存档编号 ­

华北水利水电大学

North China University of Water Resources and Electric Power

毕 业 设 计

题目 基于TensorFlow的快速对象检测

学 院 信息工程学院

专 业 计算机科学与技术

姓 名 王希晨

学 号 201516402

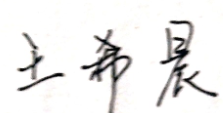
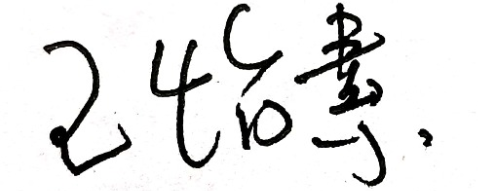
指导教师 王怡素

完成时间 2020年5月25日

教务处制

# 独立完成与诚信声明

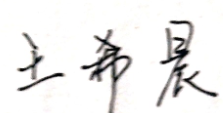
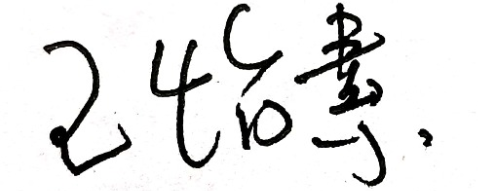
本人郑重声明：所提交的毕业设计（论文）是本人在指导教师的指导下，独立工作所取得的成果并撰写完成的，郑重确认没有剽窃、抄袭等违反学术道德、学术规范的侵权行为。文中除已经标注引用的内容外，不包含其他人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确的说明并表示了谢意。本人完全意识到本声明的法律后果由本人承担。

毕业设计（论文）作者签名： 指导导师签名：

签字日期：2020.6.10 签字日期：2020.6.10

# 毕业设计（论文）版权使用授权书

本人完全了解华北水利水电大学有关保管、使用毕业设计（论文）的规定。特授权华北水利水电大学可以将毕业设计（论文）的全部或部分内容公开和编入有关数据库提供检索，并采用影印、缩印或扫描等复制手段复制、保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交毕业设计（论文）原件或复印件和电子文档（涉密的成果在解密后应遵守此规定）。

毕业设计（论文）作者签名： 导师签名：

签字日期：2020.6.10 签字日期：2020.6.10

# 目录

[摘要 I](#_Toc44012246)

[Abstract II](#_Toc44012247)

[第一章 绪论 II](#_Toc44012248)

[1.1 研究背景与意义 1](#_Toc44012249)

[1.2 国内外研究现状 2](#_Toc44012250)

[1.2.1 传统检测算法: 2](#_Toc44012251)

[1.2.2 基于深度学习结合区域候选的分类算法 2](#_Toc44012252)

[1.2.3 基于深度学习的回归算法 3](#_Toc44012253)

[1.3 本文组织结构 4](#_Toc44012254)

[第二章 相关理论 5](#_Toc44012255)

[2.1 神经网络 5](#_Toc44012256)

[2.1.1 起源 5](#_Toc44012257)

[2.1.2 多层神经网络 6](#_Toc44012258)

[2.1.3 反向传播 6](#_Toc44012259)

[2.1.4 激活函数类型 7](#_Toc44012260)

[2.2 卷积神经网络 8](#_Toc44012261)

[2.2.1 应用背景 8](#_Toc44012262)

[2.2.2 基本结构 8](#_Toc44012263)

[第三章 算法分析 10](#_Toc44012264)

[3.1发展历程 10](#_Toc44012265)

[3.1.1 Yolov1 10](#_Toc44012266)

[3.1.2 Yolov2 10](#_Toc44012267)

[3.1.3 Yolov3 10](#_Toc44012268)

[3.2 网络结构 11](#_Toc44012269)

[3.2.1 Darknet-53 后端网络 11](#_Toc44012270)

[3.2.2 YOLO前端网络 12](#_Toc44012271)

[3.3 算法细节 14](#_Toc44012272)

[3.3.1 先验框提取 14](#_Toc44012273)

[3.3.2 参数解码 15](#_Toc44012274)

[3.3.3 正负样本分配: 16](#_Toc44012275)

[3.3.4 损失函数: 17](#_Toc44012276)

[第四章 算法实现 19](#_Toc44012277)

[4.1 数据预处理模块: 19](#_Toc44012278)

[4.2 模型模块: 22](#_Toc44012279)

[4.3 训练模块: 23](#_Toc44012280)

[4.4 评估模块 24](#_Toc44012281)

[4.5 检测模块: 25](#_Toc44012282)

[第五章 评估与测试 27](#_Toc44012283)

[4.1 实验环境 27](#_Toc44012284)

[4.2 数据集 27](#_Toc44012285)

[4.3 评估标准 28](#_Toc44012286)

[4.4 评估结果与可视化 29](#_Toc44012287)

[第六章 总结与展望 30](#_Toc44012288)

[5.1 总结 30](#_Toc44012289)

[5.2 展望 31](#_Toc44012290)

[参考文献 32](#_Toc44012291)

[致 谢 34](#_Toc44012292)

[附 录 1](#_Toc44012293)

[附录一 外文原文 1](#_Toc44012294)

[附录二 外文译文 7](#_Toc44012302)

[附录三 任务书 13](#_Toc44012306)

[附录四 开题报告 14](#_Toc44012312)

[附录五 相关程序代码 17](#_Toc44012313)

基于TensorFlow的快速对象检测

摘要

对象检测是计算机视觉领域中的基础任务，其主要目的是根据图像或视频中的物体进行分类与定位。由于图像场景中光照，遮挡，物体大小不同，使得对象检测任务困难重重。传统时代基于构建特征工程提取图像特征并使用机器学习算法对其进一步分类的检测算法由于提取的特征有限并且受计算机硬件水平的制约，使得对象检测算法只局限于特定的领域。深度学习时代，由于硬件水平的提升，特别是GPU运算能力的提高，加速了神经网络在各个领域的应用。自从2012年AlexNet在ImagineNet图像分类挑战赛上取得了突破性的成果[LW]，卷积神经网络（CNN）重新受到到人们的关注。2014年R-CNN[LW]的问世，使得卷积神经网络第一次应用在对象检测领域。2016年RadmonJ, Farhadi等研究人员提出的YOLO[LW]检测算法,将分类与边界框回归合并为一个阶段，开创了单阶段检测器的先河。2018年作者提出的YOLOv3[LW]检测框架，在保持检测精度相近的同时，其检测速度远超同期的对象检测算法。

本文主要基于TensorFlow机器学习框架，基于COCO数据集[LW]采样后的子集实现了YOLOv3检测算法，并进一步从输入，模型，训练，评估, 检测五个模块构建了一个小型的对象检测系统。

**关键词**: 对象检测; YOLOv3; TensorFlow; 快速检测

# Fast Object Detection Based On TensorFlow

Abstract

Object detection is the basic task in the field of computer vision. It mainly locates and classifies the objects in the image or video. Because of the illumination， occlusion and different size of the target object in the scene， the task of object detection is difficult. The traditional object detection algorithm based on manual feature extraction and machine learning classification is difficult to mine more abundant features and has a large amount of calculation， which makes the object detection difficult to achieve the characteristics of real-time detection and the detection accuracy is not ideal. In the era of deep learning， due to the development of hardware， the improvement of data storage and the application of convolutional neural network， the task of object detection has made a key breakthrough， and the detection accuracy and speed have been greatly improved.

Yolov3 object detection algorithm [1] is the third improved version of Yolo algorithm proposed by Redmon J， Farhadi and other researchers in 2018. It is based on the development of darknet-53 network and programmed in C language. When the accuracy is slightly lower than some two-stage detection algorithms based on window extraction and classification， its detection speed is far faster than that of the object detection algorithm in the same period. Based on TensorFlow framework， this paper implements YOLOv3 object detection algorithm， which can achieve real-time detection speed at present. It realizes five modules: data preprocessing， model building， model training， prediction and evaluation. On the Pascal VOC dataset [2]， 87% of the detection accuracy is achieved based on its official evaluation standard. And its detection speed can reach 45 frames per second， which can achieve the effect of fast detection.

