图。

对于反向传播，此时 $\frac{\partial J}{\partial a^l} \in \mathbb{R}^{N \times C}$ 为已知量。Batch Normalization 层的反向传播

过程为：

首先求得：

$$\frac{\partial J}{\partial \hat{x}_{i,j}} = \frac{\partial J}{\partial a_{i,j}^l} \cdot \gamma_j \quad (j=1, \dots, C, i=1, \dots, N) \quad (2.24)$$

$$\begin{aligned} \frac{\partial J}{\partial \delta_j^2} &= \sum_{i=1}^M \frac{\partial J}{\partial \hat{x}_{i,j}} \cdot (a_{i,j}^{l-1} - \mu_j) \cdot \frac{-1}{2} (\delta_j^2 + \epsilon)^{-3/2} \\ &\quad (j=1, \dots, C, i=1, \dots, N) \end{aligned} \quad (2.25)$$

$$\begin{aligned} \frac{\partial J}{\partial \mu_j} &= \left(\sum_{i=1}^M \frac{\partial J}{\partial \hat{x}_{i,j}} \cdot \frac{-1}{\sqrt{\delta_j^2 + \epsilon}} \right) + \frac{\partial J}{\partial \delta_j^2} \cdot \frac{\sum_{i=1}^M -2(a_{i,j}^{l-1} - \mu_j)}{M} \\ &\quad (j=1, \dots, C, i=1, \dots, N) \end{aligned} \quad (2.26)$$

然后可求得：

$$\begin{aligned} \frac{\partial J}{\partial a_{i,j}^{l-1}} &= \frac{\partial J}{\partial \hat{x}_{i,j}} \cdot \frac{1}{\sqrt{\delta_j^2 + \epsilon}} + \frac{\partial J}{\partial \delta_j^2} \cdot \frac{2(a_{i,j}^{l-1} - \mu_j)}{M} + \frac{\partial J}{\partial \mu_j} \cdot \frac{1}{M} \\ &\quad (j=1, \dots, C, i=1, \dots, N) \end{aligned} \quad (2.27)$$

$$\frac{\partial J}{\partial \gamma_j} = \sum_{i=1}^M \frac{\partial J}{\partial a_{i,j}^l} \cdot \hat{x}_{i,j} \quad (j=1, \dots, C, i=1, \dots, N) \quad (2.28)$$

$$\frac{\partial J}{\partial \beta_j} = \sum_{i=1}^M \frac{\partial J}{\partial a_{i,j}^l} \quad (j=1, \dots, C, i=1, \dots, N) \quad (2.29)$$

得到的 $\frac{\partial J}{\partial a^{l-1}} \in \mathbb{R}^{N \times C}$ 即为误差对该层输入数据 a^{l-1} 的导数，用来将梯度继续反向传

播。得到的 $\frac{\partial J}{\partial \gamma}$ 为误差对 γ 的导数，用来更新 γ 的值。 $\frac{\partial J}{\partial \beta}$ 为误差对 β 的导数，用
来更新 β 的值。

当 Batch Normalization 层的输入为 $a^{l-1} \in \mathbb{R}^{N \times C \times H \times W}$ 时，为了保留卷积层的参数
共享特性，也是将通道序号一样的数据，求均值、方差，进而归一化。所以也会

有 $\mu \in \mathbb{R}^C$, $\delta^2 \in \mathbb{R}^C$ 。同样, 有 $\gamma \in \mathbb{R}^C$ 及 $\beta \in \mathbb{R}^C$ 。根据这样的规则, 即可推出其前向传播、反向传播的计算公式, 这里就不再详细描述了。

2.3 残差网络

自从 2012 年 AlexNet 在 ILSVRC 大赛获得第一名以来, 人们发现卷积神经网络的深度, 对它的图像分类准确率有着至关重要的影响: 越深的网络, 能得到越高的准确率^{[45][46]}。但随着深度再增加, 其在图像分类上的准确率逐渐不变, 并开始出现下降的现象。并且准确率下降的原因不是过拟合, 而是在训练集上就产生比浅层网络更大的训练集误差^{[47][48][49]}, 人们称这种现象为退化现象^[47]。

Kaiming He 和他的团队试图去解决深度网络退化现象的问题, 从而获得更高的图像分类准确率, 因而提出了深度残差网络(Deep Residual Network, ResNet)^[47]。ResNet 和普通网络的区别是, 它拥有跳跃连接(Shortcut)^{[50][51][52]}, 并且加在 ResNet 中的跳跃连接, 是将某层产生的特征图和其前几层产生的特征图直接相加。从一个跳跃连接的起始位置, 到它的结束位置, 被称作一个残差模块, 它的网络结构如图 2.1 所示。在残差模块中, 左边第一个卷积层是为了将特征图的通道数降低, 然后将通道数降低了的特征图输入中间核大小为 3x3 的卷积层, 最后一个卷积层是为了将特征图的通道数恢复为, 输入该残差模块的特征图的通道数。这样做好处是: 使得输入核大小为 3x3 的卷积层的特征图的通道数减少, 减少了该卷积层参数的数量、计算量, 同时增加了网络的深度。

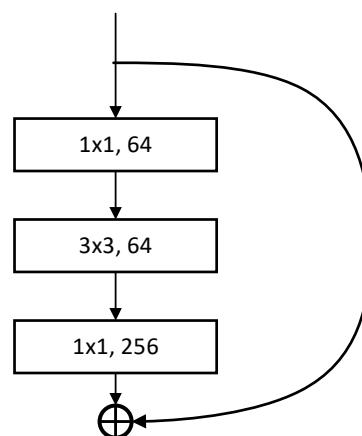


图 2.1 残差网络中的第一类残差模块

在 ResNet 中，除第一层卷积层以外，之后所有的每几层卷积层，都拥有跳跃连接。当需将特征图降低空间维度，提升通道数时，跳跃连接上会存在一个步长为 2，卷积核大小为 1 的卷积层，用于使相加的两个特征图有同样的空间维度、通道数，本文把拥有这样跳跃连接的模块，称为第二类残差模块，它的结构如图 2.2 所示。

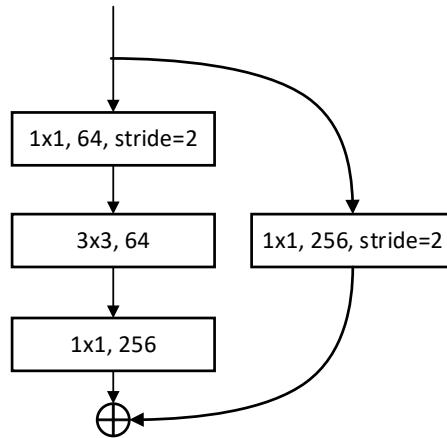


图 2.2 残差网络中的第二类残差模块

另外，在 ResNet 中，所有卷积层后都接有一个 Batch Normalization 层。且在每个残差模块内，除最后一个 Batch Normalization 层外，其他 Batch Normalization 层后均接有一个 ReLU 激活层。

在使用跳跃连接(Shortcut)后，Kaiming He 和他的队友将 CNN 构建到了 152 层，还有更深的版本，1000 层。一般常用的残差网络为 50 层、101 层和 152 层的版本。本实验为了节约实验时间，选用了 50 层的版本。

50 层的残差网络，被称作 ResNet-50。它由一个底座和 4 个阶段组成。每个阶段包含若干个残差模块。底座是由一个卷积层（卷积核大小为 7x7，步长为 2），一个 Batch Normalization 层，和一个 ReLU 层所构成。之后，从第 1 个阶段到第 5 个阶段分别包含：3, 4, 6, 3 个残差模块。如前文所述，残差模块分两类：第一类如图 2.1 所示，第二类如图 2.2 所示。第二类残差模块是用于调整维度的，在该残差模块中，左边第一个卷积层和右边的卷积层均是步长为 2，它会将输入的特征图下采样到原来空间大小的一半。ResNet 中，每个阶段的第一个残差模块为

第二类残差模块，随后所有残差模块为第一类残差模块。所以，除去每个阶段最开始输入的特征图外，在一个阶段中，特征图的空间维度是不变的。每个阶段的第一个残差模块的左边第一个卷积层，和右边的卷积层，负责将输入的特征图降维，后续该阶段内的特征图空间维度都相同。

2.4 Faster R-CNN 目标检测算法

Faster R-CNN 算法是一种两阶段(Two-stage)的目标检测算法。同其他两阶段目标检测算法一样，第一阶段是为了判断图像中各个区域有无物体，第二个阶段是为了判断每个可能有物体的区域具体是什么类别的物体。如第一章中国内外研究现状所述，它是基于 Fast R-CNN 得出来的。Faster R-CNN 的两个阶段第一个被称为区域建议网络(Region Proposal Network, RPN)阶段，第二个被叫做 Fast R-CNN 阶段，本文亦称之为分类(Classification, CLS)阶段。RPN 阶段和 CLS 阶段共享了网络主干上的所有卷积层。Faster R-CNN 的网络结构图，如图 2.3 所示。

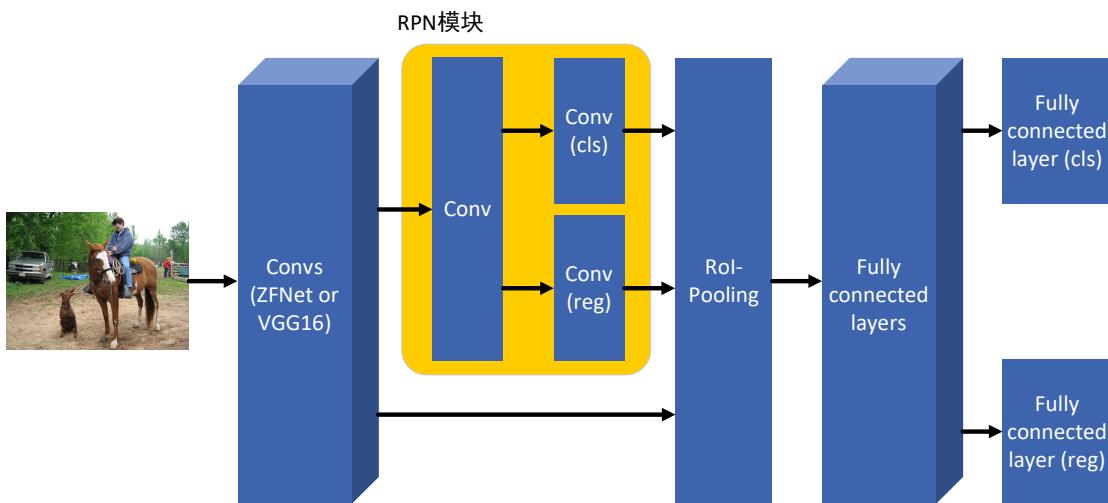


图 2.3 Faster R-CNN 网络结构图

如图 2.3 所示，图片从最左端输入网络，经过若干卷积层、激活层、池化层处理后，得到的特征图传给了两个模块，本文分别称它们为 RPN 模块和 CLS 模块。CLS 模块同时需要 RPN 的输出作为输入，所以 RPN 模块先于 CLS 模块执行。

RPN 模块由三个层组成，分别是：RPN 隐层、用于分类的卷积层、用于物体框回归的卷积层，如图 2.4 所示。

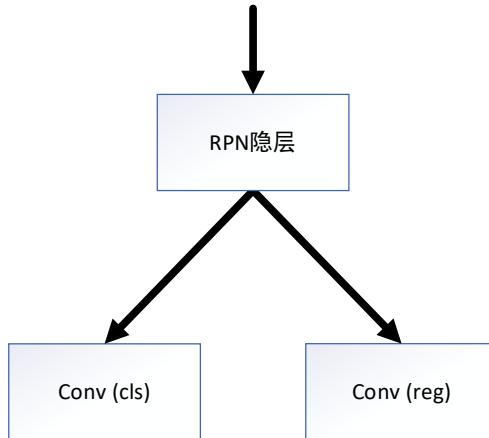


图 2.4 RPN 模块

RPN 模块中，用于分类的卷积层进行的是二分类过程，其任务是给出预先定义的每个锚点(anchor)是物体，还是背景的概率。其使用的损失函数为 softmax 交叉熵损失函数(SoftmaxCrossEntropy)。关于 anchor 的定义，如下图 2.5 所示：

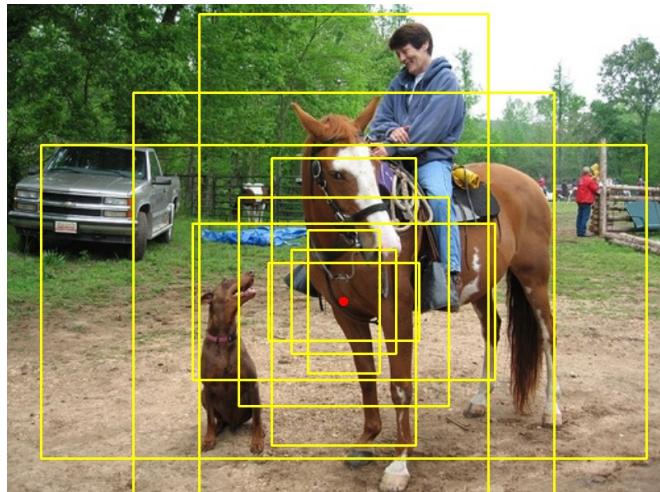


图 2.5 anchor 的示意图

关于图 2.5，假设图中的红点，是 RPN 隐层输出的特征图上一个像素的感受野的中心。那么在其周围的 9 个黄色矩形框，就是预先定义的，以这个像素感受野中心为中心的 9 个 anchor。那么，对于一张 40x40 的特征图来说，就存在 $40 \times 40 \times 9 = 14400$ 个 anchor 待被判断。RPN 模块中，用于物体框回归的卷积层，对于每个 anchor 会输出 4 个参数。Anchor 通过使用这 4 个参数做变换，会和真实物体框更加的重合。所以，在训练时，这 4 个参数的真实值为式(2.30)所定义。

$$\begin{aligned} t_x &= (x - x_a) / w_a, t_y = (y - y_a) / h_a, \\ t_w &= \log(w / w_a), t_h = \log(h / h_a) \end{aligned} \quad (2.30)$$

