

# 专业学位论文

## 面向边缘计算的目标检测算法研究与应用

Research and Application of Target Detection Algorithm for Edge Computing

作者姓名: 张钊

工程领域: 控制工程

学号: 31909107

指导教师: 杨建华

完成日期: 2021.6.15

大连理工大学

Dalian University of Technology

---

## 大连理工大学学位论文独创性声明

作者郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用内容和致谢的地方外，本论文不包含其他个人或集体已经发表的研究成果，也不包含其他已申请学位或其他用途使用过的成果。与我一同工作的同志对本研究所做的贡献均已在论文中做了明确的说明并表示了谢意。

若有不实之处，本人愿意承担相关法律责任。

学位论文题目：面向边缘计算的目标检测算法研究与应用

作者签名：张钊 日期：2021 年 6 月 15 日

## 摘 要

现代安全防护领域多采用基于云计算的人工视频监控方法，在监测低概率出现的异常目标时，监控系统易出现由于视频信息冗余导致网络资源浪费、数据传输存在延迟以及人为失误导致检测质量下降等问题。边缘计算通过网络边缘端对前端采集数据进行实时分析运算，可以有效减轻云端负载压力，提高系统响应实时性。同时随着深度学习技术尤其是目标检测算法迅猛发展，使得视频监控更加高效智能，有效缓解人工视频监控方法存在的缺陷。因此，本论文提出一种面向边缘计算的视频监控系统架构方案，利用基于深度卷积神经网络的目标检测算法实现异常目标高效捕捉，并通过煤层气井站监控数据验证了方案可行性。具体研究内容如下：

首先，针对基于云计算的人工视频监控方案存在的性能问题，设计面向边缘计算的视频监控系统架构方案。该方案采用 NVIDIA TX2 作为边缘设备端，执行数据的采集以及检测算法前向推理；PC 电脑作为边缘服务器，执行视频数据和检测算法参数文件的存储；智星云服务器为云端，执行算法参数迭代更新并定时推送参数文件到边缘服务器端实现协同工作。通过在煤层气井站监控数据检验，有效验证设计方案具备可行性。

其次，针对当前目标检测算法的特征提取网络模型参数较大、运行内存占用高等问题，提出一种面向边缘计算的轻量化深度卷积网络 MMGNet。该网络通过深度可分离卷积模式，有效降低模型计算消耗与存储要求，并结合 GhostNet 进一步实现模型轻量化。提出的网络在结构上保持“先扩张，后压缩”形式，并引入通道注意力机制提高模型精度。此外，提出的网络还加入多尺度卷积核进行特征信息提取，模拟人不同视角看待物体特性，进一步提高了检测精度，在 CIFAR100 数据集上取得 71.52% 的 Top-1 准确率。

最后，针对视频监控中异常目标监测方法，采用基于 MMGNet 的目标检测算法实现异常目标快速准确捕捉。本论文首先以 Faster-RCNN、SSD 以及 YOLOV4-tiny 作为基础算法框架，将特征提取网络替换为本论文设计的 MMGNet。然后通过 Pascal VOC 2007 与 2012 的部分目标检测数据集进行对比实验，验证了设计网络能够有效降低目标检测算法参数量，检测精度与运行速度均满足实际需要。经横向对比各类指标，选择基于 MMGNet 的 YOLOV4-tiny 算法对煤层气井站人工标注数据集测试，在 TX2 平台运行取得 92.15% 的分类平均精度均值以及 0.102s 的单张检测时长，测试结果表明提出的方法是可行有效的。