\_­­­­­­

**Keywords:** Object detection; YoLov3; TensorFlow; Fast detection

1. 绪论
   1. 研究背景与意义

在计算机视觉领域，对象检测是在对象分类基础上实现对象定位的基础课题，也是实现对象跟踪，图像场景语义信息理解的前提。对象检测将对象分类与对象定位结合起来，不仅需要正确识别对象类别，还需要输出对象的位置和大小信息。与分类任务相比，面对图像中光照，物体之间遮挡等复杂场景，使对象检测更具难度。

对象检测的研究可分为泛化目标检测和应用型目标检测两个课题， 前者旨在探索统一框架下的检测场景来模拟人类的视觉认知，后者则是在特定应用下的检测场景， 如行人检测，人脸检测，文本检测等。相比于传统基于构建复杂的手工特征提取器， 近年来深度学习技术的快速发展使对象检测取得了显著突破[3ＬＷ]。

传统的对象检测基于特征工程并结合机器学习分类器实现，如VJ检测器［ＬＷ］。但是特征工程是手工实现并且特征设计需要很多经验与技巧，加上当时计算机硬件的限制，需要大量优化计算的技巧，使得系统的设计十分复杂。 深度学习时代，与传统人工构建特征提取的方式不同，利用卷积神经网路学习图像特征，极大降低了构建特征提取器的难度，并且能够提取更丰富的特征信息， 同时结合机器学习分类算法的预测结果，极大提高了检测的精度和速度，但检测速度与精度受限于对象检测器的构建方式，往往需要在二者之间权衡。

在无人驾驶，视频监控，机器导航等领域，往往通过构建特定的自动化流程以减少人力资本的消耗。其中对象检测在这些领域发挥了重要作用。例如，无人驾驶中需要对传感器采集的图像定位周围车辆的位置，交通信号灯的状态，之后将这些信息传送到相应系统中来及时调整车辆的行进方向与速度。并且，在实时性要求高的场景中，对对象检测的速度提出了更高的要求。因此，构建一个端到端的检测系统就显得尤为必要。

### 

* 1. 国内外研究现状

由于看待检测任务思路的不同，对象检测算法大体上可以分为三类: 传统检测算法， 将分类与边界框回归分离的二阶段检测算法，将分类与边界框回归结合的一阶段回归算法。

* + 1. 传统检测算法

传统检测算法通常使用不同大小的窗口以一定的步幅滑动整张图像来提取候选区域，在候选区域内提取一类或多类局部特征，例如 HOG， SIFT等特征，最后应用支持向量机或逻辑岭回归等机器学习分类算法对候选区域分类来实现对象检测算法。代表性算法如Viola-Jones检测器［ＬＷ］， HOG行人检测器［ＬＷ］， 可变形部件模型(DPM)［ＬＷ］等。这类算法最主要的问题是由于手工构建的特征提取器与机器学习分类器两者很难同时达到最优，使得传统检测算法迟迟无法取得阶段性突破。

* + 1. 二阶段检测算法

2012年，Hinton课题组使用基于卷积神经网络的Alex Net在ImageNet图像识别比赛中夺冠， 卷积神经网络在计算机视觉领域得到了重视， 此后深度学习在计算机视觉领域内多个方向得到了广泛的应用，对象检测算法也得到了突飞猛进的发展， 相比于传统对象检测算法， 深度学习能够通过深层次的模型计算在海量数据中提取更为复杂特征结构。代表性算法如R-CNN［ＬＷ］，Fast R-CNN[ＬＷ]，Faster R-CNN［ＬＷ］等。

**R-CNN**：突破了传统的目标检测算法的思想，为深度学习在目标检测领域的首次成功突破。该算法主要包括区域候选、归一化处理、特征提取、分类及回归等步骤。首先，使用选择性搜索算法从原始图片中区域提取适量候选区域。然后，将候选区域进行尺度归一化，并通过预训练的卷积神经网络提取候选区域的目标特征表达，特征层之后是全连接层，并使用SVM作为分类器。R-CNN算法在目标检测精度上相较传统方法有了质的提升，但是也存在不少缺点，如候选框数量众多导致运算量庞大，效率低等特点。

**Fast-RCNN**：在R-CNN的基础上，采用自适应尺度池化对整个网络进行优化，规避了R-CNN中冗余的特征提取操作，提高了网络识别的准确率。此外，使用感兴趣区域池化层，用以提取特征层上各个候选框的固定维度的特征表示。同时，使用 SoftMax 非线性分类器，以多任务学习的方式同时进行分类和回归。由于Fast R-CNN 无需存储训练和测试过程产生的中间值，因此其速度相较于 R-CNN 大为提升。

**Faster R-CNN**：Faster R-CNN通过构建区域建议网络(Region Proposal Network)提取候选框，取代时间开销大的选择性搜索方法，区域提名、分类、回归等操作一起共用卷积特征，进一步提升了检测速度。

* + 1. 一阶段检测算法

基于深度学习的回归方法很好的解决了检测速度的问题，由于不需要事先提取候选框而直接对对象的位置与类别回归，不同于二阶段因此此类方法又被称为一阶段算法。代表性算法如Yolo算法，SSD算法等。

**YOLO**: YOLO检测算法使用Darknet特征提取网络提取图像中的特征图，特征图中的每一点对应原图中的网格，每个网格预测类别标签，边界框偏移与置信度这三类参数。在其后的版本中又引入了基于聚类选择的先验框以及多尺度预测，提高了对小物体的检测精度。

**SSD**: SSD检测算法不产生任何候选区域，也不涉及对图像的重采样。它通过一次卷积神经网络的传播来实现对象检测。算法首先选择一组不同宽高比的先验框，先验框参数并不是通过聚类产生而是手工确定的。随后，网络通过预测边界框与先验框的偏移而不是直接预测其边界框的参数来提高检测精度。并且，该算法通过使用许多不同深度的卷积层输出的特征图作为分类器的输入来预测不同尺度目标对象边界框参数。与YOLO算法一个最大的区别是，YOLO算法仅仅当训练样本中的区域存在物体时才会预测类别概率，而SSD是预测一个类别存在于先验框中的概率, 并且在训练中使用了在线困难样本挖掘技术对负样本进行过滤。

* 1. 本文组织结构

第一章为绪论部分。阐述了论文的研究背景与意义以及国内外研究现状并比较了传统检测算法与单阶段, 二阶段检测算法这三种不同类别的对象检测算法的优缺点。

第二章介绍基于卷积神经网络的对象检测算法的相关理论背景，对相关概念进行简要说明。

第三章分析YOlOv3算法的基本原理，并对算法中的细节进行了深入研究。

第四章从数据输入，模型，训练，评估，检测四个模块的功能出发，详细说明了检测系统的实现。

第五章对评估所用数据集及评估方法与评估结果做出了分析。

第六章作为总结与展望，总结了在算法实现过程中面对的问题以及收获，并展望了下一步的方向。

1. 相关理论

在本章主要阐述基于深度学习的对象检测算法的相关理论背景，主要包括神经网络，卷积神经网络，梯度下降算法，激活函数 四个方面。

* 1. 神经网络

神经网络是一种机器学习模型，CNN是其中的一个特例， 在介绍CNN之前，先说明一些简单的神经网络的工作模式。

* + 1. 起源

由于神经网络最初是人们为了模拟人类神经元而构建的，因此，神经网络在早期也被称为人工神经网络。尽管神经网络来源于生物学上的灵感，但是神经网络的工作模式与人类神经元并没有内在的联系。人脑约有1000亿个神经元在并行运行，而人工神经元却大都是在计算机上串行计算的数学函数[8LW]。图2-1是一个最基本的二分类线性网络。

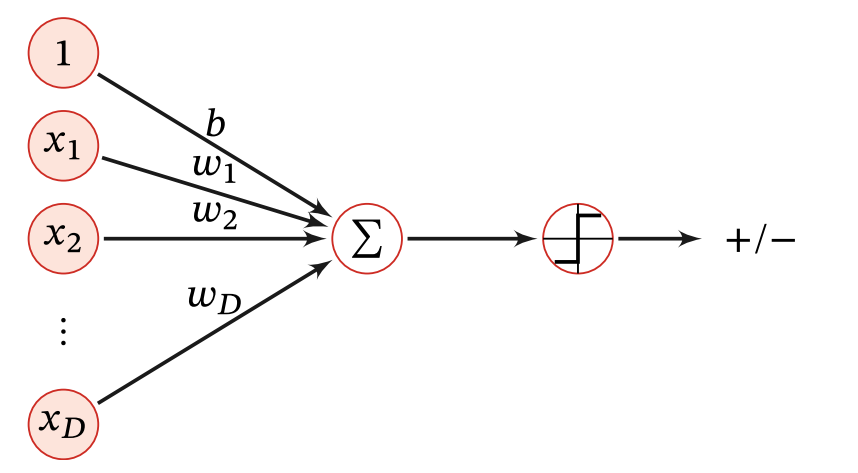


图2-1 二分类线性网络结构图

其中，参数个数为， 神经元由个权值为的参数组成，输入参数与神经元线性求和，最后将结果输出到激活函数中来产生相应的输出。

* + 1. 多层神经网络

典型的多层全连接网络如图2-2所示。一个多层全连接的网络通常包括三种类型的层：输入层，隐藏层，输出层。其中大多数计算发生在隐藏层，输出层将隐藏层激活转换为输出。一个至少有一个隐藏层的多层前馈网络理论上可以拟合任何函数[LW]。