其中 x 为分配给这个 anchor 的真实物体框中心点的横坐标, y 为其中心点纵坐标, w 为其宽, h 为其高。 x_a 为这个 anchor 的中心点的横坐标, y_a 为其中心点纵坐标, w_a 为其宽, h_a 为其高。 t_x , t_y , t_w , t_h 即为 4 个参数。在测试时, 得到这 4 个参数后, 并且已知 anchor 的坐标, 即可求出该位置物体的预测框。做物体框回归的卷积层, 后面连接的是 SmoothL1 损失函数。

SmoothL1 损失函数可以看作 Huber 损失函数的变种, 它的计算公式为:

$$SmoothL1(x) = \begin{cases} \frac{x^2}{2\beta} & |x| < \beta \\ |x| - \frac{\beta}{2} & \text{其他情况} \end{cases} \quad (2.31)$$

SmoothL1 损失函数最早由 Ross Girshick 在 Fast R-CNN 中提出, 当时并不包含 β 因子, 后被拓展为包含了 β 因子。式中, x 是预测值和真实值之间的差值。 β 是一个人为指定的因子, 该 loss 函数在 x 小于 β 时是一个二次函数, 等同于 L2 损失函数, 在 x 大于 β 时, 是一个一次函数, 等同于 L1 损失函数。默认情况下, β 取 1。SmoothL1 损失函数在 x 值较大时, 不会像 L2 损失函数那样产生过大的梯度, 避免梯度爆炸, 使得训练更稳定; 在 x 值较小时, 会产生较小的梯度, 不会像 L1 损失函数那样使预测值在真实值左右震荡。

利用 RPN 模块中分类卷积层、物体框回归卷积层的输出, 我们可得到若干区域建议框/感兴趣区域(Region of Interest, RoI)。这些 RoI 之间可能有很高的重叠度, 即矩形框的交并比(IoU)很大。为了剔除重叠度很高的矩形框, 在这里执行了一个非极大值抑制(NMS)。

NMS 的过程是: 将所有 RoI, 按是物体的概率从高到低进行排序, 形成一个队列, 先将第一个 RoI 挑出, 往后遍历所有的框和第一个框计算 IoU, 若某个框和第一个框的 IoU 大过一个阈值, 如 0.5, 则将该框剔除。接下来, 从更新过的队列中挑出第二个框进行上述的重复操作, 直到要挑出的框是更新过的队列中最

后一个框，则 NMS 过程结束。

进行过 NMS 后，将还剩下的 RoI 在输入 RPN 模块的特征图中对应的区域，从特征图中截取出来，处理成空间分辨率一样（例如 7×7 ）的一个个特征子图。这个处理的过程，被称为 RoIPooling。RoIPooling 的过程为：对于一个 RoI，该过程的输入是特征图、该 RoI 的坐标。该过程的输出是一个空间分辨率为 7×7 的特征图。它是将特征图上与 RoI 对应的区域，均分为 7×7 个像素的过程，每个像素的通道数不变。

将这些特征子图每个都展平(Flatten)成一列向量，送入 CLS 模块。CLS 模块包含 4 个全连接层，其中有一个用于分类的全连接层，一个用于物体框回归的全连接层，还有两个是全连接隐层，两个隐层后面都有 ReLU 激活层，如图 2.6 所示。

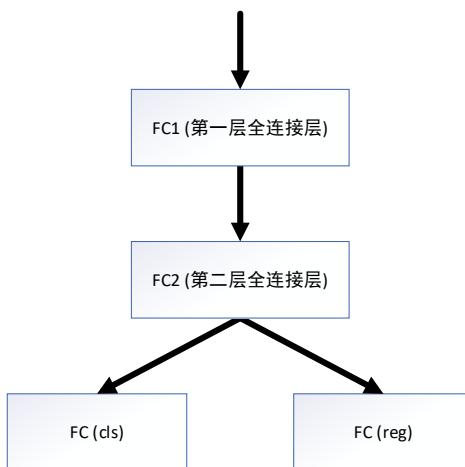


图 2.6 CLS 模块

用于分类的全连接层，将判断每个特征子图到底是什么类别的物体；用于物体框回归的全连接层将得出物体框的变换参数。同 RPN 模块一样，分类的损失函数使用的是柔性最大值交叉熵损失函数。为了回归物体框的变换参数，使用的是 SmoothL1 损失函数。

值得注意的是，每个类别都有 4 个物体框变换参数。所以，若有 80 类物体（加上背景为 81 类物体），当输入一个特征子图时，用于回归的全连接层将输出 $81 \times 4 = 324$ 个浮点数。之后，将 CLS 模块得出的关于每个特征子图的物体类别

概率、物体框变换参数，与 RoI 的坐标值进行结合，即可求出一个个物体框，及其是每类物体的概率。

2.5 特征金字塔网络目标检测算法

特征金字塔网络(Feature Pyramid Network, FPN)是基于 Faster R-CNN 所作的工作。FPN 将深层的特征图上采样，和浅层的特征图相加后，在相加的特征图上对每个位置进行有无物体的判别；而后将相加的结果再上采样，再和分辨率更高的浅层特征图相加，再在这个结果特征图上对每个位置进行有无物体的判别，如此重复下去。FPN 和 Faster R-CNN 一样，同样是两阶段(Two-Stage)的目标检测算法。

FPN 论文中使用的是 ResNet 作为主干网。它在多个分辨率不同的特征图上对原图每个位置做有无物体的判别。即，它有多个连接 RPN 模块的特征图。下面以 ResNet-50 为主干网，说明这些特征图的来源。首先 FPN 在 ResNet 上添加了侧面连接层，为卷积层。对于 ResNet-50 来讲，侧面连接层有 4 个。它们的输入分别是 ResNet-50 第 5 阶段最后输出的特征图，记为 res5_2_sum；ResNet-50 第 4 阶段最后输出的特征图，记为 res4_5_sum；ResNet-50 第 3 阶段最后输出的特征图，记为 res3_3_sum；ResNet-50 第 2 阶段最后输出的特征图，记为 res2_2_sum。将这些特征图经侧面连接层处理后的输出分别记为：fpn_inner_res5_2_sum、res4_5_sum_lateral、res3_3_sum_lateral、res2_2_sum_lateral。将 fpn_inner_res5_2_sum 上采样为原来长、宽的 2 倍（上采样方法为双线性插值）后，和 res4_5_sum_lateral 按元素进行相加(element wise addition)，记结果为 fpn_inner_res4_5_sum。再将 fpn_inner_res4_5_sum 上采样后和 res3_3_sum_lateral 相加得出 fpn_inner_res3_3_sum。以同样方式得出 fpn_inner_res2_2_sum 后，以 fpn_inner 为开头命名的四个特征图，分别再经过一个卷积层，得出 fpn_res5_2_sum、fpn_res4_5_sum、fpn_res3_3_sum、fpn_res2_2_sum，这四个特征图即是分别输入四个 RPN 模块的特征图。

RPN 模块：FPN 中的 RPN 模块和 Faster R-CNN 中几乎一样，只是在其 RPN

模块中使用的是 Sigmoid 交叉熵损失函数。另外，FPN 中多个 RPN 模块的参数是共享的，即相同的。

CLS 模块：FPN 中的 CLS 模块也和 Faster R-CNN 中几乎一样，只是隐层神经元的数目不同。

FPN 的网络结构，如图 2.7 所示。

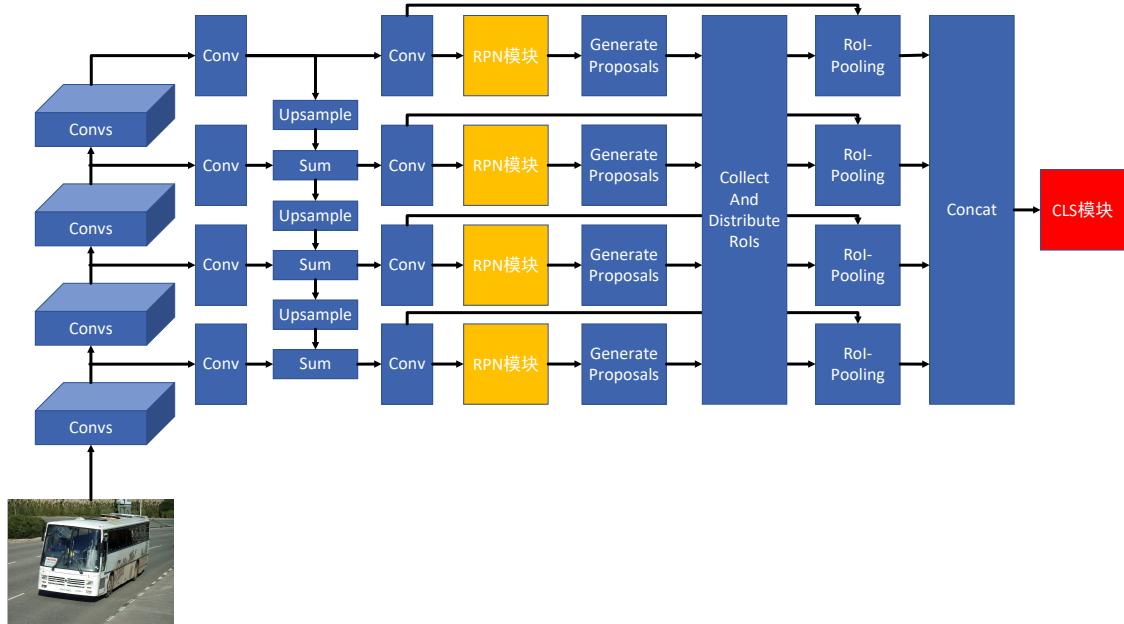


图 2.7 FPN 网络结构图

测试过程：将一张图像从网络头部输入网络。将所有 RPN 模块得出的 ROI（得出 ROI 的方法如介绍 Faster R-CNN 中所描述），统一起来，进行分组。分组的方法如公式(2.32)所示。

$$k = \left\lfloor k_0 + \log_2(\sqrt{wh} / 224) \right\rfloor \quad (2.32)$$

每个 ROI 组和一个输入 RPN 模块的特征图相对应。对于一对，输入 RPN 模块的特征图，和与其对应的 ROI 组，将它们输入一个 RoIPooling 层中，进行特征子图的提取（RoIPooling 的过程如介绍 Faster R-CNN 中所述）。将所有 RoIPooling 层的输出，堆叠(Concat)起来，输入 CLS 模块。得到 CLS 模块的输出后，按介绍 Faster R-CNN 中描述的操作，就得到了关于该测试图片的所有检测框。

训练过程：在所有 RPN 模块中，在 cls 层后部添加 Sigmoid 交叉熵损失函数，

在 reg 层后部添加 SmoothL1 损失函数;在 CLS 模块中,在 cls 层后部添加 Softmax 交叉熵损失函数, 在 reg 层后部添加 SmoothL1 损失函数。这些损失函数真实标签的得来: 对于 RPN 模块的真实标签, 对于不同的 RPN 模块, 由于它们接在不同大小的特征图后。所以它们负责判断不同大小的区域。所以不同 RPN 模块使用的真实标签不一样。另外, 在一个 RPN 模块中, anchor 只有 1 种大小, 但是有 3 种长宽比。对于 CLS 模块的真实标签, 和 Faster R-CNN 中的得来方式一样。

2.6 模型蒸馏

模型蒸馏^[30], 是 Hinton 等人在 2015 年提出的, 是为了改进 Caruana 等人^[29]的工作。Caruana 等人在 2006 年发现, 将一个繁重的、集成的机器学习模型, 所学到的知识压缩进一个较简单的单模型中, 是可以实现的^[30]。Caruana 等人为了实现这个目的, 是让被训练的小模型对于数据产生的输出, 尽可能和, 训练好的繁重的、集成的机器学习模型产生的输出相同。具体来讲, 是类似于让两者产生的输入 Softmax 的值(logits)尽可能相同, 定义小模型训练用的损失函数为两者 logits 的均方误差。

Hinton 等人提出, 在以这种方式做模型压缩时, 可将 Softmax 函数的表达式添加一个温度因子(Temperature, T)。添加温度因子后的 Softmax 函数如式(2.33)所示。

$$q_i = \frac{\exp(z_i / T)}{\sum_j \exp(z_j / T)} \quad (2.33)$$

式中, T 即为温度因子, z 为输入 Softmax 函数的 logits。q 为 Softmax 函数输出的概率。T 可以调控 Softmax 输出的概率的分布。当 T 取较小值时, Softmax 输出的概率值, 元素之间差异较大, 即概率较大的值和概率较小的值相差较大; 当 T 取较大的值时, Softmax 输出的概率值, 元素之间差异较小, 即概率较大的值和概率较小的值相差较小。

Hinton 等人提出让小模型产生的添加温度因子的 Softmax 输出和已训练好的繁重的、集成的机器学习模型产生的添加温度因子的 Softmax 输出, 组成交叉熵

损失函数，来训练小模型。这样一来，该误差对小模型 logits 的导数为：

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right) \quad (2.34)$$

并且证明了，在 T 取较大值时，且当小模型和大模型输出的 logits 都是均值为 0 时，误差对小模型 logits 的导数，等于 Caruana 等人用 logits 组成均方误差损失函数时，误差对小模型 logits 的导数。即，当温度因子较大时，式(2.34)可近似为：

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1+z_i/T}{N + \sum_j z_j/T} - \frac{1+v_i/T}{N + \sum_j v_j/T}\right) \quad (2.35)$$

此时，当 $\sum_j z_j = \sum_j v_j = 0$ 时，式(2.35)可写为：

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}(z_i - v_i) \quad (2.36)$$

所以，当温度因子较大，且当小模型和大模型输出的 logits 都是均值为 0 时，蒸馏等价于去最小化 $1/2(z_i - v_i)^2$ ，即小模型和大模型 logits 的均方误差。说明了，Caruana 等人使用的最小化 logits 的均方误差的方法，可以作为模型蒸馏的一个特殊情况。