**关键词：**边缘计算；NVIDIA TX2；深度卷积神经网络；MMGNet；目标检测算法

## Research and Application of Target Detection Algorithm for Edge Computing

### Abstract

In the field of modern security protection, manual video monitoring method based on cloud computing is often used. When monitoring abnormal targets with low probability, the monitoring system is prone to the problems such as the waste of network resources due to redundancy of video information, the delay of data transmission and the decline of detection quality caused by human error. Edge computing performs real-time analysis and operation on the collected front-end data at the edge of the network, reduces the cloud load pressure effectively and improves the real-time response of the system. At the same time, with the rapid development of deep learning technology, especially target detection algorithm, video surveillance is more efficient and intelligent, and the defects existing in artificial video surveillance methods are effectively alleviated. Therefore, this paper proposes a scheme for edge computation-oriented video surveillance system, which utilizes the target detection algorithm based on deep convolutional neural network to realize the efficient capture of abnormal targets, and verifies the feasibility of the scheme through the monitoring data of coalbed methane well station. Specific research contents are as follows:

Firstly, in view of the performance problems existing in the cloud-based manual video surveillance scheme, a video surveillance system architecture scheme oriented to edge computing is designed. The scheme uses NVIDIA TX2 as the edge device to perform data collection and forward reasoning of detection algorithm. PC is the edge server, which is executes the video data and detects the algorithm parameter file cache. The intelligent cloud server performs algorithm parameter iterative update and regularly pushes parameter files to the edge server to achieve collaborative work. The feasibility of the design scheme is verified by monitoring data in coalbed methane well station.

Secondly, a lightweight deep convolutional network named MMGNet for edge computing is proposed to solve the problem that the feature extraction network of the current target detection algorithm has large parameters and high operating memory. The network can effectively reduce the calculation consumption and storage requirements of the model by the depthwise separable convolution, and further realizes the model light weight with GhostNet. The proposed network maintains the form of "expansion first, then compression" in structure, and introduces channel attention mechanism to improve the model accuracy. In addition, the proposed network also adds multi-scale convolution kernel to extract feature information,

which simulates people's view of object characteristics from different perspectives, and further improves the detection accuracy. The TOP-1 accuracy rate of 71.52% is achieved on CIFAR100 dataset.

Finally, aiming at the abnormal targets monitoring method in video surveillance, the target detection algorithm based on MMGNet is adopted to realize the rapid and accurate capture of abnormal targets. In this paper, Faster-RCNN, SSD and YOLOV4-tiny are used as the basic algorithm framework, and the feature extraction network is replaced by MMGNet designed in this paper. Then, through comparative experiments on part of Pascal VOC 2007 and 2012 target detection data sets, it is verified that the designed network can effectively reduce the parameters of target detection algorithm, and the detection accuracy and running speed can meet the actual needs. After horizontal comparison of various indicators, YOLOV4-tiny algorithm based on MMGNet is selected to test the manually labeled data set of coalbed methane well station. The average classification accuracy is 92.15% and the detection time of single sheet is 0.102s when running on the TX2 platform. The test results show that the proposed method is feasible and effective.

**Key Words:** Edge computing; NVIDIA TX2; Deep convolutional neural network; MMGNet; Target detection algorithms

## 目 录

摘 要.....	I
Abstract .....	II
1 绪论.....	1
1.1 论文研究背景和意义 .....	1
1.2 国内外研究现状.....	1
1.2.1 边缘计算.....	2
1.2.2 深度卷积神经网络 .....	2
1.2.3 目标检测算法 .....	5
1.3 论文研究内容与章节安排.....	6
1.3.1 研究内容.....	6
1.3.2 章节安排.....	7
2 基于边缘计算的视频监控系统设计.....	9
2.1 边缘计算概念及特点 .....	9
2.2 边缘计算应用场景 .....	10
2.3 嵌入式边缘计算设备 .....	10
2.4 面向边缘计算的视频监控系统架构方案 .....	11
2.5 本章小结 .....	13
3 深度卷积神经网络算法分析 .....	14
3.1 网络基本结构.....	14
3.2 MobileNet 系列轻量化神经网络.....	20
3.2.1 MobileNet V1.....	20
3.2.2 MobileNet V2.....	22
3.2.3 MobileNet V3.....	23
3.3 本章小结 .....	24
4 面向边缘计算的深度卷积网络设计.....	25
4.1 MMGNet 深度卷积网络.....	25
4.2 实验结果与分析 .....	28
4.2.1 实验环境.....	28
4.2.2 数据集.....	29
4.2.3 评价指标.....	29
4.2.4 实验结果与分析 .....	30