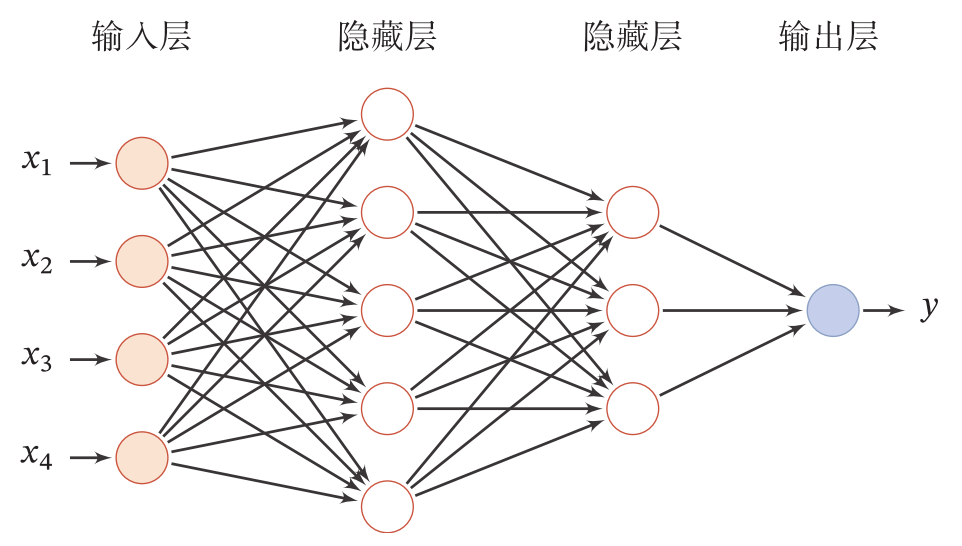


图2-2 典型的两层全连接网络

* + 1. 卷积神经网络

在这一节，主要介绍卷积神经网络在计算机视觉应用的原因及其基本思想。

使用传统的神经网络解决计算机视觉问题的困难在于，即使是一个中等大小的图像也包含了大量的信息。以620x480的图像为例，其中包含620x480x3=297600像素。如果将该图像的每个像素组成一个向量单独输入到一个全连接网络中，每个神经元需要297600个权值。一个1920x1080的图像需要一个大小为2073600的向量。由此可见，随着图像尺寸的增大，网络中自由参数的总体数量迅速变得极大。过大的模型会导致过拟合和性能下降。

卷积神经网络的基本思想是受生物学中感受野概念的启发。感受野是动物视网膜上的神经元收到特定刺激后，将神经冲动传到大脑皮层所能激发的区域. 当看到图像时，视网膜只对空间变化或者时间变化做出反应而忽略了图像缺乏这些特征的部分，因此可被看作为特征检测器.为衡量景物的变化，例如边缘变化，位置变化，需要对像素之间建立横向的联系，即神经元细胞之间的联系。而建立这种联系的过程可在计算机种使用卷积运算来模拟。图像处理中，可以通过卷积对图像进行滤波，图2-3显示了使用手动选择的Sobel算子进行滤波来检测水平边缘， 其功能类似于感受野。



图2-3 使用Sobel算子进行水平边缘检测

一组卷积过滤器(filter)可以组合成一个神经网络的卷积层。然后利用机器学习将过滤器的值作为参数进行训练。卷积运算代替了常规神经网络层的乘法运算。该层输出高度和宽度取决于卷积的步长， 深度取决于过滤器的数目。

由于对图像的所有部分使用相同的卷积核，与全连接层相比，自由参数的数量大大减少。卷积层的神经元大多共享相同的参数，只连接到输入的局部区域。由卷积产生的参数共享保证了图像的平移不变性。描述卷积层的另一种方法是将其看作为一个全连接层，它的权值有一个无极强的先验值， 这个先验要求隐藏单元的权值必须与其相邻单元的权值相同但在空间中改变。

卷积神经网络通常由卷积层， 批量归一化(Batch Normalization)层， 非线性激活函数， 池化(Pooling)层堆叠组成。一个简单的卷积网络结构如图2.4所示。将卷积神经网络应用在对象检测任务时，网络中较浅的层能够学习到图像的底层特征因此具有较小的感受野，而深层的网络融合了浅层网络学习到的多种特征， 具有较大的感受野。

* 1. 梯度下降算法

神经网络通过更新所有神经元权值进行训练从而学习从输入到目标输出的映射关系，反向传播算法为多层网络中神经元权值的迭代更新提供了简单有效的方法。经典的优化算法是梯度下降算法，在算法的第一阶段，初始化神经元的权重，输入向量通过神经网络前向传播，并保留一些中间变量，通过使用损失函数衡量网络的输出与期望的输出之间的差异并计算损失函数的梯度。之后通过应用链式法则使用前向传播中保留的中间值将损失函数值的梯度传播回去。并使用一个比率乘上梯度从权重中减去这一部分来更新神经元的权重，这个比率也被称为学习率。反向传播算法主要是基于局部估计来最小化损失函数，因此有可能出现基于单个数据的梯度方向不断改变而不会问题，另一种方式是基于小批量数据进行更新权重，并且还有一些改进的算法例如带动量的梯度下降，RMSProp， Adam等算法可以使损失更快地收敛。

* 1. 激活函数

激活函数决定了每个神经元的最终输出。为了构建一个有效的网络，正确地选择激活函数类型显得尤为重要。早期的研究人员发现感知机和其他线性分类系统无法解决线性不可分的问题，比如异或问题。有时，虽然可以通过使用手工制作的特征检测器与分类器来解决这类问题，但这并没有充分利用机器学习自动化的特点。即使在线性分类器中添加再多的线性层也对此毫无帮助， 因为由一系列线性层组成的网络的表达能力与一层线性网络的表达能力相同。非线性整流层（ReLU）是目前最常用的构建非线性网络的激活函数，其数学表达式如下所示。

ReLu函数易于计算和求导，逐渐取代了过去常用的Sigmoid激活函数。Sigmoid函数表达式如下：

由于在零点附近其导数趋近于0， 因此在使用Sigmoid构建网络时容易出现梯度消失的问题。

对于多类分类问题，SoftMax激活函数[]通常用于网络的最后一层输出结果：

SoftMax以长度为K的向量为输入并输出一个0到1之间的概率，其输出值可以用作类别概率。

1. 算法分析

本章对Yolov3对象检测算法进行分析，以模型结构图的形式对网络的后端与前端进行了拆分，对网络中基本模块进行了分析，随后对算法中的先验框提取， 特征融合，损失函数进行了研究.

* 1. 发展历程
     1. YOLOv1

2015年 Redmon J等研究者提出YOLOv1算法[11]，基本思想是将生成候选框与分类回归合并成一个步骤， 预测时特征图被分成 7x7个单元格， 对每个单元格进行预测， 大大降低了计算复杂度， 加快了目标检测的速度， 但是由于一个单元格只能预测两个框和一个类别， 这种空间约束必然会限制预测的数量；而且模型根据数据预测边界框，很难将其推广到具有新的或不同寻常的宽高比。

* + 1. YOLOv2

YOLOv1虽然检测速度快，但在定位方面不够准确，并且召回率较低， 时隔一年，Redmon J 再次提出了以Darknet-19特征提取网络为后端的YOLOv2[12]算法。为了提升定位准确度， 改善召回率， YOLOv2在YOLOv1的基础上提出了几种改进策略， 一是借鉴了Faster R-CNN的思想预测目标框的相对偏移而不是直接预测目标框的位置从而简化了计算并且使模型预测损失更快收敛。二是使用k-means算法对训练集上对目标边界框进行聚类产生合适的先验框。， 根据先验框预测目标框的宽度高度信息， 三是使用了融合特征图， 在深层特征图中堆叠浅层特征图的特征， 使模型有了细粒度的特征。

* + 1. YOLOv3

YOLOv3使用了更强的特征提取网络Darknet-53, 含有53个卷积层.并且借鉴ResNet的思想， 基于残差块构建卷积网络.同时YOLOv3将YOLOv2的融合堆叠浅层网络特征图的方式扩展到从三个尺度提取特征图， 在每个特征图上分别预测， 将9个先验框均匀分布到大， 中， 小三中尺度特征图中，增加了对多种尺度的目标对象检测的准确率。

* 1. 网络结构

YOLOv3网络分两部分组成， 第一部分是作为后端网络的Darknet-53， 第二部分则是YOLO层， 通过与FPN(特征金字塔网络)类似的结构将三个尺度的特征图分别输出， 用于检测大， 中， 小三个尺度的不同尺寸的物体