第3章 实验数据的准备

3.1 本文使用的数据集

在本文中，共使用了 3 个数据集，它们分别是：Pascal VOC 数据集^[6]、MS-COCO 数据集^[7]、和 Open Image Challenge 2018 数据集。下面分别进行简单介绍。

3.1.1 Pascal VOC 数据集

Pascal VOC 数据集是目前目标检测、图像语义分割领域最具权威和标准性的数据库之一，也是最为复杂，最具挑战性的数据库之一。因此，常被这两个研究领域的学者们用来作为评判算法好坏的数据集。

该数据集在目标检测任务上包含 20 种物体类别。这 20 类物体分别是飞机、自行车、鸟、船舶、瓶子、公共汽车、小轿车、猫、椅子、牛、餐桌、狗、马、摩托车、人、盆栽、羊、沙发、火车、显示器。

该数据集由 Pascal VOC 2007 子集和 Pascal VOC 2012 子集组成。每个子集均包含训练集、验证集、测试集。且该数据集中的图片并不全用于目标检测任务。

Pascal VOC 2007 数据集中：训练集、验证集共有 5011 张图片。用于目标检测训练集的图片有 2501 张，用于目标检测验证集的图片有 2510 张。测试集有 4952 张图片，用于目标检测测试集的图片为所有测试集图片。

Pascal VOC 2012 数据集中：训练集、验证集共有 17125 张图片。用于目标检测训练集的图片有 5717 张，用于目标检测验证集的图片有 5823 张。测试集有 16135 张图片，用于目标检测测试集的图片有 10991 张。

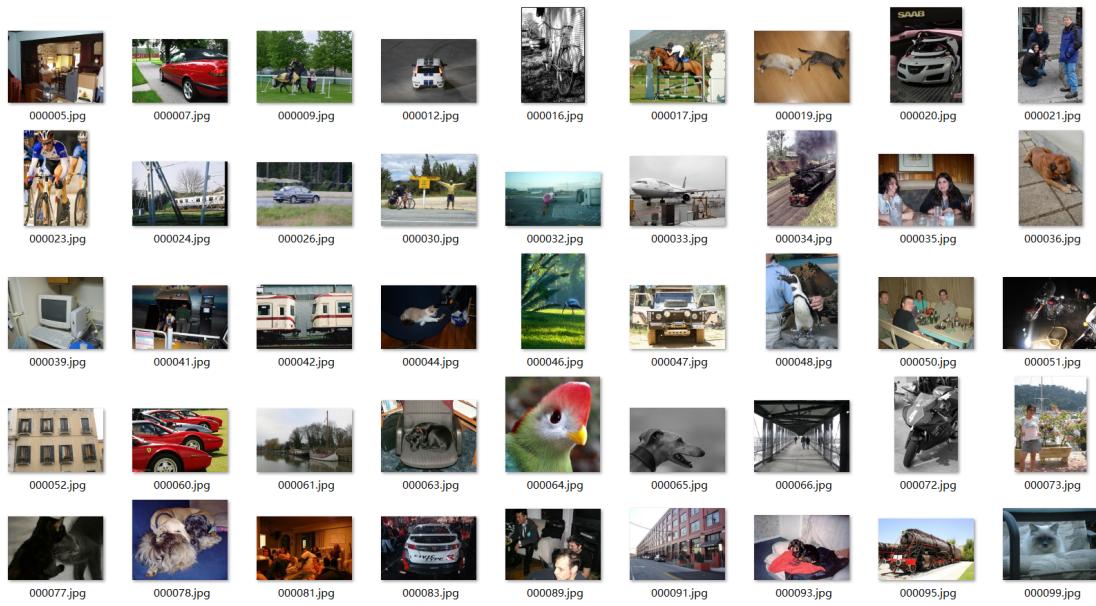


图 3.1 Pascal VOC 数据集中的图片

3.1.2 MS-COCO 数据集

MS-COCO 是由微软公司和一些大学联合制作的数据集，在 2014 年制作完成。和 Pascal VOC 数据集一样，也是举办比赛公开出来的数据集。其包含关于 80 种物体的标注。它包含 Pascal VOC 中的所有物体类别，此外，涵盖蔬菜、水果、随身行李、居家物品等物体类别。

该数据也是不只用来进行目标检测任务。其中用于目标检测训练集的图片有 82783 张，用于目标检测验证集的图片有 40504 张，由于用于目标检测测试集的标注没有公开，所以用于目标检测测试集的图片数目是不知道的。

另外，在后来学者的研究过程中发现，验证集的图片数目没有必要太多。于是 Ross Girshick 将验证集缩减为了 5000 张，将其余的验证集图片加入了训练集。MS-COCO 官方在 2017 年又公布了一次数据集，同意了这种做法，将验证集的数目缩减为了 5000 张。



图 3.2 MS-COCO 数据集中的图片

3.1.3 Open Image Challenge 2018 数据集

Open Image Challenge 2018 是谷歌公司和 ECCV 计算机视觉会议，在 Kaggle 平台上联合举办的一项计算机视觉赛事。本文将这项比赛使用的数据集称为 Open Image Challenge 2018 数据集。该赛事拥有两个分赛，为目标检测分赛、视觉关系检测分赛。Open Image Challenge 2018 数据集是 Open Image Dataset V4 数据集的一部分。Open Image Challenge 2018 数据集比 Pascal VOC、COCO 数据集要大不少，有空前的规模，拥有 1743042 张图片，12195144 个物体框，500 个物体类别。本来 Open Image Dataset V4 拥有 600 个物体类别，Open Image Challenge 2018 将其中概念宽泛的类别（如衣服），和一些稀少的类别（如裁纸机）去掉了。

Open Image Challenge 2018 数据集和 Pascal VOC、MS-COCO 数据集有很不同的地方，它有图片级别的标签。具体为：对于一张图片，它拥有多个类别的标签，表示某类出现在这张图片中，或不在这张图片中，分别用 1 和 0 表示。对于没有被标明的类别，则该图片对这些类别不用作训练或评估准确率。另外，在其官方说明中描述：若一张图片被标识为存在某类，官方会尽全力地标注出该图片上出现的该类别的每一个个体。

Open Image Challenge 2018 数据集只提供了训练集，但是它提供了一个从训

练集中抽取验证集，推荐的图片名单。且官方不建议使用 Open Image Dataset V4 数据集的验证集作为验证集，因为 Open Image Challenge 2018 数据集被标注的更细致，无误。

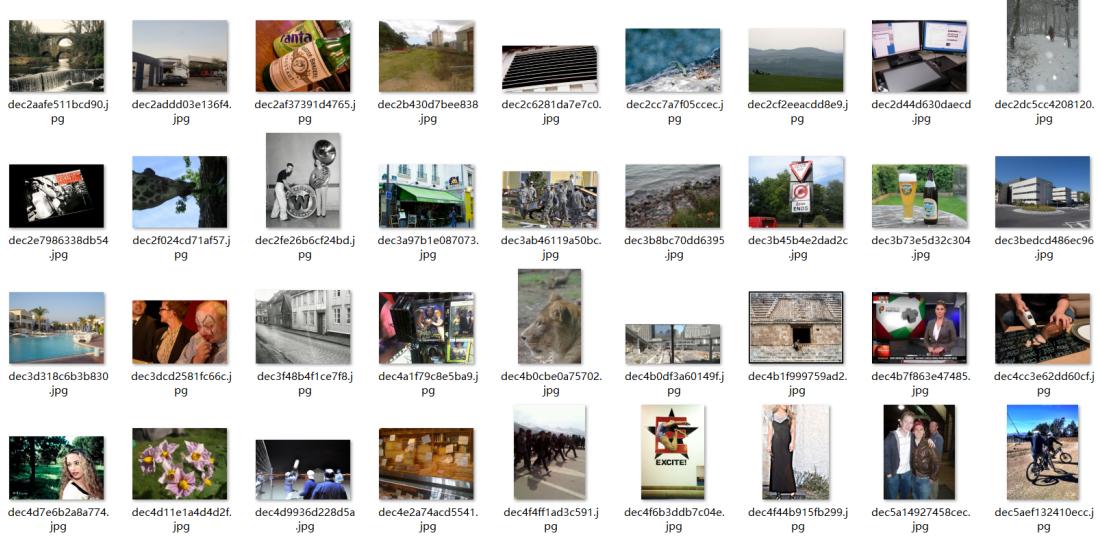


图 3.3 Open Image Challenge 2018 数据集中的图片

3.2 数据集的使用与处理

本文使用在 MS-COCO 上训练的模型作为已有模型，故新添加的类别不能是 MS-COCO 中被标注的类别。而 Pascal VOC 中被标注的类别，是 MS-COCO 中被标注的类别的子集。所以本文不使用 Pascal VOC 作为新类别的训练集（本文对其的使用是在 4.4 节中，用其作为旧类别的训练集）。Open Image Challenge 2018 中被标注的类别有 500 个，故其一定有 MS-COCO 中未标注的类别。所以本文从 Open Image Challenge 2018 中选取要添加的新类别。

本文首先对 Open Image Challenge 2018 数据集中各类物体的实例数进行了统计，发现不同类别的实例数有较大差异。统计发现，实例数最多的男人(Man)类别，有 1418594 个实例；实例数最少的高压锅(Pressure cooker)只有 14 个实例。为了不使新类别的训练样本、测试样本过少，本文从实例数多的类别开始选择。

最终，本文选择了 3 种数量合适的、且 MS-COCO 中没有的类别，作为新添加的类别进行实验，分别是：花(Flower)、眼镜(Glasses)、人脸(HumanFace)。

Open Image Challenge 2018 数据集十分庞大,其图片数据的压缩包共有 512G,在其提供的 figure8 下载网站上, 图片数据被分为了 9 个压缩包。平均每个 57G。由于其整个数据集十分庞大, 所以本文只使用了前 3 个压缩包的花、眼镜数据, 和第 1 个压缩包的人脸数据, 将这些数据用作训练和测试。并且去除了被标注为一堆(Group)标签的物体框。实验中不包含验证集, 即只有训练集和测试集, 关于训练集, 测试集的划分: 本文将出现在官方提供的建议验证集列表中的图片作为测试集, 未出现在其中的作为训练集。

这样处理后, 花卉类别的训练集图片有 10315 张, 测试集图片有 1467 张; 眼镜类别的训练集图片有 14071 张, 测试集图片有 1266 张; 人脸类别的训练集图片有 36941 张, 测试集图片有 2605 张。本文对所有训练图片进行水平翻转来进行数据增强, 所以最终每个新类别的训练集图片数是上述数目的 2 倍。

第4章 基于特征提取思想的研究

在 1.3.2 节中，本文列出了一些为图像分类网络添加新的可分类类别的方法。其中就有特征提取方法。该方法是一种直观的方法，其优点是：没有改动测试时，图片从输入网络，到输出关于旧类别预测值的计算通路上所有的参数。所以，网络关于旧类别的分类能力被完整保留。该方法适合用在不能损失一点旧类别分类准确率的场合。

本章的工作是将特征提取方法应用到了目标检测网络上，对其性质进行了详细的实验。同时，提出了改进特征提取方法的两个做法。下面进行详细介绍。

4.1 具体方法

本章以 Detectron^[8]开源的，ResNet-50 作为主干网的 FPN，在 MS-COCO 数据集上训练得到的模型为例，为其添加新的可检测类别。

该模型是在 MS-COCO 数据集上训练得到的，是一个在 MS-COCO 数据集上能较好地检测 80 类物体的模型。那么有理由认为，其 RPN 模块、CLS 模块最后一层隐层产生的特征能做的任务不是那么特定。所以其网络前部的层，可直接作为一个通用的特征提取器。下面介绍本章方法的具体细节。

网络结构上：为了能检测新的类别，我们需要对已有网络添加新的层。具体为：在 RPN 模块的隐层后，为新类别添加一个专有的分类(cls)层和回归(reg)层。在 CLS 模块的两层全连接后，也为新类别添加专有的 cls 层和 reg 层。同原版 FPN 一样，RPN 模块中添加的 cls 层和 reg 层是卷积层，CLS 模块中添加的 cls 层和 reg 层是全连接层。

改动后的网络结构，与未改前的结构大部分一样。只有 RPN 模块和 CLS 模块不一样。这两个模块改动后的结构如图 4.1 所示。

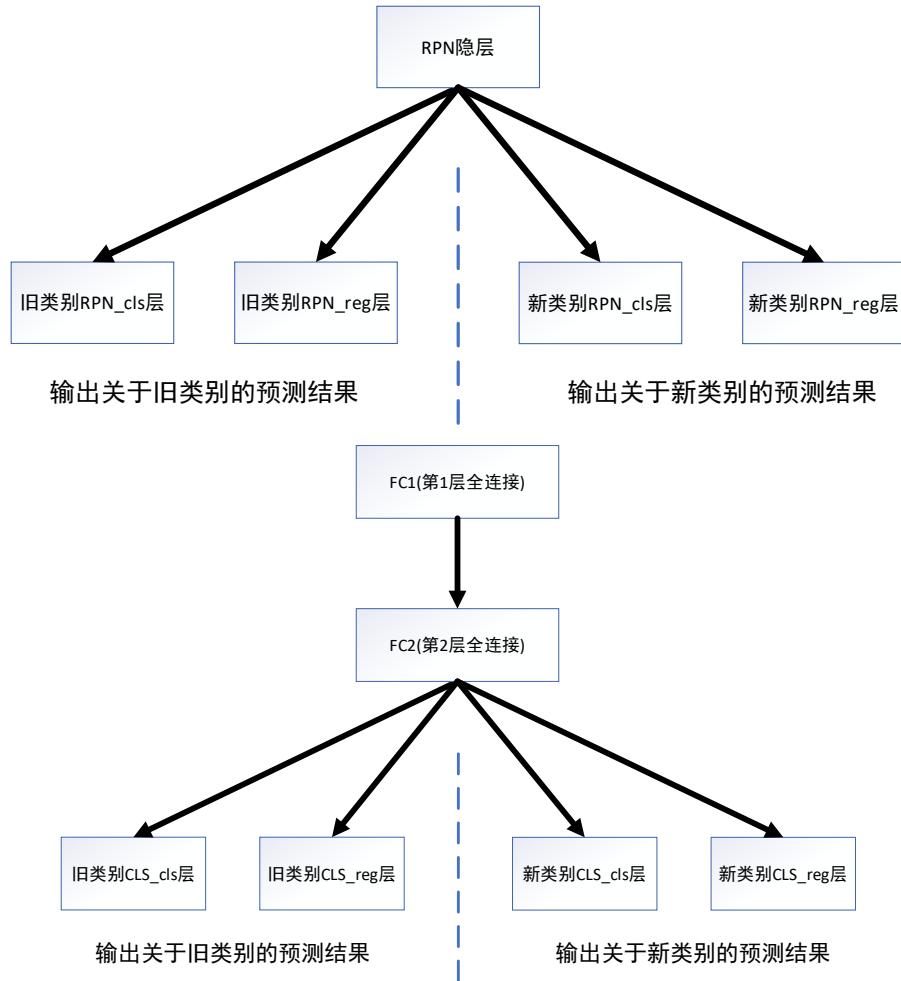


图 4.1 改动后的 RPN 模块和 CLS 模块

测试时：RPN 模块、CLS 模块中新添加的 cls 层和 reg 层输出关于新类别的预测值。即，RPN 模块的 cls 层输出每个 anchor 是新类别中目标的概率，reg 层输出每个 anchor 的形变、平移参数；CLS 模块的 cls 层输出每个 RoI 具体是什么新类别的概率，reg 层输出每个 RoI 的形变、平移参数。