4.3	本章小结 .....	33
5	基于 MMGNet 的目标检测算法设计 .....	34
5.1	目标检测算法.....	34
5.1.1	Faster-RCNN 算法 .....	34
5.1.2	SSD 算法.....	37
5.1.3	YOLOV4-tiny 算法 .....	39
5.2	实验结果与分析 .....	42
5.2.1	数据集.....	42
5.2.2	评价指标.....	43
5.2.3	实验内容与结果分析.....	43
5.3	本章小结 .....	45
6	煤层气井站异常目标检测 .....	46
6.1	应用场景介绍.....	46
6.2	数据集 .....	47
6.3	评价指标 .....	48
6.4	实验结果分析.....	48
6.5	本章小结 .....	50
结 论	.....	51
参 考 文 献	.....	53
致 谢	.....	58
大连理工大学学位论文版权使用授权书	.....	59

# 1 绪论

## 1.1 论文研究背景和意义

在现代安全防护领域，行业普遍采用人工视频监控模式对特定区域进行异常信号检测，再结合云计算范式<sup>[1]</sup>实现全方位、系统化的安全防护。该模式较依赖于监控人员的主观判断，若监控人员职业技术能力较低，则容易出现异常信号漏检、误检等情况发生。在如今万物互联时代，视频监控节点呈指数增加，采集传输的视频流数据激增，对传统云计算范式的数据存储、传输以及网络带宽资源势必造成严重负担。针对视频监控这类低延时要求的应用场景，传统云计算范式将无法满足现有的生产生活需要。因此，研究人员提出边缘计算<sup>[2]</sup>，该计算方式通过在网络边缘端执行计算，实现在数据源头进行数据采集、分析处理工作，既保证系统实时处理要求也极大地优化网络资源利用率。与此同时，芯片、嵌入式设备等技术快速发展，为边缘计算提供了强有力的硬件基础。因此，基于边缘计算架构的视频监控系统具备重要理论研究与实际应用价值<sup>[3]</sup>。

视频监控系统中的首要问题是能否对异常目标进行精准高效捕捉。随着计算机视觉技术<sup>[4,5]</sup>迅速发展，将目标检测算法运用到异常信号捕捉可有效缓解人为因素干扰，提高视频监控质量。深度卷积神经网络凭借高效的特征提取能力，成为解决目标检测相关问题的重要方法之一。但由于深度卷积神经网络参数冗余导致模型文件普遍较大，对硬件设备算力以及存储要求较高，使得基于卷积网络的目标检测算法在边缘嵌入式平台上运行存在巨大挑战。因此，如何设计一种高效且轻量化的深度卷积神经网络用于特征信息提取，是现如今研究的热点与难点。

本文选择煤层气井站作为研究应用场景。煤层气是我国重要的可燃烧清洁能源，而作为开采煤层气的作业平台，煤层气井站多分布于偏远郊区且无人值守。因此，为保证煤层气安全开采，需要对煤层气井站实现视频监控。传统监控方法多采用基于云计算的人工视频监控方式，需要安排大量人员对视频画面监测，较依赖监控人员主观判断，容易因人为疏忽导致异常目标错检漏检。同时煤层气井站监控系统涉及的摄像头数以千计，产生的数据流量较大，对网络传输负载以及云中心存储计算会产生巨额消耗，容易导致系统延时。因此针对煤层气井站这类场景，基于边缘计算架构智能视频监控系统以实现异常目标实时精确捕捉，便有着重要研究价值与广阔的应用前景。