* + 1. Darknet-53 后端网络

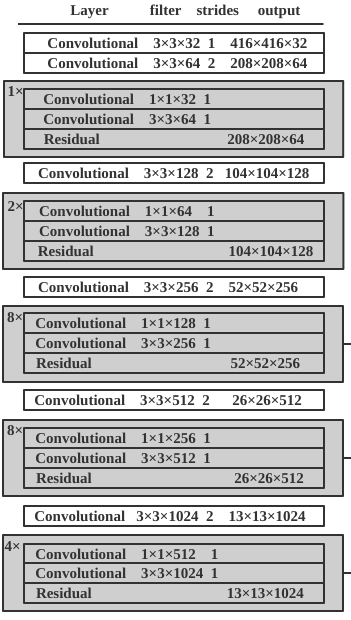


图3-1 输入图像尺寸为416， 416去除最后一层全连接的Darknet-53

1. 卷积层:

Darknet53只使用了3x3和1x1两种尺度的卷积核， 并且使用了1和2两种步幅进行卷积， 3x3x1卷积用于提取特征， 使用3x3x2卷积来降采样而没有使用池化层， 1x1卷积用于降低通道维数并且实现了跨通道信息的整合。

(2) 残差层:

Yolov3借鉴了ResNet中构建残差块的思想。何凯明等研究者在2015年正式提出了ResNet， 相比于之前卷积与池化相互堆叠的网络，在残差块中引入了短路连接， 这主要是解决训练深层网络时网络的退化问题，直觉上人们认为，在构建卷积神经网络时，层数越深，其网络可以进行到更复杂的特征提取，模型可以取得更好的效果。虽然有批量归一化层和Dropout层在很大程度上避免反向传播中梯度消失或爆炸的问题， 但是作者发现随着网络加深， 训练误差也会提高.而残差块的思想是构建一个恒等映射关系H(x) = f(x) + x， 将前两层的映射直接短路连接到第三层的输出构造一个基本的残差块， 再将这些残差块串联形成更深的网络， 被短路连接的卷积核只需要学习H(x)-x， 比直接学习H(x)要容易的多， 串联起来后在更深层的网络融合了浅层特征。与ResNet不同之处在于， YOLOv3先使用1x1卷积降低通道维数再通过3x3卷积操作提取特征并提高维度与短接层维度一致。

* + 1. YOLO前端网络

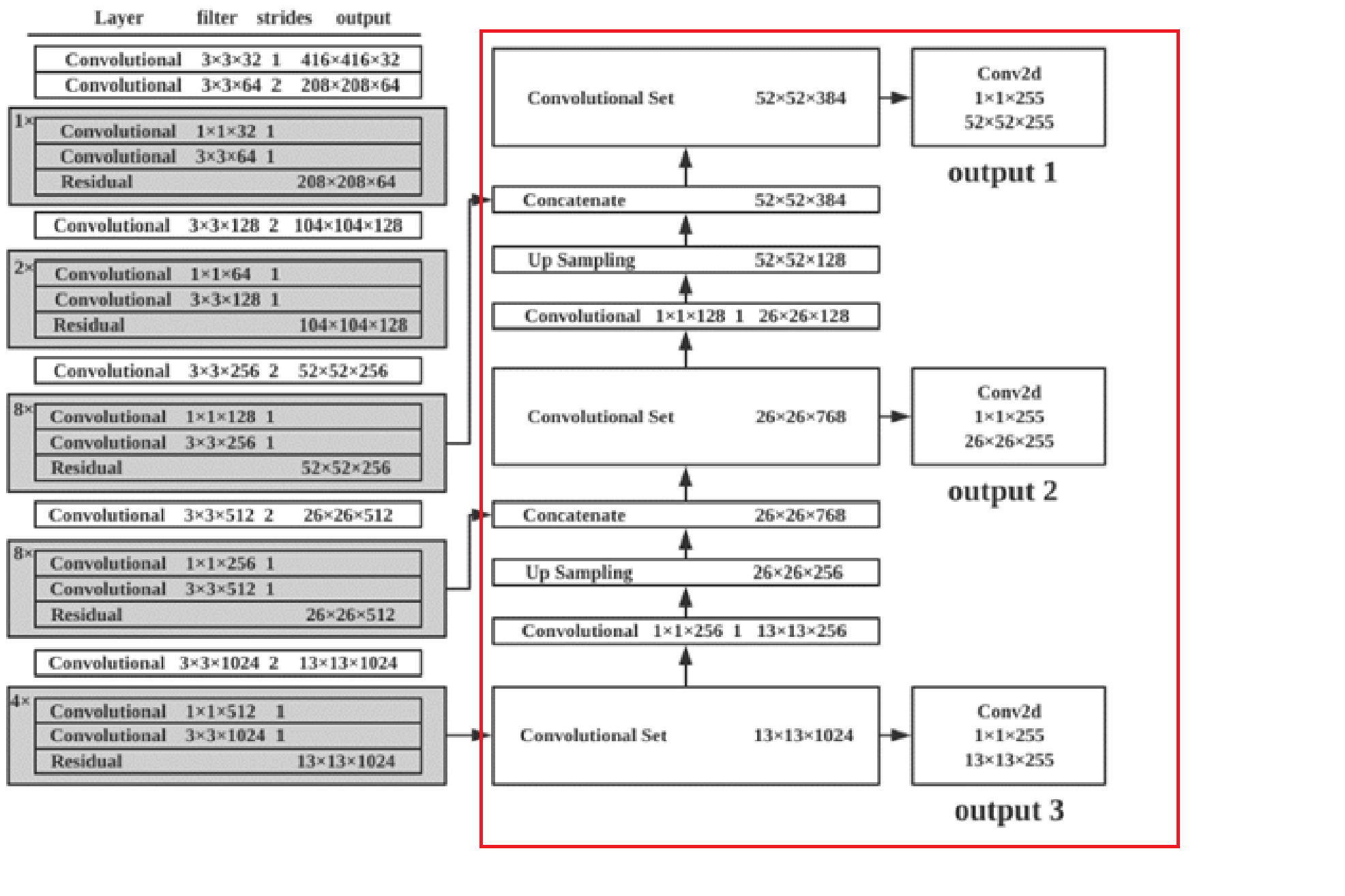


图3-2 右半边部分为Yov3前端网络

图像信息经过Darknet-53后端网络前向传播提取特征后，YOLO层负责将提取到的特征进行融合来产生大中小三个尺度的特征图。对于基于深度学习的对象检测器，卷积神经网络中较深层次的层往往能提取到较高层的特征，具有较高的感受野却丢失了位置信息，而浅层次的网络具有较低的感受野，能够提取到部分位置的特征，因此将深层网络的输出特征与较低层网络的输出特征进行融合可获得更丰富的特征信息.但是由于层与层之间输出特征的尺寸不同， 因此需要通过上采样或下采样的方式调整到同一尺寸再对特征进行拼接融合.Yolov3使用高层特征上采样与经过下采样的浅层特征拼接产生三个尺度的特征图， 并通过1x1卷积实现全连接将通道维数固定在255， 极大减少了计算量与模型内存占用， 由于Yolov3论文中使用COCO[8]数据集进行结果评估， COCO数据集一共有80个类别， 加上预测目标框经过编码后的X，Y，W，H四个坐标以及用于判断是否是背景还是前景的信度分数， 共85个参数， 并且由于每个尺度的特征图分配了3个先验框， 因此对于3个输出特征图的每一个格点需要输出3x(1+4+80)=255维度的特征向量， 将输出的三个特征图格点还原到原始图像上如下， 每个网格负责预测边界框中心落在其中的对象， 三个尺度特征图在原图像上感受野图3-3所示：