训练时：RPN 模块中，旧类别的 cls 层和 reg 层后没有损失函数，新类别的 cls 层和 reg 层后跟有训练用的 Sigmoid 交叉熵损失函数、SmoothL1 损失函数。CLS 模块中，旧类别的 cls 层和 reg 层同样没有损失函数，新类别的 cls 层和 reg 层后跟有训练用的 Softmax 交叉熵损失函数、SmoothL1 损失函数。这些损失函数的真实标签如介绍 FPN 时所述的方式得出。且在 RPN 模块、CLS 模块中新类别 cls 层，reg 层之前，添加阻断梯度传播的 StopGradient 层，阻止梯度向前传播，

即不训练网络的其他层。

该方法可以看成是：旧类别、新类别使用同样的网络前部层进行特征提取。所以被称作特征提取方法。然后将相同的特征输入到各自专有的最后一层。因为 FPN 是两阶段的目标检测算法，所以其中关于 RoIs 的细节是：在 RPN 阶段得到新、旧类别的 RoIs 后，将新、旧类别的 RoIs 对应的特征子图堆叠(Concat)起来，输入后面的 CLS 阶段。

4.2 实验结果

本章使用在第 3 章描述的，从 Open Image Challenge 2018 数据集获得的 3 类物体的训练图片，分别对 ResNet-50-FPN 网络做添加类别的实验。本章的实验代码已开源在 <https://github.com/zhonhel>。

注意，本文是将 3 种类别分别添加进已有网络，得到了 3 个网络。而不是将 3 种类别连续添加进已有网络，得到 1 个网络。虽然对于本章的方法，这两种做法最终得到的准确率一样。但对于第 5 章的方法，这两种做法得到的结果应是不一样的。在第 5 章中，我们同样得到的是 3 个网络，而不是连续添加类别，去得到 1 个网络。还有值得注意的是，本章和第 5 章中基于的 FPN 算法，均是将 RoIPooling 换为 RoIAlign^[53]的 FPN 算法。

本章使用的优化算法是：随机梯度下降(Stochastic Gradient Descent, SGD)；学习率设置为 0.0025；动量设置为 0.9；正则项系数(Weight Decay)设置为 0.0001；训练进行至 2/3 时，学习率降为原先的 10%，训练进行至 8/9 时，学习率再降为原先的 10%；批大小(Batch Size)设置为 2。对于迭代次数，在添加眼镜的实验中进行 18000 次迭代，在添加花卉的实验中进行 36000 次迭代，在添加人脸的实验中也进行 36000 次迭代（因为花卉类别的训练的收敛速率较慢，人脸类别的训练图片较多）。

表 4.1 是使用本章方法实验得到的，新类别的准确率。本章没有将 3 个网络在旧类别测试集(COCO minival set)上的准确率展示出来，因为它们和已有网络在 COCO minival set 上的准确率是一样的，为 59.2，单位为 mAP@0.5。

表 4.1 本章方法的实验结果

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	新类别平均值	人脸	眼镜	花卉
默认设置	52.7	70.3	56.5	31.4

从表 4.1 的结果可看出：仅在最后一层，并行地添加关于新类别的输出层，并且只训练新添的层，是可以使模型对新类别有检测能力的，且准确率达到了一定水平。

对比用新类别训练集微调整整个网络的情况（见表 5.1），仅训练新添层确实和其在新类别准确率上有差距。但用新类别训练集微调整整个网络时，最终网络对旧类别的检测能力会严重丧失（见表 5.1）。而仅训练新添层则完好地保留了网络对旧类别的检测能力。

为了进一步提高网络在新类别上的准确率，并且增加的时间开销尽可能少。本章做了如下的实验。它们分别是：4.3 节-微调层数与新类别准确率的关系、4.4 节-已有网络可检测类数与新类别准确率的关系、4.5 节-新类别专有层的添加位置、4.6 节-添加浅层旁路。

4.3 微调层数与新类别准确率的关系

本节探究微调层数与新类别准确率的关系。具体做法是，逐步复制已有网络的 RPN 隐层、CLS 模块中的全连接层、Post-hoc 层、侧面连接层，给新类别作为专有层。新、旧类别的这些层呈现并行、姊妹的关系。如图 4.2，即为新类别添加专有的 RPN 隐层后的图示。

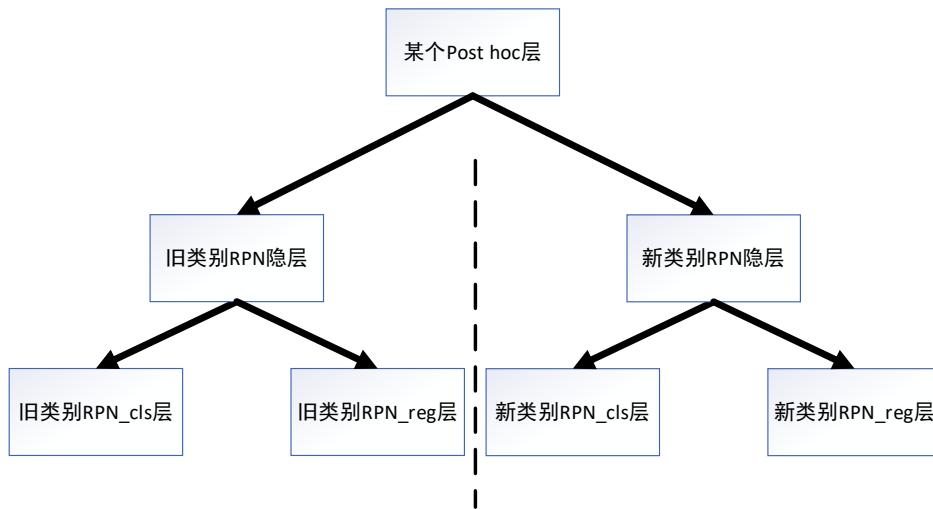


图 4.2 为新类别添专有的 RPN 隐层图示

本节从网络尾部到网络头部，逐步增加新类别的专有层数量。并通过实验，来量化这样做，对新类别准确率的影响。本节将仅在 RPN 模块、CLS 模块中为新类别添加专有输出层，记为：RPN+0，CLS+0，即 4.1 节描述的网络结构。

进一步地：RPN+1 表示为新类别添加专有的、可训练的 RPN 隐层；RPN+2 表示为新类别添加专有的、可训练的 Post-hoc 层；RPN+3 表示为新类别添加这样的侧面连接层。CLS+1 表示为新类别这样的 CLS 阶段第二个全连接层；CLS+2 表示为新类别添加这样的 CLS 阶段第一个全连接层。

在本节中，实验的设置有(RPN+1,CLS+1)、(RPN+2,CLS+2)、(RPN+3,CLS+2)。关于这些网络结构的实验结果如表 4.2 所示。表 4.2 括号中的值，是与表 4.1 中对应值进行比较的结果。

表 4.2 微调层数与新类别准确率的关系

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	新类别 平均值	人脸	眼镜	花卉
RPN+1,CLS+1	63.8(+11.1)	82.0	69.3	40.0
RPN+2,CLS+2	70.5(+17.8)	88.0	75.1	48.4

RPN+3,CLS+2	72.4(+19.7)	88.4	76.7	52.0
-------------	-------------	------	------	------

从表 4.2 的结果可看出：随着新类别可微调层数的逐渐增加，新类别的准确率逐渐上升，但同时时间开销也在逐渐增大。

并且，若更仔细地观察可发现：在只多添加一层微调层时，新类别准确率增加较多；再多添加层时，新类别准确率的增益递减。

故可得出结论：在本章的方法框架中，通过逐步增加微调层数来使新类别准确率提升的做法，是和时间开销之间的一种权衡(Trade-off)。主要看实际使用时，是更注重新类别的准确率，还是更注重检测的速度。且添加的层数越多，准确率增益递减。在 FPN 中，只添加到侧面连接层（即没有训练 ResNet 的所有层），就可相对微调整个网络只低 2.4（对比表 5.1 得出）的准确率。

4.4 已有网络可检测类数与新类别准确率的关系

本节探究在本章的方法中，已有网络可检测类数与新类别准确率的关系。本节想要探究，是否一个能检测更多类物体的网络，通过这种简单的方法在增加新的可检测类别时，能使新类别有更高的准确率。如果是这样的话，那么就可以知道，如果想用这种简单方法为网络添加新的可检测类别，需尽可能地训练一个能检测更多类的已有网络。

在 4.1 节的默认设置中，已有网络是在 MS-COCO 2014 train+valminusminival（等同于 MS-COCO 2017 train+val）数据集上训练得到的，且是网上公开可获取的模型。由于 MS-COCO 目标检测数据集有 80 个物体类别，所以该模型也是针对检测这 80 类物体的。

为完成本节的实验，本节将在一个少于 80 类物体的目标检测数据集上微调，这个已有的在 MS-COCO 上训练的 FPN 目标检测模型。微调使用的数据集是 Pascal VOC 数据集，具体为：Pascal VOC 2007 训练集，加上 Pascal VOC 2007 验证集，加上 Pascal VOC 2012 训练集，加上 Pascal VOC 2012 验证集。

因为 FPN 在多个分辨率不同的特征图上做有无物体的判断（RPN 阶段），而

产生这些分辨率不同的特征图的计算通路上有不共享的，含参数的层（侧面连接层、Post-hoc 层）。因为 Pascal VOC 中物体的尺寸都比较大，为使这些层被充分地训练，本节在使用 Pascal VOC 数据集微调已有模型时，将用于训练的图片较短边的长度随机调整为 300、500、800 中的一个值，并保持图片的高宽比不变。通过调整训练图片的大小，使 FPN 中分辨率较高的分支，得到了充分的训练。最终本节得到了一个在 Pascal VOC 2007 测试集上 mAP@0.5 为 87.44 的模型。本节使用这个模型，重复本章 4.1 节的做法。即，只为新类别添加 RPN 模块、CLS 模块中的输出层，形成了本节的实验，表 4.3 为实验结果。表 4.3 括号中的值，是与表 4.1 中对应值进行比较的结果。

表 4.3 已有网络可检测类数与新类别准确率的关系

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	新类别 平均值	人脸	眼镜	花卉
已有网络使用 Pascal VOC 数据集 训练得到	49.2(-3.5)	68.6	50.9	28.0

从表 4.3 的结果可看出：在较少类的数据集上训练得到的模型（即便是先在多类上训练，再在少类上微调），没有在较多类的数据集上训练得到的模型，在添加新的可检测类别时，在新类别上的准确率高。

这进一步说明了：（1）神经网络在尾部的层的输出确实是任务特定的 (Task-specific)。即，这些层输出的是，对特定类别分类有用的特征。特定类别即为在训练时使用的类别。（2）一个图像分类/检测网络能识别的类别数越多，那么其后部的层输出的特征越丰富，越适合用本章的方法为其添加新的可分类/检测类别。