## 1.2 国内外研究现状



### 1.2.1 边缘计算

随着万物互联时代到来,可接入互联网设备产生的流量数据呈指数型增长,云端负载激增以及网络带宽消耗问题亟待解决。因此研究人员开始探索在接近数据源的网络边缘增加计算能力的可行性,其间诞生了很多新型计算模型,具有代表意义的是移动边缘计算<sup>[6]</sup>、雾计算<sup>[7,8]</sup>以及海计算<sup>[9]</sup>,它们均与边缘计算在架构理念上有着相似性。2016年,施威松<sup>[10]</sup>等人将边缘计算定义为在网络边缘实现计算功能的全新计算模型,被计算对象不仅涵盖物联网数据还包括云端数据。边缘被定义为从数据产生源头到云端中心之间任意一个网络资源节点。同时施威松也指出边缘计算运用到智慧城市、视频分析、智能家居等应用场景的可能性以及未来发展面临的机遇与挑战。Satyanarayanan 等人<sup>[11]</sup>为解决边缘计算架构问题提出将云端计算能力以及存储功能下沉到网络边缘节点,有效降低云端负载,提高系统响应实时性。Jie Lin 等人<sup>[12]</sup>探索边缘计算结合物联网技术,有效降低物联网系统响应延时,提升物联网服务质量。Hossaion 等人<sup>[13]</sup>针对智慧城市应用场景,采用大规模物联网智能设备结合边缘计算架构系统,有效降低服务延迟。随着 5G 技术发展<sup>[14]</sup>,移动智能设备通信能力提升,Sun X 等人<sup>[15]</sup>就此提出新型物联网架构的移动边缘计算方法 edgeIoT,实现海量移动边缘数据流地有效处理。蔡锴等人<sup>[16]</sup>通过研究指出了边缘计算在智能制造领域应用的关键技术能力,分析了边缘计算在智能制造领域面临的机遇与挑战,并具体给出边缘计算运营模式以及应用方向。

边缘计算系统低延时性使其在视频监控领域有广阔的应用价值。Wang R 等人<sup>[17]</sup>基于边缘计算以及权限区块链架构新型视频监控系统,该系统通过边缘计算采集视频数据并预处理,分部式文件系统用于数据存储,深度卷积网络用于数据分析,有效降低系统延时,监控质量也有所提升。Chen J 等人<sup>[18]</sup>提出采用深度学习的分布式智能监控系统,结合边缘计算有效减少网络通信损耗,为监控视频分析提供了一种高效解决方案。亓慧等人<sup>[19]</sup>针对仓库视频监控场景,提出基于边缘计算架构的监控系统。在一定可靠性对边缘视频数据处理分析,有效降低系统延时。葛畅等人<sup>[20]</sup>架构一种基于边缘计算的新型视频监控系统,该系统中边缘节点利用帧过滤算法对视频数据进行处理分析,实现对目标有效识别并选出有效视频帧上传,同时设计资源调度算法提高系统效率,经实验有效验证该系统降低数据传输以及处理分析的资源消耗。

### 1.2.2 深度卷积神经网络

作为人工智能领域重要技术分支,深度学习在现如今的生产生活中发挥着重要作用。而卷积神经网络则是解决深度学习领域相关问题的基础工具之一,凭借优异的特征信息提取能力,被广泛运用到了图像分类、自然语言处理以及目标检测等实际应用场景。