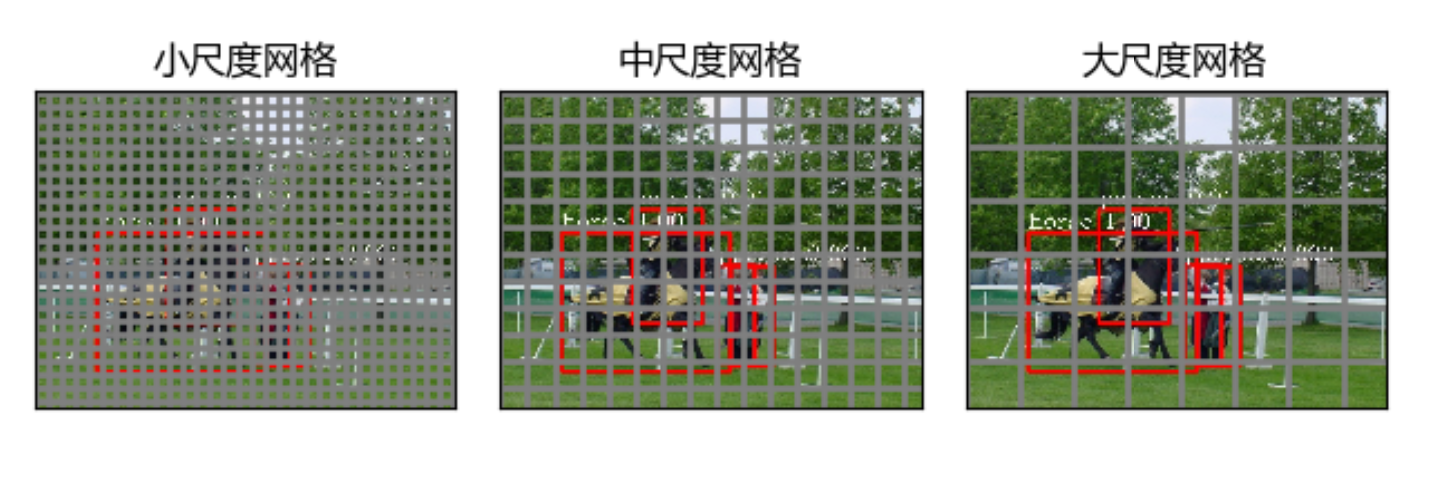


图3-3 三个尺度特征图感受野如灰色格点所示

* 1. 算法细节
     1. 先验框提取

Yolov3对于输出的三个尺度特征图分配了9个尺度的先验框， 每个特征图对应三个先验框， 特征图中每个格点基于这三个先验框的宽度高度对目标框进行预测， 其中先验框的选择通过对训练集标定的边界框进行k-means聚类产生， 其中k-means距离的度量使用了交并比(Intersection over Union)来衡量边界框之间的相似度， IOU越高， 边界框之间的度量距离越近. 由于不涉及边界框的实际位置， 因此IOU计算时只需要沿边界框四个顶点其中之一对齐即可， 计算公式如下:

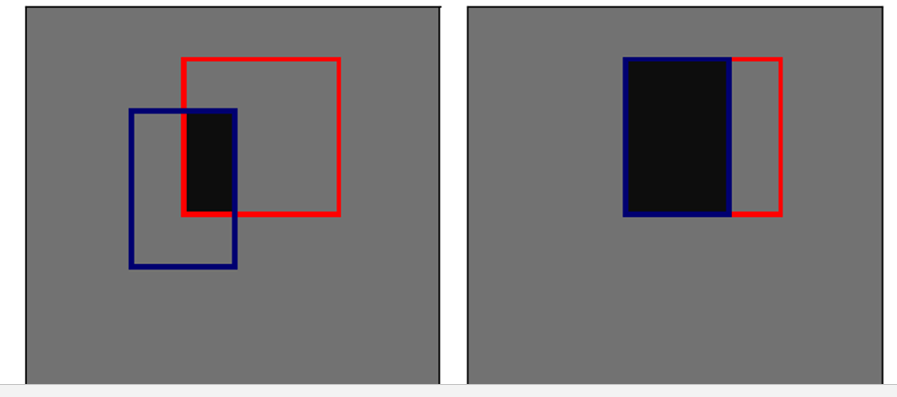


图3-4 左图黑色部分面积为IOU值， 右图黑色部分面积为对齐后的IOU

* + 1. 参数解码

对于Yolov3网络的三个预测特征图，每个特征图单点通道预测目标对象的边界框参数与类别概率， 其中的边界框参数并不是直接预测目标对象的边界框信息。对于位置参数， 预测中心点所在网格的相对位置， 其值在0到1之间，为了将测值限定在0到1之间， 使用了sigmoid函数来进行二值化， 之后再加上网格左上角的偏移来得出相对于原图的位置;对于边界框的宽度和高度， 使用基于先验框高度与宽度参数的相对比例来计算，如图3-5所示。其中为预测值， 为网格左上角偏移，对应图像中为其宽度与高度的， 为先验框的宽度与高度， 之所以需要乘的指数是因为在网络训练时， 对于目标对象标签的宽度和高度参数首先除以对应格点中心相应先验框的宽度高度来得到相对于先验框的尺度缩放， 之后对其值取对数来平滑输输入来根据编码后的参数进行损失的计算， 预测时， 再采用用如下图所示的解码来得到最后的预测输出。

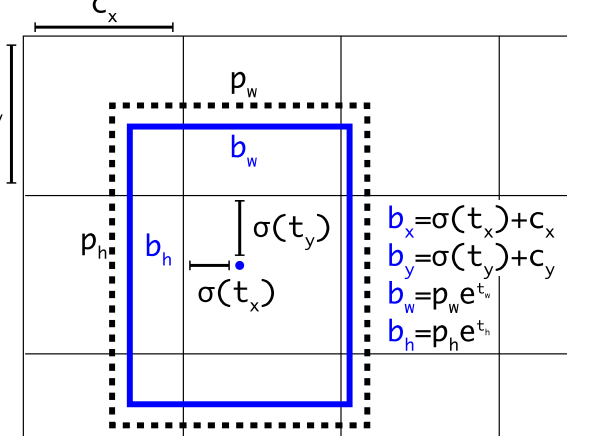


图3-5 预测解码边界框参数[2]

* + 1. 正负样本分配:

对于三个尺度特征图中每一个格点， 如果先验框与训练集标签中标定框的交并比大于一定的阈值，一般取0.3，标定为正样本， 在以上规则下基本能够产生足够多的正样本，但是如果它们的交并比小于等于该阈值，那么只能把交并比最大的那个先验框标记为正样本，这样便能保证每个训练集标定框都至少匹配一个先验框。

按照上述原则，一个标定框会同时与多个先验框匹配. 增加了正样本的数目， 一定程度上避免了正负样本不均衡的问题，分配效果如图3-5所示.按照上面两种规则标记出正样本后，剩下的都是负样本了。这些负样本是不会参与到边界框损失和分类损失的计算中去，而只会参与到用于判断是否是背景的置信度损失的计算。

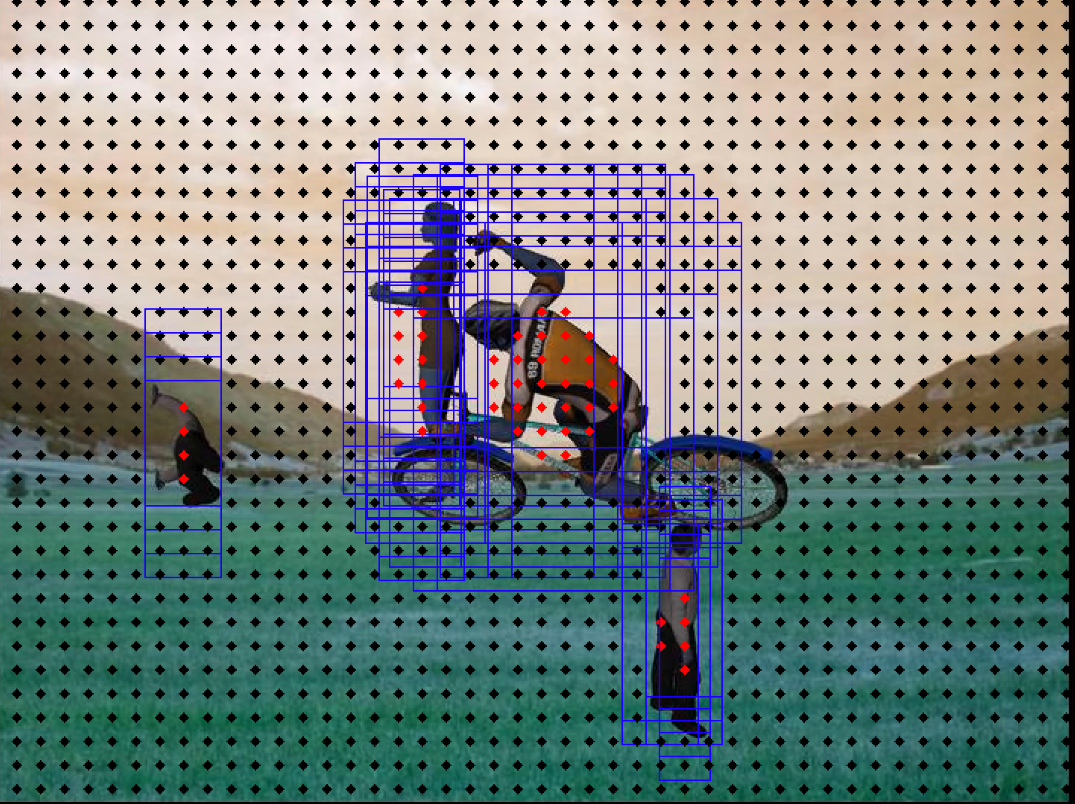


图3-6 正负样本分配可视化，蓝色矩形框视为包含物体的正样本

* + 1. 损失函数:

在 YOLOv3 中，作者将目标检测任务看作目标区域位置预测和类别预测的回归问题， 因此它的损失函数也与其他基于二阶段的检测算法不同。对于损失函数， Redmon J 在论文中并 没有进行详细的讲解。但通过查看 Darknet 源码，可以得出YOLOv3 的损失函数如下:

(1) 信度损失，由于特征网格中存在背景与检测对象，置信度用于判断图像网格中存在检测对象的概率. 置信度损失分为两部分，一部分是背景损失， 另一部分为存在目标对象的前景损失， 对于背景损失，先计算yolov3前向传播得到的特征图中回归的的边界框参数与含有目标对象的正样本边界框的交并比， 将那些交并比小于0.5的格点算作前景. 对于前景损失， 直接用正样本中边界框对应的特征图中格点计算， 其计算使用交叉熵损失:

1. 边界框损失分为位置损失与宽度高度损失两部分. 其中加入了一个2-wh的系数用于增加小物体的损失对整个边界框损失的贡献， 从而改进小物体检测的精度， 其中位置损失使用了交叉熵函数， 宽度高度损失使用了平方损失函数。
2. 分类损失

分类损失先使用sigmoid函数将特征图输出的类别数值二值化，再使用交叉熵损失函数，并且训练时只对正样本计算分类损失。

综上， 网络的总损失为这三个损失之和。

1. 系统实现

检测系统基于TensorFlow机器学习框架2.3版本，使用Python语言实现。整个检测系统主要由数据集加载，模型，训练，评估，检测五个模块构成。模块之间的相互关系如图4-1所示。以下各节主要从模块功能出发，详细介绍各个模块的实现细节。

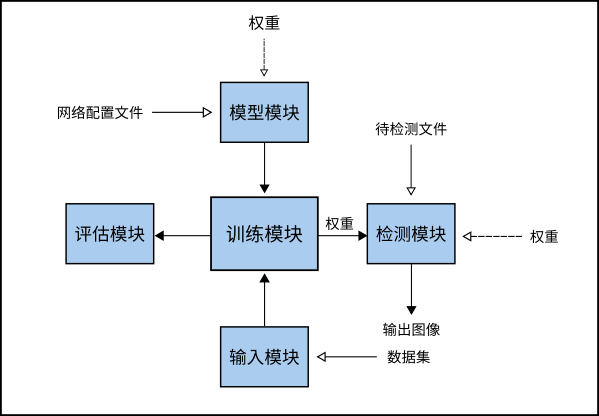
­—­

图4-1 模块关系图

* 1. 数据集加载模块

数据集加载模块的功能是对输入图像进行读取并对其应用一系列图像增强变换操作后送入训练模块。由于训练模块与其串联组成，数据加载的速度会极大影响网络训练的时间，因此构建一个高效的数据输入流水线就显得十分必要。

* + 1. 数据读取

对于数据读取操作，可使用OpenCV库中的imread函数批量读取数据集中单个图像文件实现，不过由于对大量小文件的读取会增加I/O的开销，因此论文中使用了将整个数据集的所有图像文件先转为TFRecord文件格式再进行加载的方式。TFRecord是TensorFlow中用于序列化结构化数据的一种二进制文件格式，压缩效果较好并且可与TensorFlow中数据读取API无缝衔接。经过转换后，整个数据集被分为多个300-400MB的编码后的TFRecord文件，加载时根据设置的缓冲区大小来分批加载数据。如果逐条解码这些编码后的图像再进行图像增强，这种串行操作会很低效，一般使用向量化操作的代码对一整批的数据进行处理再传入网络。对于解码操作而言，一个批次的每个图像之间的解码操作都相同，并且TensorFlow内含对批量图像解码的API，但是对于图像增强操作，每个批次中的图像所应用增强的参数，例如是否水平旋转，饱和度参数，等等都不相同，而进行纯向量化操作地实现很难，出于这样的原因，使用了TensorFlow中Dataset类的map成员方法实现，具体来说是对一整批图像中的每个图像进行解码，归一化，裁剪到固定尺寸，并应用多种图像增强操作，由于map方法底层调用了C++的多线程实现，因此效率也不差。此外，加载时还使用了预取，随机交织读取多个TFRecord文件来加速加载过程。

* + 1. 图像增强

接下来介绍对图像的后处理操作。首先是裁剪图像到固定尺寸。其原因是由于每一批次的图像数据具有相同的尺寸（高度，宽度， 通道数）便于GPU并行计算 ，并且TensorFlow绝大数操作仅支持维度固定的Tensor对象。裁剪的方法使用了保持宽高比裁剪再将缺失的部分使用固定数值填充，如图4-2所­示。



图4-2

左上角为原始图像，左下角为直接缩放后的图像，右下角为保持宽高比缩放的图像

这个方法的好处就是可以最大程度上保持物体的形状不变性，从而避免网络学习到不正确的数据特征。为提高模型的泛化能力并减少过拟合，使用了多种图像增强操作。主要是根据生成的随机数等概率（0.5）地应用剪切，仿射变换，水平翻转，增加饱和度等，具体效果如图4-3所示。图像增强主要有两类，一类是只涉及图像色彩的颜色变换，例如将颜色空间由RGB转为HSV（Hue， Saturation， Value）再进行调整饱和度，色调，明度，这种情况下不需要改变标签的原始边界框坐标。另一类是涉及坐标空间的变换，例如水平翻转，仿射变换，这类操作还需要对标签边界框坐标作相应变换。



4-3 图像增强策略

* 1. 模型模块

模型模块的功能是检测网络的构建，前向传播，损失计算，以及权重加载。模型模块通过Detector类实现。其中，考虑到检测器的运行速度，检测模型后端使用了轻量的YOLOv3 Tiny架构。

* + 1. 网络构建

网络构建通过将Detector的构造函数中的网络配置文件参数传入build\_model函数实现。起初，在编码时面临两种选择，一种是将网络架构硬编码实现，即直接根据网络结构图写出相应的前向传播代码，另一种是写一个解析器直接读取网络配置文件并根据读能够用取的参数构建网络。前一种的弊端是特定于某一种网络架构，不够通用，后一种支持多种网络架构，更具有通用性。在设计初期，使用了前一种方式构建网络，之后通过build\_model函数实现。网络配置文件中部分信息如图4-4所示。以[\*]为界，将整个配置文件分为若干段，逐行读取文件，将这些段读入Python中的字典数据结构中并插入一个列表中（忽略空行和注释）。之后将构造的列表传入网络构造函数中。网络构造函数逐项读取这个列表并根据相应的参数信息实例化相应的网络层（卷积层，批量归一化层），在实例化的同时对一个预定义的符号输入张量进行一次前向传播来构建网络，这也是TensorFlow两种最常见的构建网络的方法之一，即利用函数式API，另一种是通过继承TensorFlow中Keras.Module类实现。在build\_model函数内部，还进行了损失函数的构建，因为YOLOv3的损失函数比较复杂，因此将其包装成了YOLOLoss类实现。

* + 1. 损失计算

YOLOLoss类用于计算检测网络的损失，由于YOLO Tiny网络有两个不同尺寸的输出层，因此实际上实例化了两个实例来分别计算两个输出层的损失。在计算损失时，首先计算6个先验框与所有标注边界框的IOU，选出与标签具有最大IOU的边界框以确定哪些标注对象的大小最接近该输出层的感受野。其次对这些标注对象计算其在输出层的格点位置并将标注对象坐标转化为相对于输出格点的相对偏移。之后，通过计算这些标注对象与该输出层所有预测的IOU来确定忽略损失的格点。

最后，对所有挑选出的标注对象所在格点计算置信度前景损失，类别损失，以及边界框损失，对所有不属于此格点同时不属于忽略格点的部分计算置信度背景损失，将以上两者相加得到对于该输出层的一次前向传播所得损失。Detector类保存了对这两个损失类的引用，并在计算整个网络的损失时将两层损失相加即得到整个网络的损失。

* + 1. 权重加载

模型模块还支持对部分或全部权重文件的加载来进行模型微调以及进行迁移学习。从0开始训练模型很难，一方面是由于本身数据集不大，训练网络时所学到的类别特征有限，另一方面是使用预训练模型可以增加模型的泛化能力并且预训练模型包含一些与分类任务相关的映射关系， 使得训练从一个好的起点开始。权重的加载支持TensorFlow训练时保存的CheckPoint格式或者YOLOv3作者给出的二进制文件格式。对于前者，直接使用TensorFlow的CheckPoint.restore方法实现，对于后者，除最后两层输出层外，模型架构和YOLO Tiny相同，因此直接按照构建后模型中权重的尺寸按字节读取二进制文件并转为float32格式后对权重赋值。流程如图4-4所示。