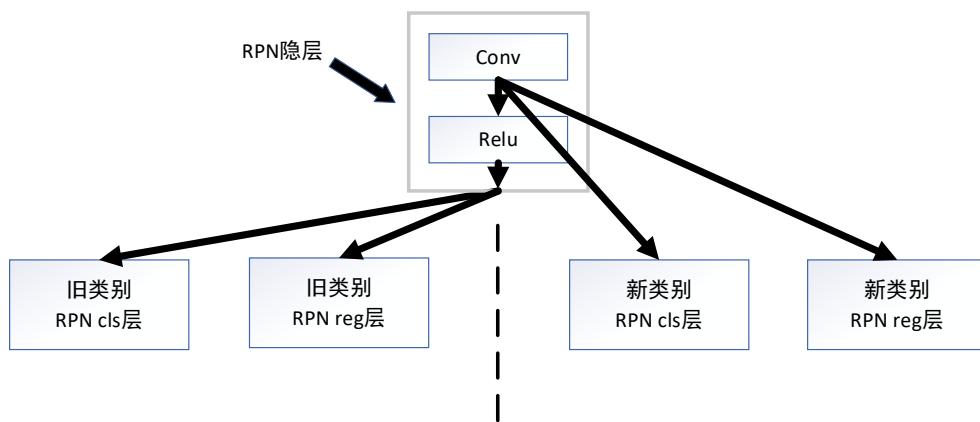
4.5 新类别专有层的添加位置

在 4.1 节的具体方法中，只为新类别添加了 RPN 模块、CLS 模块中最后的输出层。在本节中，同样也只添加新类别专有的输出层。但是，在本节中，我们讨

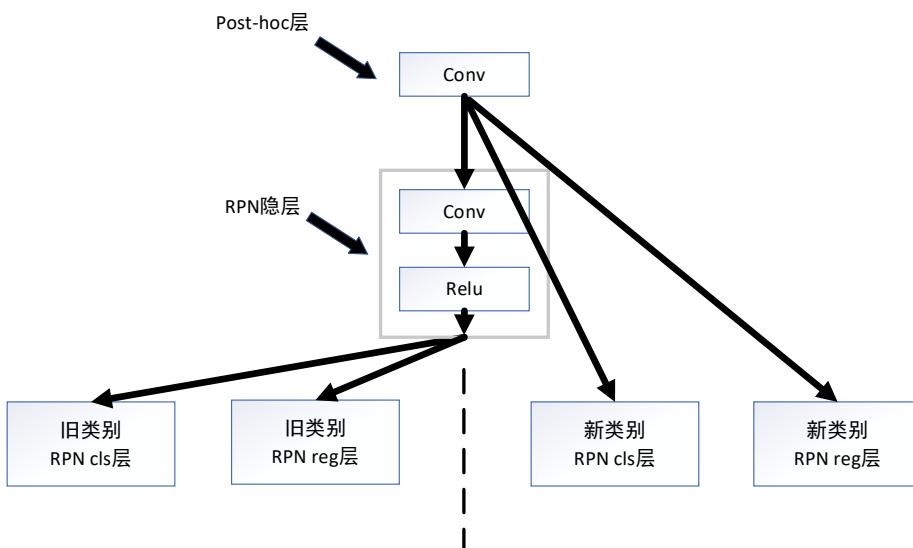
论为新类添加专有输出层的位置。

通过 4.4 节的实验可得出结论：神经网络靠后的层是针对其被训练的特定任务的(Task-specific)，将他们作为较通用的特征提取器可能不太合适。那么，在其后添加新类别的专有层是否是不好的做法，是否可将新类别专有层的位置前移呢？

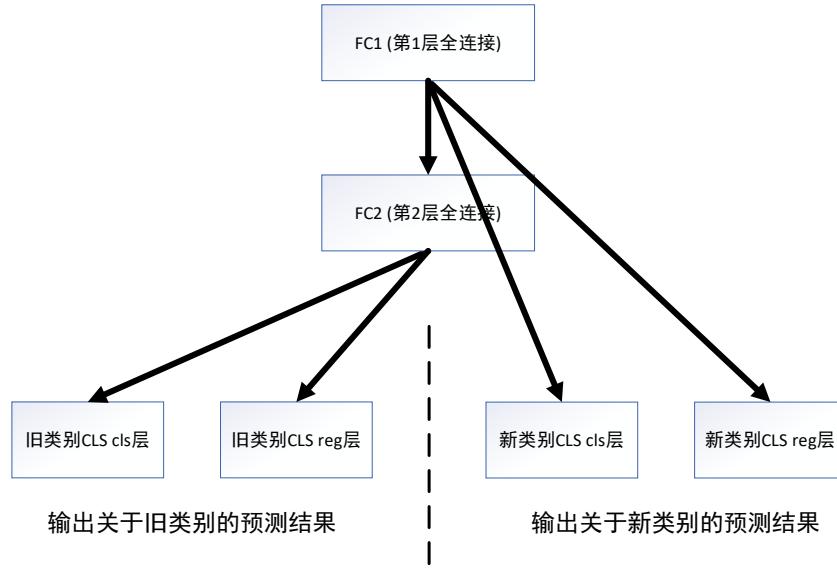
为探究上述的问题。本节实验了 3 种设置：(1) 将 RPN 模块中，新类别的专有输出层添加到 RPN 隐层的 ReLU 层之前。(2) 将 RPN 模块中，新类别的专有输出层添加到 RPN 隐层之前。(3) 将 CLS 模块中，新类别的专有输出层添加到第二个全连接层之前。图 4.3 展示了这 3 种设置下的网络结构。



(1) 将 RPN 模块中，新类别的专有输出层添加到 RPN 隐层的 ReLU 层之前



(2) 将 RPN 模块中，新类别的专有输出层添加到 RPN 隐层之前



(3) 将 CLS 模块中，新类别的专有输出层添加到第二个全连接层之前

图 4.3 三种新类别专有层的添加位置

训练时的迭代次数等设置，与默认设置相同。这三种新类别专有层添加位置的实验结果如表 4.4 所示。表 4.4 括号中的值，是与表 4.1 中对应值进行比较的结果。

表 4.4 三种新类别专有层添加位置的实验结果

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	新类别平均值	人脸	眼镜	花卉
RPN 隐层中 ReLU 之前	52.9(+0.2)	70.4	57.5	30.8
RPN 隐层之前	52.9(+0.2)	70.7	56.7	31.2
第 2 个全连接层之前	56.8(+4.1)	74.1	63.5	32.9

从表 4.4 的结果可看出：将新类别的专有输出层前移，在除该层外的所有层不训练时，确实可提高新类别的准确率。且，将这种做法应用在 RPN 阶段提高较少，但将这种做法应用在 CLS 阶段时，产生了出乎意料的结果，新类别的准确率提升很大，提升了 4.1。值得注意的是，这种做法和默认设置的计算量是一样的。

同时，其检测速度能比默认设置快一些，因为这样做时，网络关于新类别的检测结果能更早地得出。

同时，实验结果也验证了：卷积神经网络越靠后的层，越是和训练任务紧密相关(Task-specific)，其通用的特征提取作用越弱，它滤除了一些和训练任务无关的信息。

所以，本节得出了一个改进本章方法的做法，即为：将新类别专有层的添加位置前移。从实验结果可以看出，这个改进能有效地提高网络对于新类别的检测准确率。

4.6 添加浅层旁路

为了进一步提高本章方法的准确率。本节提出添加浅层旁路的方法，为本章方法添加改进。从 4.3 节的实验结果可以看出：从网络的后部到前部，逐步为新类别添加可微调的专有层，可逐步提升新类别的准确率。当为新类别添加专有的侧面连接层后，可以发现：由于 FPN 是在多个分辨率的特征图上做物体建议区域的提取。所以，如果想再往前为新类别添加一个专有层，误差的反传会在 ResNet 上不连续。因为这些分辨率不同的特征图之间，有多个残差模块。

鉴于此，本节提出添加浅层旁路的方法。下面介绍本节方法的具体内容。

网络结构：本节延续第 2.5 节介绍 FPN 时的名称，将输入侧面连接层的分辨率从低到高的特征图称为：res5_2_sum、res4_5_sum、res3_3_sum、res2_2_sum。这些名称的含义，如 2.5 节中所述。

本节将 res2_2_sum 输入一个新的卷积层中，该卷积层核为 1，步长为 2，卷积核数为 512。将该卷积层得到的结果输入一个 ReLU 层，将该 ReLU 层输出的结果记为 shallow_bypass_conv1。将 shallow_bypass_conv1 和 res3_3_sum 堆叠(Concat)起来，作为新的 res3_3_sum。

接着，再将 shallow_bypass_conv1 输入第二个新的卷积层（核为 1、步长为 2、卷积核数为 512），ReLU 层，将输出的结果记为 shallow_bypass_conv2。将 shallow_bypass_conv2 和 res4_5_sum 堆叠起来，作为新的 res4_5_sum。

类似地，将 shallow_bypass_conv2 再输入一个新的卷积层（同样核为 1、步长为 2、卷积核数为 512），ReLU 层，将输出的结果记为 shallow_bypass_conv3。将 shallow_bypass_conv3 和 res5_2_sum 堆叠起来，作为新的 res5_2_sum。将更新过的 res3_3_sum、res4_5_sum、res5_2_sum 输入它们各自的侧面连接层。添加浅层旁路后的 FPN 网络结构如图 4.4 所示。

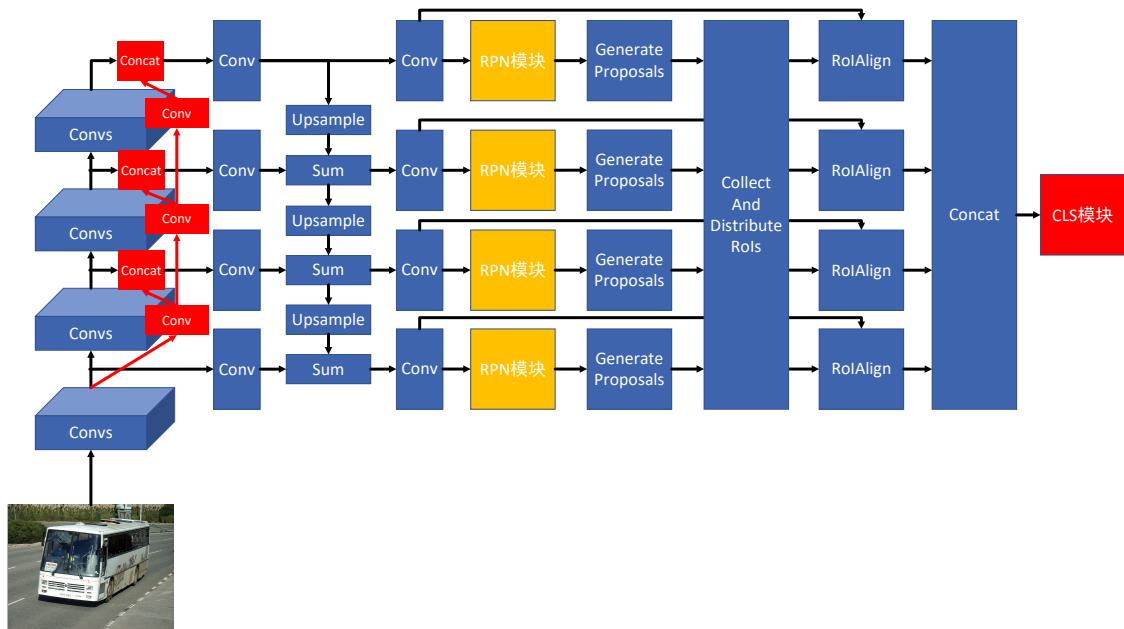


图 4.4 添加浅层旁路后的 FPN 网络结构

图 4.4 中，左侧红色模块即是本节添加的浅层旁路。称其旁路的原因是：相对于 ResNet-50 这个主干网来讲，它是旁边的支路。称其浅层的原因是：它只有 3 层卷积层，相对于其旁边的 ResNet-50 主干网是相当浅的。因此这样做额外添加的时间开销也相当的少。但是却可以使新类别准确率有一些提高。

为新类别添加浅层旁路后，新类别的准确率如表 4.5 所示。表 4.5 括号中的值，是与 4.3 节中(RPN+3,CLS+2)的设置进行比较的结果。

表 4.5 添加浅层旁路后的结果

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	新类别 平均值	人脸	眼镜	花卉
添加浅层旁路	72.9(+0.5)	88.9	78.2	51.6

从表 4.5 的结果可看出：增加浅层旁路后，网络在新类别上的准确率相较于(RPN+3,CLS+2)的设置平均提升了 0.5。但值得注意的是：该方法对于不同类别有不同的表现，在人脸类别上，提升了 0.5；在眼镜类别上提升了较多，为 1.5；在花卉类别上有所下降，下降了 0.4。所以在使用该方法前，应先实验是否可以提升所添加类别的准确率。

值得指出的是：根据表 5.1 可知，将网络完全用新类别训练集训练后，网络在新类别上的准确率为 74.8。而(RPN+3,CLS+2)设置下的准确率为 72.4。故在这种情况下，不微调 ResNet 主干网（由于 FPN 的特殊性，无法较好地只微调 ResNet 中的部分层），通过增加 3 层计算开销相当少的卷积层，可以提高 0.5 的准确率，是具有一定意义的。

另外，仍需指出的是，本节在实验时尝试了不同的设置。具体为：(1) 将堆叠(Concat)替换为对应元素相加(Element-wise Sum)，实验结果明显低于(RPN+3,CLS+2)的设置。(2) 将添加的三层卷积层设置为：核为 3、步长为 2、卷积核数为 256，实验结果没有目前的设置效果好。(3) 将添加的三层卷积层设置为：核为 3、步长为 2、卷积核数为 256，实验结果和目前的设置差不多。但这样的设置中，因卷积核的大小为 3，所以添加的三层卷积层，参数量和计算量是目前设置的 9 倍。所以，通过进行了上述这些实验，本节最终才采用了目前的设置。

4.7 本章小结

本章讨论了，在以本章方法（特征提取方法）为框架时，为一个已有网络添加新的可检测类别时，所做的实验，和改进它的几个方法。

在本章中，我们通过实验，说明了以下的几点结论。这些结论对使用本章方

法（特征提取方法）为网络添加新的可检测类别有重要的意义。

- (1) 仅为新类别添加专有的输出层，网络就有一定的检测新类别的能力。
- (2) 若从网络后部到前部，为新类别添加的专有层越多，其准确率会逐渐升高，并且这种准确率增益会逐步减少。
- (3) 已有网络的通用特征提取能力，和其能识别的类别数正相关。若已有网络能识别越多的类别，那么用本章方法为其添加新的可检测类别时，会使网络在新类别上达到越高的准确率。
- (4) 本章提出的将新类别专有层的添加位置前移的方法，能可观地提升新类别的准确率。本章通过实验探究发现：新类别专有层的添加位置是一个重要的，影响新类别准确率的因素。这验证了神经网络越靠后的层，越与其原始训练任务相关，它滤除了一些对其原始任务无用的信息，而这些信息中有些恰恰是识别新类别所需要的。
- (5) 本章提出的添加浅层旁路的方法，能提升新类别的准确率。该方法将浅层特征图，经少量卷积层作用后堆叠(Concat)到 RPN 作用的特征图上。同时，主干网仍是不被新类别任务训练的。该方法能在仅增加少量计算量的情况下，提升网络在新类别上的准确率。

第5章 基于保留损失函数方法的研究

在本章中，将介绍本文基于 LwF 提出的基于保留损失函数的方法。不同于第 4 章基于特征提取思想的方法，本章的方法，改动了网络在测试时，图片从输入网络，到输出关于旧类别预测值的计算通路上的参数。故，最终形成的网络在旧类别上的检测准确率上会有变化。

在介绍完该方法及其实现后，本章进行了多个关于细节的实验，来研究该方法中的各种细节。最终，本章对于每一个细节，都通过实验给出了最好的处理方式。

5.1 具体方法

本章的工作是基于 LwF^[28]和 Shmelkov 等人所做的工作^[40]。如 1.3 节国内外研究现状中所述：LwF 是 ZhiZhong Li 等人借鉴模型蒸馏，做图像分类的增量学习。而 Shmelkov 等人是借鉴 LwF 方法，做目标检测的增量学习。

但如 1.3 节国内外研究现状中所述，Shmelkov 等人实现的是基于 Fast R-CNN 的目标检测增量学习，没有区域建议网络(RPN)模块，它的物体候选框是用 Edge Boxes 算法^[14]或 MCG 算法^[41]提取的。这两个算法提取的物体候选框和要检测的物体类别无关(Class-agnostic)，这两个算法的目的是提取所有非背景的物体区域。在 LwF 的论文中提到其未来工作是想将其方法拓展到目标检测等其他任务上；在 Shmelkov 等人的论文中提到未来工作是想将其方法拓展到有区域建议网络(RPN)模块的目标检测算法中。所以，本章工作可看成是 LwF、Shmelkov 等人工作的未来工作。本文率先将 LwF 的方法拓展到了具有区域建议网络(RPN)模块的目标检测算法中。

下面介绍本章的具体方法。

在网络结构上：和 LwF、Shmelkov 等人的工作相同，训练需要有两个网络。一个是固定不训练的网络，本章称之为固定网络。该网络是已有的、在旧类别上训练好的了，FPN 目标检测模型；另一个是被训练的网络，本章称之为被训练网

络。被训练网络的结构和 4.1 节描述的网络结构一样，即，在 RPN 模块和 CLS 模块中分别添加了检测新类别物体用的 cls 层和 reg 层。

测试时：使用训练好的被训练网络进行测试。其中 RPN 模块、CLS 模块中旧类别的 cls 层、reg 层输出关于旧类别的预测值，新类别的 cls 层和 reg 层输出关于新类别的预测值。新、旧类别共享除各自专有的 cls 层、reg 层之外的所有参数。

训练时：将被训练网络 RPN 模块中旧类别专有的 Sigmoid 层的输出、reg 层的输出，与固定网络对应位置的输出计算 SmoothL1 误差（即添加 SmoothL1 损失函数层）。将被训练网络 CLS 模块中旧类别专有的 cls 层的输出、reg 层的输出，与固定网络对应位置的输出计算 SmoothL1 误差（也即添加 SmoothL1 损失函数层）。

同时，被训练网络还被新类别的检测误差进行训练。新类别的检测误差和普通 FPN 的训练过程中定义的相同。即，在 RPN 模块中，是 cls 层的输出和真实标签的 Sigmoid 交叉熵误差、reg 层的输出和真实标签的 SmoothL1 误差；在 CLS 模块中，是 cls 层的输出和真实标签的 Softmax 交叉熵误差、reg 层的输出和真实标签的 SmoothL1 误差。

在训练的一次迭代中，将被训练网络的这些误差同时反传，得到它们关于每个参数的导数。根据梯度下降，进行参数的更新，即完成了一次迭代的训练。

若将被训练网络中，关于旧类别的误差定义为 $Loss_{old}$ （即，保留误差），关于新类别的误差定义为 $Loss_{new}$ ，那么训练时，被训练网络的误差可表示为：

$$Loss_{total} = Loss_{old} + Loss_{new} \quad (5.1)$$

$Loss_{old}$ 可表示为：

$$\begin{aligned} Loss_{old} = & SmoothL1_{RPN_cls} + SmoothL1_{RPN_reg} \\ & + SmoothL1_{CLS_cls} + SmoothL1_{CLS_reg} \end{aligned} \quad (5.2)$$

$Loss_{new}$ 可表示为：

$$\begin{aligned} Loss_{new} = & SigmoidCrossEntropyLoss_{RPN_cls} + SmoothL1_{RPN_reg} \\ & + SoftmaxCrossEntropyLoss_{CLS_cls} + SmoothL1_{CLS_reg} \end{aligned} \quad (5.3)$$

另外，在训练中，只使用新类别的训练集图片，和其标注信息(Label)。不使用旧类别的训练图片。对于每次迭代，可将它的过程分为 4 步：

- (1) 将训练图片同时输入固定网络和被训练网络。将固定网络 RPN 模块中 Sigmoid 层的输出结果、reg 层的输出结果保存下来。
- (2) 将被训练网络 RPN 模块得出的，关于旧类别的物体候选框(RoIs)数据传给固定网络。
- (3) 固定网络和被训练网络根据一模一样的 RoIs，从各自的特征图中，提取与 RoIs 对应的特征子图，将这些特征子图传递给各自后续的 CLS 模块。之后，固定网络将其 CLS 模块 cls 层的输出结果、reg 层的输出结果保存下来。
- (4) 被训练网络将其 RPN 模块中旧类别 Sigmoid 层的输出结果、旧类别 reg 层的输出结果、CLS 模块中旧类别的 cls 层输出结果、旧类别 reg 层的输出结果，分别以固定网络对应的输出为回归目标，计算 SmoothL1 误差；将关于新类别的输出像训练普通 FPN 那样，与真实标签计算误差。利用反向传播算法，计算所有误差对网络中每个参数的导数，然后根据梯度下降更新这些参数。

训练过程可用图 5.1 表示。其中，虚线箭头表示将梯度反向传播。

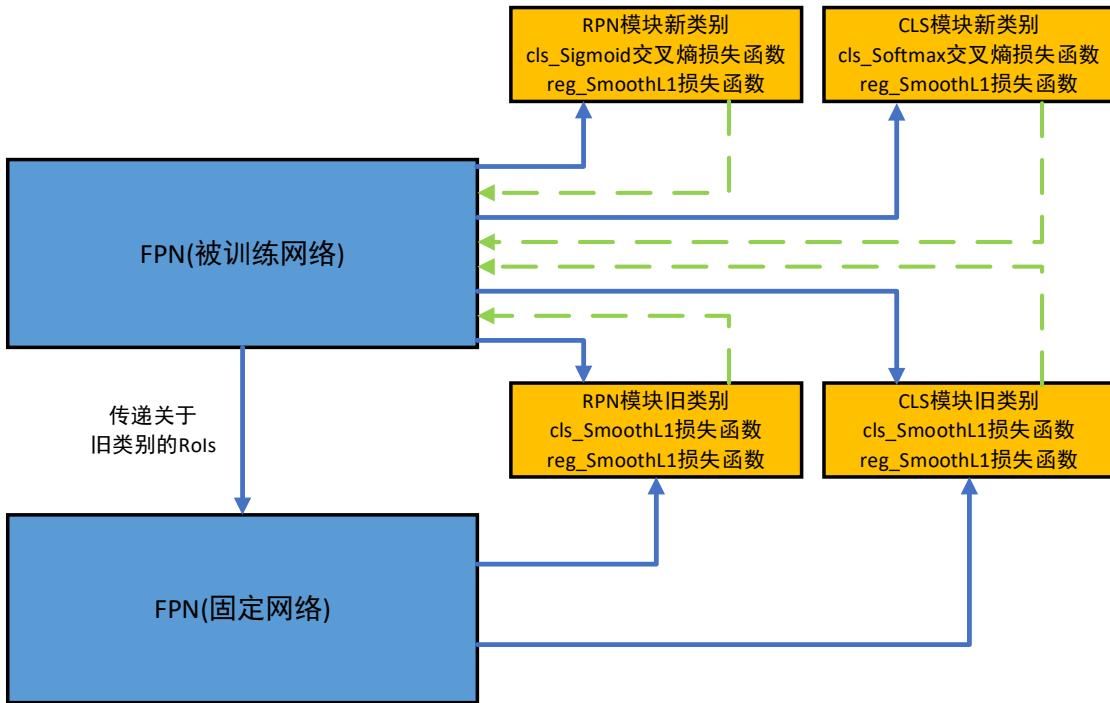


图 5.1 基于保留损失函数方法的训练过程

5.2 实现细节

本章使用两块 GTX 1080Ti 显卡进行训练。在一张显卡上进行固定网络的前向传播，在另一张显卡上进行被训练网络的前向传播和反向传播。使用的深度学习框架是 Caffe2，主要开发语言是 Python。本章的实验代码已开源在 <https://github.com/zhonhel>。

在训练过程中，需要在两个网络之间，即两张显卡之间传输数据。具体为：

- (1) 被训练网络需将其 RPN 模块得出的 RoIs 数据传递给固定网络。
- (2) 固定网络需将其 RPN 模块中 Sigmoid 层的输出、reg 层的输出传递给被训练网络。
- (3) 固定网络需将其 CLS 模块中 cls 层的输出、reg 层的输出传递给被训练网络。

受限于 Caffe2 框架关于张量(Tensor)的封装，本章没有使用 NCCL 进行不同 GPU 之间数据的传输。本章采用的是先将数据传输到内存，再从内存将数据传输到另一张显卡。本章传递数据选用的工具是 memcached。Memcached 是一个将数据存储在内存，可通过互联网在多机间提供数据共享服务的程序，也可为本机多个程序提供数据共享服务。其设计初衷是为了使 Web 服务器在分发数据时，更加

快捷。Memcached 是在 linux 上开源的、用 C 撰写的程序，但互联网上有人为其开发了 Python 的接口，本章使用的接口程序为 pylibmc。

5.3 实验结果

在展示本章方法的结果前，本节先展示一个对照实验的结果。该实验的内容是：在训练时，不为被训练网络添加保留误差。即，只用新类别检测误差去训练被训练网络。通过该实验想展示的是，经过这样的训练，被训练网络在初始时拥有的对旧类的检测能力，会丢失大部分，即产生了灾难性忘记。表 5.1 即为不添加保留误差的结果。

本章分别对添加人脸、眼镜、花卉类别的情况进行实验。表格中“旧类别-人脸”指在添加人脸类别时，训练好的被训练网络在旧类别上的准确率。“旧类别-眼镜”、“旧类别-花卉”的意思遵循上述规则。如 4.2 节所述，本章是将这三类分别添加进已有网络进行实验，而不是连续地添加进一个已有网络。

值得注意的是，被训练网络在初始时，在旧类别测试集(COCO minival set)上的准确率为 59.2，单位为 mAP@0.5。在分别用三种新类别图片训练后，被训练网络在旧类别测试集上的准确率分别降为了 4.8、14.7、7.3，平均起来只有 8.9。

表 5.1 不添加保留误差的结果

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	旧类别平均值	新类别平均值	旧类别-人脸	旧类别-眼镜	旧类别-花卉	人脸	眼镜	花卉
不加保留误差的结果	8.9	74.8	4.8	14.7	7.3	90.5	78.7	55.3

表 5.2 本章方法的准确率

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	旧类别平均值	新类别平均值	旧类别-人脸	旧类别-眼镜	旧类别-花卉	人脸	眼镜	花卉
默认设置	58.1	71.8	58.5	58.7	57.0	89.0	75.3	51.0

表 5.3 与 Shmelkov 等人的方法的对比

(表中数值表示 mAP@0.5)

数据集	实验设置	旧类别准确率	新类别准确率	平均
Pascal VOC 2007	Shmelkov 等人的方法 ^[40]	63.2	63.1	63.1
	本章方法	68.6	71.1	69.9
MS-COCO	Shmelkov 等人的方法 ^[40]	/	/	37.4
	本章方法	62.8	46.8	54.8

本章方法的实验结果，即添加保留误差的实验结果，如表 5.2 所示。从表 5.2 与表 5.1 的对比可看出：添加保留误差有非常大的作用。本章方法在旧类别测试集上的 mAP@0.5 比表 5.1 的结果高出了 49.2，在新类别测试集上的 mAP@0.5 只比表 5.1 的结果下降了 3.0。说明，保留误差极大程度地保留了网络对旧类别的检测能力，且对新类别检测能力的影响并没有多大。图 5.2 是用训练好的被训练网络，在新、旧类别测试集上进行检测得到的一些结果。

在之前的章节，提到了 Shmelkov 等人也借鉴 LwF 的思路，做目标检测的增量学习。本节也将本章方法应用到 Shmelkov 等人用的实验数据上，进行实验。得到的结果如表 5.3 所示。可看出，本章方法的准确率高于 Shemlkov 等人的方法的准确率。这个原因主要是，Shmelkov 等人是基于 Fast R-CNN 做目标检测的增量学习。他们采用 Edge Boxes、MCG 算法，分别从 Pascal VOC 2007、MS-COCO 数据集中提取物体候选框。这些是类别无关的(Class-agnostic)候选框提取算法。对于 RPN 训练过的类别，这些算法没有 RPN 方法好。故现今流行的目标检测算法，大多都使用 RPN 进行候选框提取。另不可否认的是，本章方法还存在 RoIAlign 和 FPN 的优势。

从该实验的成功中，可得出以下结论：通过在训练中添加保留误差，可有效地为目标检测网络添加新的可检测类别。被训练好的网络在新类别上拥有很高的检测准确率，而对旧类别的检测能力被极大地保留了下来。

在本章剩下的小节中，将探究：在本章方法中，使用不同的细节设置，对新、旧类别准确率的影响。



图 5.2 用训练好的被训练网络进行检测的结果展示(a)