* 1. 训练模块

训练模块的功能是对网络训练过程的控制，主要包括对网络前向传播与反向传播的控制，训练期间权重的保存，训练过程监测，以及日志记录。具体功能是通过运行时构建的单一Trainer类实例来实现的。Trainer类的构造参数包括对Detector实例的引用，训练参数字典和数据集参数字典。训练模块相当于检测系统的控制单元。

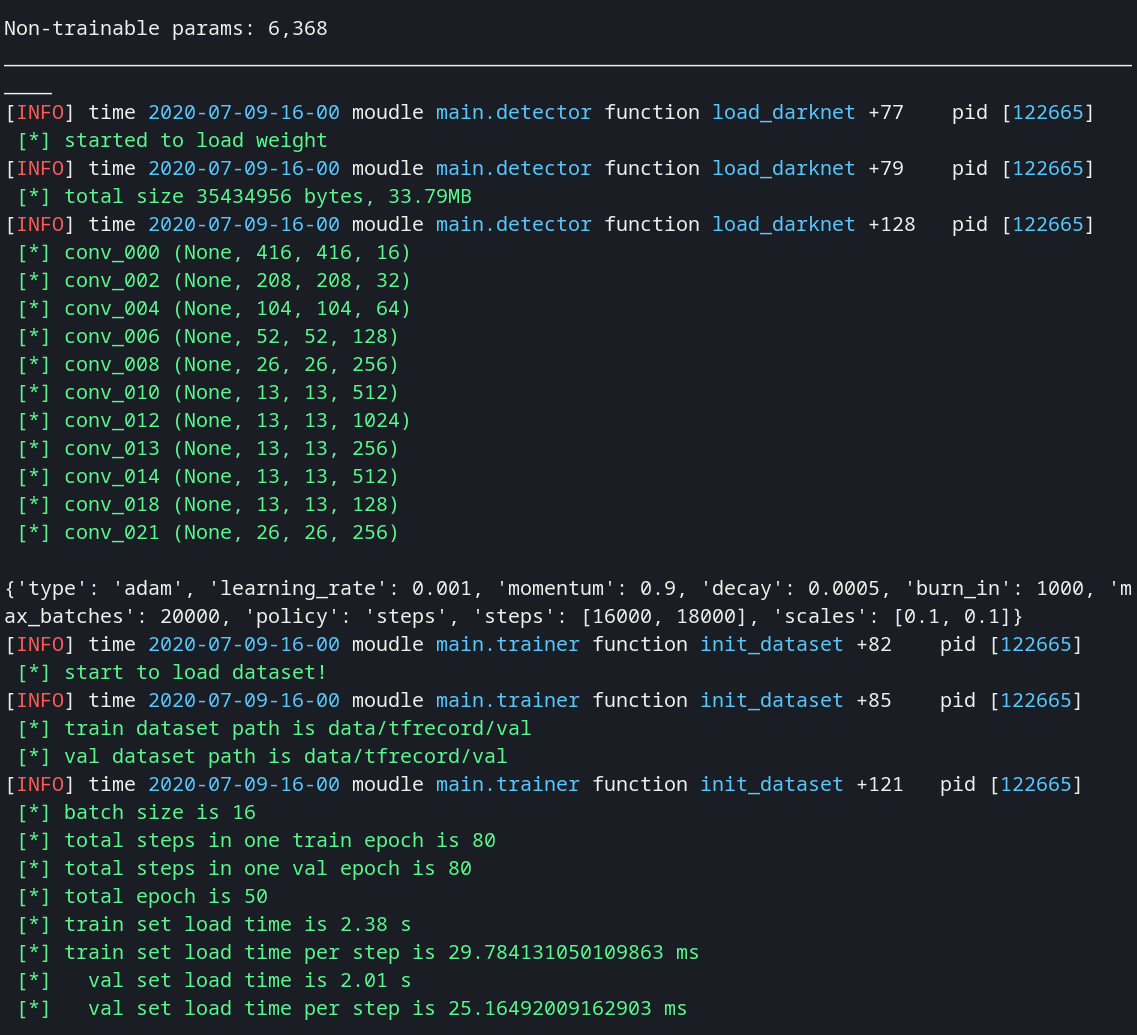
* + 1. 训练控制

训练模块通过保存对Detector实例以及Dataset实例的引用来实现对网络的前向传播与反向传播过程的控制。在Trainer类实例化时，通过其构造参数中的数据集参数字典来初始化Dataset实例（数据集模块），由于数据集分为训练集与验证集，所以实际上是对两个实例的引用。此外，通过训练参数字典中的参数初始化训练迭代次数，初始学习率，训练日志保存路径，中间权重保存路径等一系列成员变量。在训练时调用Dataset实例生成可迭代对象来遍历数据集，将批量训练数据传入Detector实例进行前向传播，并通过Detector返回的损失应用梯度下降来更新网络权重。训练模块还支持冻结部分层的网络而只对输出层进行训练来进行迁移学习，以及通过应用余弦调度和warm up策略来动态调节学习率使网络训练更稳定。

* + 1. 训练监测

训练监测是对训练过程中的网络损失，评估指标，以及各个模块的输出日志进行监测。每经过一次对整个数据集的前向传播，train方法会调用评估模块对模型进行评估，并将网络损失及评估数值（AP, mAP）写入TensorBoard日志文件中以监测训练过程。并且，根据上一步计算出的mAP评估指标保存最佳的训练权重。出于在训练过程中调试的需要，在训练过程中还实现了对每个模块的输出信息定位到代码行数，输出时间，并根据输出消息的类型将关键信息写入到训练日志目录中。

训练过程中部分输出如图4-所示。



* 1. 评估模块

评估模块是对模型检测效果的度量。其度量标准使用了IOU大于0.5条件下的平均准确度（mAP@0.5），评估标准的介绍见5.3。

评估模块通过实例化Evaluator类并调用eval方法实现，并且支持训练时评估以及离线评估。这两者之间的区别是，训练时评估是累积更新一个批次的预测信息与标注信息，离线评估是直接对网络的所有预测与所有标签进行评估。因为评估时需要的信息比较多且具有结构化的特点，所以使用了pandas数据处理库中的DataFrame类来保存预测与标注信息。训练期间评估时，网络首先经过一次前向传播，之后应用NMS算法过滤重叠度高的低置信度预测，然后调用Evaluator类的update\_pred和update\_gt方法来累计统计一个批的预测和标注信息，当遍历完整个数据集时，Evaluator实例就保存了整个数据集的预测及标注信息。最后，通过eval方法来对整个数据集进行评估。对于离线评估，评估模块接受两个分别含有预测与标信息的CSV文件参数，再通过这两个文件构建相应的DataFrame后传入eval方法。在eval方法中，首先将预测与标注两个DataFrame根据标注的图像编号与物体类别字段合并成一个DataFrame, 这样就过滤掉了那些预测类别与标注类别不一致的假正例（FP），之后再根据图像编号分组对所有组进行组内计算IOU并作为新属性插入到这个DataFrame中，然后以0.5为IOU阈值过滤掉IOU不满足条件的假负例，最后以置信度从高到低对所有组进行重排序，找出与标注图像中物体唯一匹配的真正例（TP）。对每个预测来说，只能是TP与FP二者之一，再加上标注信息即可求得假负例(FP)的数目。根据以上信息计算每个类别的P-R曲线下面积再求平均后即可求得mAP。

* 1. 检测模块

检测模块负责对图像或视频文件应用检测模型来检测特定类别的对象并进行标注。在训练网络时保存的权重是CheckPoint文件格式，这种格式仅仅保存了网络的权重和训练参数，而并没有保存网络结构信息，在导入时需要事先构建模型再进行加载。因此在实际检测前先将模型转化为包含网络结构与权重的文件。导出过程通过构建模型，加载权重后调用tensorflow.save\_model.save方法实现。导出模型的同时，其预测函数也被序列化成二进制格式。检测模块通过命令行参数传入模型权重路径，检测文件名或目录，NMS参数来调用相应的处理函数。对于图像，先将图像保持宽高比缩放成网络的输入尺寸后调用反序列化的预测函数获得网络的原始预测，再通过NMS算法过滤掉冗余的预测，最后在图像上应用OpenCV库的rectangle函数绘制边界框和类别信息后保存。对于视频，将每一帧作为一张输入图像处理。检测效果如图4-所示。