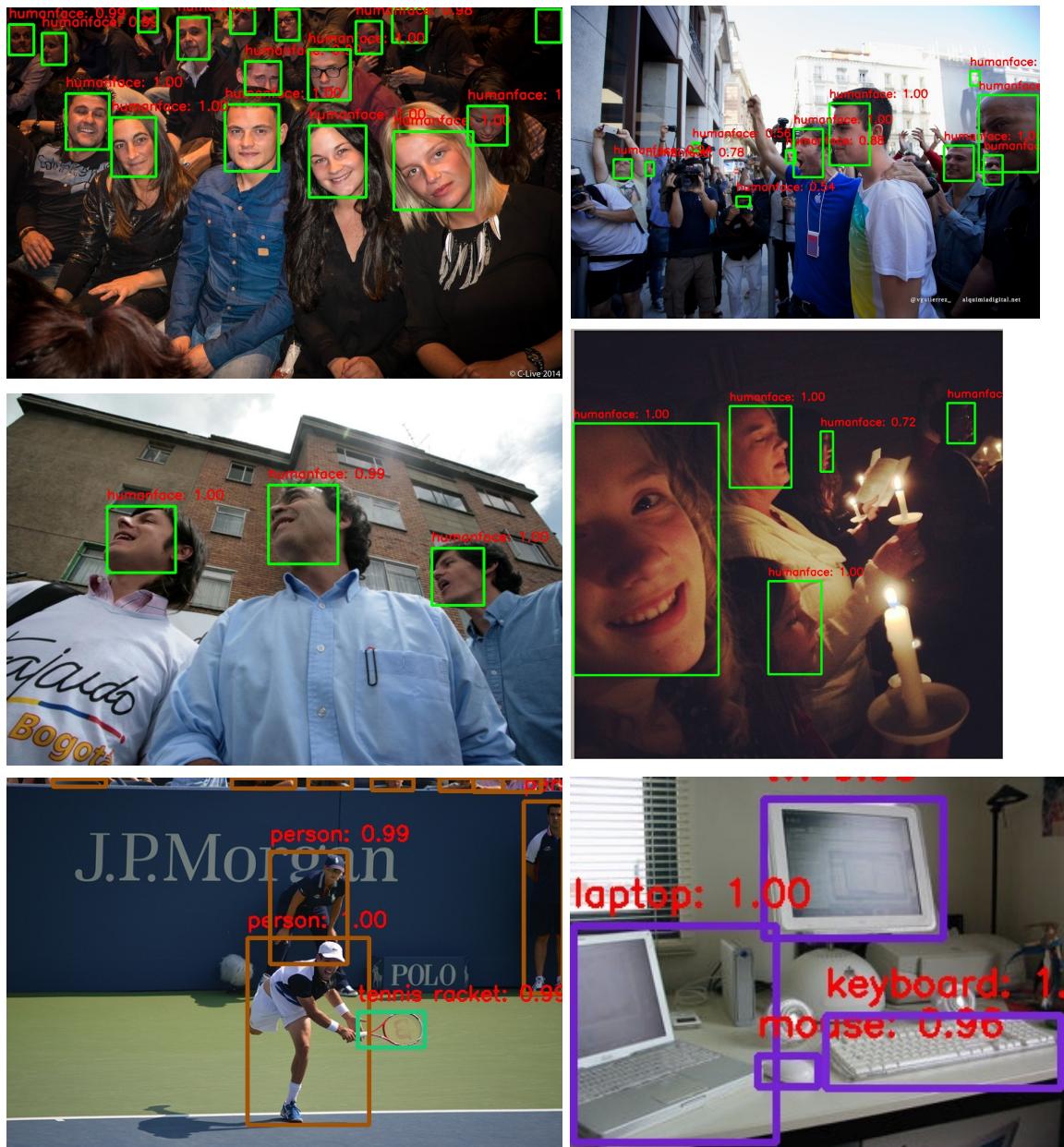


图 5.3 用训练好的被训练网络进行检测的结果展示(b)

5.4 回归目标的选取

本章方法的主要思路是，让被训练网络旧类别输出层的输出和已有网络尽可能相同。在 5.1 节描述的具体方法中，让被训练网络 RPN 阶段和 CLS 阶段的 reg 层输出，和已有网络对应的输出尽可能相同，这点没有异议。因为 reg 层的输出，没有像 cls 层那样，再经过一个 Sigmoid 或 Softmax 函数，从而得出最终的预测值。

但是，对于 RPN 阶段和 CLS 阶段的 cls 层的输出，我们不确定是让两个网络的 Sigmoid 函数、Softmax 函数之前的值相同，还是之后的值相同。即，我们不确定是将保留损失函数接在 Sigmoid、Softmax 函数之前还是之后。

在 5.1 节中介绍的具体方法中：在 RPN 模块中，是将关于分类的保留损失函数接在了 Sigmoid 之后，在 CLS 模块中，是将关于分类的保留损失函数接在了 Softmax 之前。这样做不是没有原因的，表 5.4 展示了本文在这一细节上所做的实验。表 5.4 括号中的值，是与表 5.2 中对应值进行比较的结果。

这些不同设置的本质是损失函数不同。在 LwF^[28]中，默认的损失函数是蒸馏损失函数。在 Shmelkov 等人的工作^[40]中，默认的损失函数是 L2 损失函数。在这里，本文也加入了蒸馏损失函数进行实验，在此蒸馏损失函数中，温度因子 T 的值设置为 2。T=2 是文献[30]和 LwF 中都推荐的设置。

表 5.4 回归目标的选取对准确率的影响

(表中数值表示 mAP@0.5 或 AP@0.5)

损失 函数	旧类别 平均值	新类别 平均值	旧类别 -人脸	旧类别 -眼镜	旧类别 -花卉	人脸	眼镜	花卉
Sigmoid 之前， Softmax 之前	47.8 (-10.3)	69.4 (-2.4)	52.1	51.6	39.7	87.7	74.5	45.9
Sigmoid 之后， Softmax 之后	56.1 (-2.0)	72.6 (+0.8)	57.2	58.1	52.9	89.7	76.2	51.9
Sigmoid 之后， 蒸馏损 失函数	57.5 (-0.6)	71.9 (+0.1)	58.1	58.1	56.2	89.0	75.7	50.9

从表 5.4 的结果可看出：将回归目标设定为 Sigmoid 函数之前的值，是非常不合适的，它会造成新、旧类别的准确率都降低，且在旧类别上的准确率降低很多。其他设置得到的结果和默认设置差不多。

通过分析，可得出结论：将保留损失函数接在 Sigmoid 之后、Softmax 之后，相当于缩小了旧类别的误差；将保留损失函数接在 Sigmoid 之后、CLS 模块使用

蒸馏损失函数，也相当于缩小了旧类别的误差。所以它们都使旧类别的准确率少量地降低，而新类别的准确率少量地升高。

具体地，在训练过程中发现，将回归目标设定为 Sigmoid 函数之前的值，会在几次迭代后产生很大的误差，随着训练的进行，虽然该误差会降低，但最终得到的新、旧类别的准确率确实明显低于其他设置。

5.5 物体候选框的选取

如前所述，不同于 Shmelkov 等人的工作，本章方法是直接在具有 RPN 的网络上进行的。而 Shmelkov 等人的工作是在 Fast R-CNN 上进行的。他们方法中的物体候选区域，是通过 Edge Boxes 算法^[14]或 MCG 算法^[41]生成的，这些算法不能添加保留误差去训练。所以他们的方法中，添加了保留误差的训练仅是在 RoIs 的分类和回归上进行的，没有在 RoIs 的提取过程进行。

在 Shmelkov 等人的方法中。进行保留误差计算的样本，是那些：已有网络判定为低背景概率的物体候选框。Shmelkov 等人通过实验证明了，选用低背景概率的物体候选框，可使旧类别的准确率被保留的更好。即，如果从 Edge Boxes 算法，或 MCG 算法得出的物体候选框中随机选取计算保留误差所用的样本，那么旧类别的准确率保留得没有上述情况好。

在本章方法中，会产生一个类似的问题：用什么物体候选框去进一步得出 CLS 模块的保留误差。首先能确定的是，关于保留误差的训练，我们得将相同的物体候选框传给两个网络的 CLS 模块。因为若两个网络的 CLS 模块使用不同的物体候选框，却要求它们输出相同，这明显是不正确的做法。这相当于是对两个网络输入不同的数据，却要让他们的输出相同。所以，在确定了两个网络使用相同的物体候选框后，产生的问题便是：使用哪些物体候选框？在本章方法的训练过程中，会产生 3 组物体候选框。一组是固定网络 RPN 模块得出的 RoIs；一组是被训练网络 RPN 模块得出的关于旧类别的 RoIs；最后一组是被训练网络 RPN 模块得出的关于新类别的 RoIs。

所以选取哪组物体候选框，去进一步形成 CLS 模块中的保留误差，是一个问

题。在 5.1 节描述的默认设置中，选用的是被训练网络 RPN 模块得出的关于旧类别的 RoIs。本节展示选用另外两组物体候选框得出的准确率。实验的结果如表 5.5 所示。表 5.5 括号中的值，是与表 5.2 中对应值进行比较的结果。

表 5.5 使用不同的物体建议框训练对结果的影响

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	旧类别平均值	新类别平均值	旧类别-人脸	旧类别-眼镜	旧类别-花卉	人脸	眼镜	花卉
采用固定网络产生的物体候选框	59.1 (+1.0)	70.2 (-1.6)	59.1	59.3	59.0	87.6	75.2	47.8
采用被训练网络产生的新类别物体候选框	57.7 (-0.4)	71.3 (-0.5)	58.0	58.6	56.5	88.8	74.4	50.6

从表 5.5 的结果可看出：使用固定网络产生的物体候选框，可使旧类别的准确率保留得更好，而使新类别的准确率下降。使用被训练网络产生的关于新类别的物体候选框，会使新、旧类别的准确率都轻微地下降。

5.6 预训练的重要性

在 LwF 的论文中提到了预训练的重要性。预训练，是使用保留误差训练被训练网络之前，先单独训练为新类别添加的专有层至收敛。

在 5.1 节描述的默认设置中，是添加了预训练的。所以本节展示没有预训练的结果。不加预训练的实验结果如表 5.6 所示。表 5.6 括号中的值，是与表 5.2 中对应值进行比较的结果。

表 5.6 无预训练的结果

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	旧类别平均值	新类别平均值	旧类别-人脸	旧类别-眼镜	旧类别-花卉	人脸	眼镜	花卉
无预训练	57.8 (-0.3)	71.4 (-0.4)	58.4	58.6	56.5	88.9	74.7	50.5

从表 5.6 的结果可看出：没有预训练时，最终训练得到的被训练网络，在新、旧类别上的准确率均有少量下降。所以，预训练可使最终训练得到的被训练网络，在旧类别上的准确率保留的稍好一些，在新类别上也有少量的准确率提升。

本文推测预训练使最终的结果稍好的原因是：若进行了新添加层的预训练，则在用本章方法进行训练时，新类别检测误差 $Loss_{new}$ 在训练的初始阶段，不会是太大的值，因为新类别添加的层已经被预训练过了。 $Loss_{new}$ 在训练初始时不太大，所以它不会对网络产生太大的改动，使网络的训练过程比较平缓，对旧类别准确率保留得好；反之，若无预训练，在用本章方法进行训练时， $Loss_{new}$ 在训练初始阶段是一个较大的值，会使网络产生较大变动，虽然保留误差 $Loss_{old}$ 尽力克制旧类别输出层输出值的变动，但在主干网络变化较大且剧烈时，它无法很好的做到这一点。

5.7 保留误差和新类别检测误差的赋权

以往的研究表明，在训练多任务的神经网络时，将不同任务的误差赋不同的权值，会影响网络的表现。

在 LwF 中，其作者探究得到：在他的图像分类任务中，对保留误差、新类分类误差赋不同的权值，可使网络往某方任务产生倾斜。具体为，若将保留误差权值赋得比新类分类误差权值高，则网络在旧类上的准确率保留的更好，在新类上的准确率变低；反之，若将保留误差权值赋得比新类分类误差权值低，则网络在旧类上的准确率保留的没有前者好，在新类上的准确率变高。

本文也想探究，在本章的方法中，对保留误差和新类检测误差赋不同的权值，对被训练网络的影响。在本节实验中，我们分别将保留误差、新类检测误差赋 0.1 的权值。将保留误差赋 0.1 的权值，即：被训练网络 RPN 模块、CLS 模块中关于旧类别的 SmoothL1 误差，乘上 0.1 的权值；将新类检测误差赋 0.1 的权值，即：将被训练网络 RPN 模块中关于新类别的 Sigmoid 交叉熵误差、SmoothL1 误差，CLS 模块中关于新类别的 Softmax 交叉熵误差、SmoothL1 误差，都乘上 0.1 的权值。

本节实验得到的结果如表 5.7 所示。表 5.7 括号中的值，是与表 5.2 中对应值进行比较的结果。

表 5.7 保留误差和新类检测误差赋权对结果的影响

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	旧类别 平均值	新类别 平均值	旧类别 -人脸	旧类别 -眼镜	旧类别 -花卉	人脸	眼镜	花卉
保留误差 乘 0.1 的权重	52.5 (-5.6)	73.8 (+2.0)	55.1	57.5	44.9	89.9	77.3	54.2
新类检测误差 乘 0.1 的权重	58.6 (+0.5)	66.2 (-5.6)	58.7	58.8	58.2	84.9	70.8	43.0

从表 5.7 的结果可看出：若将保留误差乘 0.1 的权值，则在旧类上的准确率会变低，在新类上的准确率会升高；反之，若将新类检测误差乘 0.1 的权值，则在旧类上的准确率会变高，在新类上的准确率会变低。所以，我们将保留误差应用于具有 RPN 模块的目标检测算法时，得到了和 LwF 一样的结论。即，对保留误差、新类识别误差赋不同的权值，可使网络往权值高的任务倾斜，在该任务上的准确率提升。

5.8 为新类别添加第二阶段专有层

FPN 属于两阶段的目标检测算法。对于两阶段的目标检测算法，第二阶段的输入包含第一阶段的输出。若第一阶段还未产生输出，则第二阶段不能进行。

本文在做目标检测的增量学习时，有个重要的目标：希望整个算法的速度尽可能快。如果不改动关于旧类别的计算量，则只能使关于新类别的计算量尽可能少。其本质是，必须尽可能地使用为旧类别已计算出的特征，去检测新类别。

对于两阶段的目标检测算法，在第二阶段，关于新类别的计算，不能使用关于旧类别已计算出的特征。因为关于旧类别的计算，使用的是旧类别的 RoIs。所以，新类别使用自己专有的第二阶段参数不会增加时间开销，只会增加存储开销。

对于本文使用的 ResNet-50-FPN 来说，若新类别第二阶段完全使用自己的专有层，会增加 40M 的显存开销，但这样做会产生一些准确率的提升。另外，ResNet-50-FPN 在测试时，总共的显存占用是 3100M。所以在显存充足的情况下，

可考虑在第二阶段，为新类别添加专有的所有层。

表 5.8 为给新类别添加第二阶段专有层的结果。表 5.8 括号中的值，是与表 5.2 中对应值进行比较的结果。另外，在本节的实验中，为新类别添加的所有层，都进行了预训练。

表 5.8 为新类别添加第二阶段专有层的结果

(表中数值表示 mAP@0.5 或 AP@0.5)

实验设置	旧类别 平均值	新类别 平均值	旧类别 -人脸	旧类别 -眼镜	旧类别 -花卉	人脸	眼镜	花卉
为新类别添 加第二阶段 专有层	58.3 (+0.2)	72.1 (+0.3)	58.6	58.8	57.4	89.1	75.7	51.5

从表 5.8 的结果可看出：在第二阶段（CLS 模块），为新类别添加专有的所有层，可使被训练网络在训练好后，在新、旧类别上的准确率，都比默认设置有了些提升。且本文认为，在本节实验中，训练时间不够长，若训练时间再长一些，提升应该会更多一些。

5.9 本章小结

本章提出了一个改动已有网络参数，以为其添加新的可检测类别的方法。该方法较 4.1 节方法的好处是：可在增加同样多参数、计算量的情况下，超过 4.1 节方法在新类别上的准确率。

本章提出的方法借鉴自，在图像分类领域被提出的 LwF 方法。本章率先将 LwF 的思路应用在了流行的、具有 RPN 模块的目标检测网络中。通过实验证明了，该方法可有效地为目标检测网络添加新的可检测类别。

之后，本章通过一系列实验来探讨该方法中的各种细节。研究了在面对可选择的细节做法时，应选取怎样的做法，可达到更高的准确率、可使训练更稳定。通过这些实验，得出以下结论：

(1) 保留误差极大地保留了网络在旧类别上的准确率。若不添加保留误差，直接用新类别训练集去训练已有网络，会使已有网络对旧类别的检测能力大量丧失。

(2) 回归目标不应选取为 Sigmoid 函数之前的值。因为这样会使训练不稳定，在训练阶段产生很大的误差，最终会使网络对新、旧类别的准确率均比默认设置低。

(3) 采用固定网络产生的物体候选框进行训练，会使旧类别准确率保留得更好，新类别准确率有些下降。采用被训练网络产生的新类别物体候选框，会使新、旧类别准确率均稍微下降。

(4) 不对新添加的层进行预训练，会使新、旧类别准确率均稍微下降。

(5) 对保留误差和新类别检测误差赋不同的权值，会使网络在权值高的一方准确率有一些提升，在权值低的一方准确率下降。

(6) 在第二阶段，为新类别添加专有的所有层，会使新、旧类别准确率均稍微提升。

第6章 总结与展望

随着以卷积神经网络为首的深度学习在计算机视觉中的广泛应用。图像中目标检测的准确率较以往有了很大提高。使越来越多应用了目标检测技术的产品出现在了生活中，方便和丰富了人们的生活。安防、门禁、辅助救援、无人驾驶、智能家居、休闲娱乐很多地方都可见到目标检测的身影。图像中目标检测的准确率也逐渐提升至快饱和的状态。然而，在这种情况下，人们对人工智能、计算机视觉又提出了更高的要求，一个个面向更智能的问题接踵而至。增量学习，是其中一个逐渐火热的话题。

鉴于这样的背景，本文研究了增量的目标检测任务。本文首先讲述了目标检测算法、基于卷积神经网络的增量学习这两方面，国内外的研究进展，并详细地列举、比较了前人的研究工作。随后，讲述了本文基于的方法和技术。接着讲述了本文研究的两个方法。

第一个方法是基于特征提取思想的方法。该方法是直观的、已有的方法。为了提升该方法的准确率，本文做了如下工作：（1）实验探究得出：为新类别添加的专有层数越多，网络在新类别上的准确率越高，且准确率的增益随着添加层数的增多而减少。（2）实验探究得出：若已有网络能检测的类数越多，那么用该方法为其添加新的可检测类别时，会在新类别上达到越高的准确率。（3）本文提出：将新类别专有层添加的位置前移，可使新类别准确率有可观的提升。（4）本文提出：为新类别添加浅层旁路，可使新类别准确率有一些提升。

第二个方法是基于保留损失函数的方法。该方法是本文提出的方法。本文将在图像分类领域被提出的 LwF 方法，应用在目标检测网络上形成此方法。率先实现了将 LwF 的思路应用于流行的、具有 RPN 模块的目标检测网络中。之后，本文详细研究了该方法中的各种细节，它们是：回归目标的选取、物体建议框的选取、预训练的重要性、保留误差和新类别检测误差的赋权、为新类别添加第二阶段专有层。本文通过实验，对该方法内的这些细节做了详细的研究，为每个细节

都挑出了最好的选择。

未来工作展望。本文在为已有网络添加新的可检测类别时，详细研究并提出了添加一种类别时，多个可增加准确率的方法。未来，可以在依次添加多个类别的情况下进行实验，以探究本文提出的方法、细节在这些情况下的有效性。

参考文献

- [1] Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009: 248-255.
- [2] 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.
- [3] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in Neural Information Processing Systems, 2012: 1097-1105.
- [4] Stanford Vision Lab. Large Scale Visual Recognition Challenge 2012 Result s[EB/OL]. <http://image-net.org/challenges/LSVRC/2012/results.html>, 2012-10-1 3.
- [5] Du P, Weber R, Luszczek P, et al. From CUDA to OpenCL: Towards a performance-portable solution for multi-platform GPU programming[J]. Parallel Computing, 2012, 38(8): 391-407.
- [6] Everingham M, Van Gool L, Williams C K, et al. The pascal visual object classes (voc) challenge[J]. International Journal of Computer Vision, 2010, 88(2): 303-338.
- [7] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common objects in context[C]. Proceedings of the European Conference on Computer Vision, 2014: 740-755.
- [8] Ross G, Ilija R, Georgia G, et al. Detectron[EB/OL]. <https://github.com/facebookresearch/Detectron>, 2018.
- [9] Viola P A, Jones M. Rapid object detection using a boosted cascade of simple features[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001: 511-518.
- [10] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,

- 2005: 886-893.
- [11]Felzenszwalb P F, Girshick R B, Mcallester D A, et al. Object Detection with Discriminatively Trained Part-Based Models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(9): 1627-1645.
- [12]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.
- [13]Uijlings J R, Van De Sande K E, Gevers T, et al. Selective search for object recognition[J]. International Journal of Computer Vision, 2013, 104(2): 154-171.
- [14]Zitnick C L, Dollár P. Edge boxes: Locating object proposals from edges[C]. Proceedings of the European Conference on Computer Vision, 2014: 391-405.
- [15]Girshick R. Fast R-CNN[C]. Proceedings of the IEEE International Conference on Computer Vision, 2015: 1440-1448.
- [16]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, 2014: 346-361.
- [17]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.
- [18]Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector[C]. Proceedings of the European Conference on Computer Vision, 2016: 21-37.
- [19]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.
- [20]Dai J, Li Y, He K, et al. R-FCN: Object detection via region-based fully convolutional networks[C]. Advances in Neural Information Processing Systems, 2016: 379-387.
- [21]Lin T Y, Dollár P, Girshick R B, et al. Feature pyramid networks for object detection[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 936-944.

- [22] Shen Z, Liu Z, Li J, et al. DSOD: Learning deeply supervised object detectors from scratch[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 1919-1927.
- [23] Bodla N, Singh B, Chellappa R, et al. Soft-NMS—improving object detection with one line of code[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 5562-5570.
- [24] 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: 2999-3007.
- [25] 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.
- [26] McCloskey M, Cohen N J. Catastrophic interference in connectionist networks: the sequential learning problem[J]. Psychology of Learning and Motivation, 1989: 109-165.
- [27] Goodfellow I J, Mirza M, Xiao D, et al. An empirical investigation of catastrophic forgetting in gradient-based neural networks[C]. International Conference on Learning Representations, 2014.
- [28] Li Z, Hoiem D. Learning without forgetting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(12): 2935-2947.
- [29] Buciluă C, Caruana R, Niculescu-Mizil A. Model compression[C]. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006: 535-541.
- [30] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015.
- [31] Zacarias A, Alexandre L A. SeNA-CNN: Overcoming Catastrophic Forgetting in Convolutional Neural Networks by Selective Network Augmentation[C]. Artificial Neural Networks in Pattern Recognition, 2018: 102-112.
- [32] Jung H, Ju J, Jung M, et al. Less-forgetting learning in deep neural networks[J]. arXiv preprint arXiv:1607.00122, 2016.

- [33]Mallya A, Davis D, Lazebnik S. Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights[C]. Proceedings of the European Conference on Computer Vision, 2018: 67-82.
- [34]Castro F M, Marinjimenez M J, Guil N, et al. End-to-end incremental learning[C]. Proceedings of the European Conference on Computer Vision, 2018: 241-257.
- [35]Rebuffi S, Kolesnikov A, Sperl G, et al. iCaRL: Incremental classifier and representation learning[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 5533-5542.
- [36]Welling M. Herding dynamical weights to learn[C]. International Conference on Machine Learning, 2009: 1121-1128.
- [37]Gidaris S, Komodakis N. Dynamic few-shot visual learning without forgetting[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 4367-4375.
- [38]Kirkpatrick J, Pascanu R, Rabinowitz N C, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2017, 114(13): 3521-3526.
- [39]Huszár F. Note on the quadratic penalties in elastic weight consolidation[J]. Proceedings of the National Academy of Sciences of the United States of America, 2018, 115(11).
- [40]Shmelkov K, Schmid C, Alahari K, et al. Incremental learning of object detectors without catastrophic forgetting[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 3420-3429.
- [41]Arbelaez P, Ponttuset J, Barron J T, et al. Multiscale combinatorial grouping[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014: 328-335.
- [42]Guan L, Wu Y, Zhao J, et al. Learn to detect objects incrementally[C]. IEEE Intelligent Vehicles Symposium, 2018: 403-408.
- [43]Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]. International Conference on Machine Learning, 2010: 807-814.
- [44]Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by

- reducing internal covariate shift[C]. International Conference on Machine Learning, 2015: 448-456.
- [45]Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[C]. International Conference on Learning Representations, 2015.
- [46]Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1-9.
- [47]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.
- [48]He K, Sun J. Convolutional neural networks at constrained time cost[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 5353-5360.
- [49]Srivastava R K, Greff K, Schmidhuber J. Highway networks[J]. arXiv:1505.00387, 2015.
- [50]Bishop C M. Neural networks for pattern recognition[M]. Oxford university press, 1995.
- [51]Ripley B D. Pattern recognition and neural networks[M]. Cambridge university press, 2007.
- [52]Venables W N, Ripley B D. Modern applied statistics with S-PLUS[M]. Springer, 2013.
- [53]He K, Gkioxari G, Dollár P, et al. Mask R-CNN[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 2980-2988.

攻读硕士学位期间主要的研究成果

已发表或已投稿论文：

- [1] Shihao Wang, Wei Zhang, Sophanyouly Thachan, and Yuntao Qian. Incremental Learning of Object Detection Based on Model Distillation[C]. IEEE International Conference on Image Processing (ICIP), 2019. (已投稿, 在评审)
- [2] Wei Zhang, Shihao Wang, Sophanyouly Thachan, Jingzhou Chen, and Yuntao Qian. Deconv R-CNN for Small Object Detection on Remote Sensing Images[C]. IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2018: 2483-2486.

参与科研项目：

- [1] 参与实验室遥感图像中船舶等物体的检测、识别、跟踪项目。
- [2] 参与实验室人脸识别项目。

致谢

行文至此，已是本文结尾。回顾这两年半的研究生生涯，感触良多。

自己从刚踏入浙江大学开启研究生生涯的懵懂少年，到如今即将走上工作岗位，给前面十多年的生涯划上了一个句号，不禁感慨时光飞逝。

这两年多，收获最大的当属对于科研的认识。在读研前没有怎么接触过科研，对科研抱有一种神秘与畏惧之心。在读研后，对科研有了一定的认识，其中最为印象深刻的，当属钱法涛导师对我科研过程中的教导。

接下来要感谢实验的同学们：廖丹萍师姐、罗志坚师兄、方佳璐师姐、熊凤超师兄、顾兆伦师兄、许洋洋师兄、沈森垚师兄、胡忠闯师兄、周佩林师兄。以及陈璟洲、张伟、陈思宇、汪俊、Yously、Reehan Ali Shah,，以及徐倩、钟倩、刁莹煜、郝萌、范小天师弟师妹们。大家关系融洽，宛如一个大家庭。在学术上一起讨论，生活上互帮互助，共同地为实验室的科研、项目不断努力，共同进步，感谢大家。

同时感谢寝室的各位，感谢来到浙江大学后认识的小伙伴们，陪我度过了这段研究生的岁月。聚餐，看电影，爬山，一起锻炼，正因为有你们的陪伴，生活才不会显得单调。友谊长存。

感谢我的父母，你们的养育、支持、包容是我这一生最宝贵的财富；还要感谢我的女朋友邹涵江，在读研、做毕设期间她给予了我莫大的支持和鼓励。

最后我还要感谢浙江大学给我提供了这次攻读硕士研究生的机会，提供了高水平的学习和学术环境，以及舒适便利的生活环境，让这两年半的研究生生涯成为我人生中宝贵的记忆。

感谢各位，致以诚挚的感谢，谢谢！

王世豪

2019年1月于浙大玉泉