1. 评估与测试
   1. 实验环境

CPU为AMD Ryzen 5， GPU为NVIDIA GTX 1650(4 G 显存) ， 内存16GB， 操作系统Arch Linux， TensorFlow版本为2.3。

* 1. 数据集

由于基于深度学习的对象检测器需要大量的数据来训练，因此数据集的选择至关重要。对于检测任务，常见的数据集有PASCAL VOC， MS COCO， Objects365等。PASCAL VOC数据集[2007， 2012]共含有包含20个类别的16，551张图像，40058个标注对象。COCO数据集[2017]含有80个物体类别，总计123，287张图像，886，284个标注对象。与PASCAL数据集相比，COCO数据集图像场景信息更丰富，并且数据集总对象数目以及单张图像平均对象数目较多，因此，论文使用了COCO数据集来进行对象检测器的训练与测试。然而，考虑到GPU设备的限制与训练成本，取了COCO数据集的一个仅包含10个类别对象的子集进行实验，并按照6:2:2比例将其划分为训练集，验证集，测试集。划分后数据集中类别对象数目的对比如图1。

图表, 条形图

描述已自动生成

图表, 饼图

描述已自动生成

​

由以上统计信息可知采样后的数据集在数量分布上保持了COCO数据集的分布规律，并且具有类别分布不平衡和物体大小分布不平衡的特点。

一方面，数据集中类别为人的对象占总数的65%， 数目占比前三的类别其数目之和为总数的80%，类别分布极不平衡。另一方面，不同类别的对象之间相对大小差距较大，例如，交通信号灯的大小远远小于火车或飞机的大小。

对于前者，在不引入外部数据集的前提下，训练时使用了类别加权的方式调整不同类别的对象对损失函数的贡献，并且使用多种图像增强方法（水平旋转，随机剪切，饱和度调整，亮度调整）增强网络的鲁棒性；对于后者，在不改变网络结构的条件下，训练时对边界框损失使用CIOU损失替换了平方损失，改善了小物体的检测准确率。

* 1. 评估标准

论文使用IOU为0.5的条件下平均准确率(mean Average Precision, mAP)来对预测结果进行评估。mAP为每个类别平均准确率(Average Precision， AP)的平均值。AP定义为P-R（Precision-Recall）曲线下面积。Recall， Precision的定义见2.3节。

（1）真正例(True Positive， TP): 预测类别为c且真实类别为c。

（2）假正例(False Positive， FP): 预测类别为c且真实类别不为c。

（3）假负例(False Negative， FN): 定义为预测类别不为c但真实类别为c。

对于对象检测任务而言，如果一个预测样本满足以下四个条件则为真正例：

1．预测样本与标注样本真实类别一致。

2．预测样本与标注样本真实边界框IOU大于一定的阈值(通常为0.5) 。

3. 预测样本的置信度大于一定阈值。

4．与之匹配的标注样本没有被检测过。

图表, 瀑布图

描述已自动生成

对于所有预测，通过应用不同的置信度阈值，可以计算出不同的TP， FP， FN数目进而得到不同的召回率与精度数值。以为横轴，为纵轴建立平面直角坐标系，在相同置信度阈值下的即为坐标系上一点，将多个点首尾相连即为P-R曲线。由于这样点的有很多个（整个数据集的预测），因此其曲线下面积，即AP， 可以近似表示为：

* 1. 评估结果与可视化

1. 总结与展望

在深度学习与计算机视觉迅速发展的今天，对象检测已经被广泛应用在无人驾驶，人脸识别，医学图像处理等领域。较以往来说，检测精度与速度也有了极大的提升。但是，由于过于依赖GPU， 上在具体应用的过程中，视应用场景的不同，往往是对两者的折衷。以下基于对检测算法的分析与实现进行总结，并结合已经完成的工作做出对未来进行展望。

* 1. 总结

本文深入分析了Yolov3对象检测算法，从网络结构与算法细节入手，研究了Yolov3的DarkNet53后端网络以及前端网络的组成，并对算法中先验框提取，正负样本分配，特征融合，参数编码，损失函数进行了详细的分析，之后基于对算法的分析与理解，以预处理，模型，训练，预测，检测五个模块实现了对象检测算法，并在Pascal VOC数据集上进行了评估，取得了88%的mAP(@0.5IOU)，检测速度为35fps。

在实现的过程中，一开始由于数据预处理模块需要先对数据做图像增强等操作，然后将数据送至训练模块进行训练，在预处理时耗费的CPU读取并对数据预处理的时间是训练耗时的瓶颈，因为此时GPU处于空闲状态，GPU利用率在到达高峰时会迅速降至0，为了加快训练时间，采用了将部分数据缓存在内存中减少读取时间并且采用了并行化预处理操作来加快训练，从而缓解了训练一个epoch时间过长的问题。此外，先验框的选取十分重要，直接影响了最后检测结果的精度。一开始选取先验框时，只是对未处理的训练样本标注框直接进行k聚类选择先验框，但在训练时边界框损失减少很慢，其中主要是边界框大小的损失占了一大部分，后来发现在提取先验框时没有对原图像宽高进行放缩以匹配训练过程中样本的大小。对于检测速度的问题，一开始的检测速度很不理想，之后通过将关键计算使用静态图从而加速了计算过程进而加速了检测速度。

* 1. 展望

未来的工作主要有三个方面。一方面也可以通过更换YOLOv3的后端网络结构，比如使用 MobileNet或者YOLOv3 Tiny的更轻量的网络架构来加速检测速度。另一方面在训练模块中可以实现多尺度图片输入训练的功能来增加对待检测图片的尺度相差过大的鲁棒性。最后一方面可以将对象检测算法通过TensorLite部署到移动端或者嵌入式设备中来模拟更加现实的应用场景。

参考文献

[1]Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv: 1804.02767

[2]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.

[3]Zou Z, Shi Z, Guo Y, et al. Object Detection in 20 Years: A Survey[J]. arXiv: Computer Vision and Pattern Recognition, 2019.

[4]Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]. computer vision and pattern recognition, 2014: 580-587.

[5]Girshick R. Fast R-CNN[C]. international conference on computer vision, 2015: 1440-1448.

[6] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[C]. european conference on computer vision, 2016: 21-37.

[7]Rojas, R. Neural Networks - A Systematic Introduction). Springer-Verlag, Berlin, New-York, 1996.

[8]Long, L. N., and Gupta, A. Scalable massively parallel artificial neural networks. Journal of Aerospace Computing, Information, and Communication 5, 1 (2008), 3–15.

[9]Hornik, K. Approximation capabilities of multilayer feedforward networks. Neural networks 4, 2 (1991), 251–257.

[10] Goodfellow, I., Bengio, Y., and Courville, A. Deep Learning. MIT Press, 2018. http://www.deeplearningbook.org.

[11] Redmon J, Divvala S K, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. computer vision and pattern recognition, 2016.

[12] Redmon J, Divvala S K, Girshick R, et al. YOLO9000: Better, Faster, Stronger computer vision and pattern recognition, 2016.

[13]李名波.基于机器学习的目标检测算法综述[J].计算机产品与流通,2019(06):154-155.

[14]李云鹏,侯凌燕,王超.基于YOLOv3的自动驾驶中运动目标检测[J].计算机工程与设计,2019,40(04):1139-1144.

致 谢

时光荏苒，四年大学时光如白驹过隙。回顾四年来在学校的日子，有过欢笑，也有过挫折后的消沉与低迷。四年来，不仅学到了专业知识，同时结交了很多朋友，可能毕业一别，很多人都很难再见到了。现在，即将离开校园，更多的是怀念与不舍，怀念和舍友生活，小打小闹的点点滴滴，怀念一起上课，一起怕迟到急急忙忙吃早餐的那些日子。

首先感谢四年来辅导员对我们在生活和学习上的照顾；其次，感谢舍友和同学的陪伴与鼓励，四年来，165班这个大家庭给了自己一种家一样的感觉，班长邵津一直都把每个人遇到的问题看成自己的事处理，尽自己所能解决问题。舍友们也相互帮助与理解，还要感谢孙昊在自己最低沉最难过的日子里的关心和安慰。

最后，还要感谢指导老师，今年因为疫情原因没有返校，一切的进度安排与资料提交都在线上，虽然最后由于自己的原因没有及时提交检测，答辩被延期了半年，但是还要感谢最后指导老师与自己的交流使自己明白了自己的问题并且对之后要改进的方向更清晰了一些，以后自己在学习上要更主动一些，多与老师同学交流而不是自己一个人单干，学习如此，做人也是如此。

青山不改，绿水长流。离开校园，走向社会这个大染缸，希望自己保持本心，自省，自立，挥手道别，来日再见。