1998 年, 由 LeCun 等人<sup>[21]</sup>首次提出的深度卷积神经网络 LeNet-5, 在手写体识别准确率达到 98%。而后深度学习理论蓬勃发展, Krizhevsky 等人<sup>[22]</sup>提出的 AlexNet 网络是卷积神经网络领域的历史突破, 该网络在 ImageNet 比赛夺冠, 自此之后, 各类结构新颖、性能优秀的卷积模型竞相提出。2014 年, Simonyan K 等人<sup>[23]</sup>提出 VGG 深度卷积神经网络, 通过较小卷积核堆叠成更深的网络使得算法性能较 AlexNet 有了大幅度提升; Szegedy C 等人<sup>[24]</sup>提出 GoogLeNet 的在 ILSVRC 分类比赛上战胜 VGG 夺得冠军, 该网络创新提出 Inception 结构块, 通过运用多种尺度卷积核运算减少参数, 增加网络宽度, 并用全局池化代替全连接网络层实现网络参数量降低且提高网络性能; 2015 年, Ioffe S 等人<sup>[25]</sup>提出 Inception V2, 该网络用小卷积核代替 GoogLeNet 中的较大卷积核, 对卷积层输出进行批量归一化处理 (Batch Normalization, BN), 有效提高模型预测精度, 该网络在 2012 年的 ILSVRC 比赛数据上取得 95.18% 的 Top-5 分类准确率; Xia X 等人<sup>[26]</sup>提出的 Inception V3 网络对卷积核做了进一步的细致分解, 它将  $n \times n$  大小的卷积核分解为两个较小的卷积核即  $1 \times n$  和  $n \times 1$ , 使得网络的非线性能力增加, 在减少网络参数量同时也保证较好网络性能。随着深度卷积神经网络结构宽度以及深度的不断增加, 模型参数量也逐渐增高, 模型训练过拟合等问题凸显。He 等人<sup>[27]</sup>提出 ResNet 有效解决由于模型深度增加导致的网络退化问题<sup>[28]</sup>, 该网络通过残差模块使网络传递的前后信息交互, 有效解决深度网络带来的梯度弥散问题, 使得增加网络层数以提高网络精度成为可能。但 He 也指出过深的网络堆叠反而容易造成性能受损。Szegedy C 等人<sup>[29]</sup>融合 ResNet 以及 Inception 优势特点提出 InceptionV4 以及 Inception-ResNetV2, 在 ImageNet 取得 96.92% 的 Top-5 准确率。Jie H<sup>[30]</sup>等人从探索特征通道之间的关系出发提出 SENet, 该网络学习不同特征通道的重要程度, 对有用通道进行增强并抑制作用较小特征, 有效提高网络精度, SENet 也在 2017 年 ImageNet 分类比赛取得冠军; Li X<sup>[31]</sup>等人在 SENet 基础上提出了一种 SKNet 模型, 通过多尺度自适应改变特征提取范围从而获得不同感受野, 网络性能进一步提高; Woo S<sup>[32]</sup>等人则提出一种结合空间与通道注意力机制模块 CBAM, 使得网络模型特征提取能力相较于只关注通道注意力的 SENet 更佳。

深度卷积网络不断发展, 各类型网络层出不穷, 但是这些网络往往伴随着高参数量、高计算量的特点, 导致网络在较低算力与运行内存的嵌入式设备上难以运行。因此构造轻量化的深度卷积神经网络成为近些年国内外学者研究热点。2016 年 Iandola 等人<sup>[33]</sup>提出轻量化卷积网络 SqueezeNet。该网络由数个 Fire Module 堆叠而成, 该模块由 squeeze 与 expand 两部分构成, 前者即  $1 \times 1$  卷积层, 后者是由多个  $1 \times 1$ 、 $3 \times 3$  卷积层并行组成。同时该算法借鉴 GoogLeNet 网络结构, 用平均池化代替全连接层, 实现了将 AlexNet 压缩至小于 0.5MB, 且保持了一定的模型精度。2017 年 Howard 等人<sup>[34]</sup>提出 MobileNet,

通过采用深度可分离卷积模块实现卷积网络轻量化, 在 ImageNet 数据集取得 70.6% 的 Top-1 分类准确率, 预测结果与 VGG16 的近似, 但该算法有效减少 32 倍的参数量以及 27 倍的模型计算量。同时作者设置宽度因子以及分辨率因子两个超参数, 使网络既保持一定分类精度又进一步实现轻量化。MobileNet 网络结构是一个类似 VGG 的直通式结构, 在 ResNet 网络模型的研究中指出重复使用提取的特征, 通过通道连接融合能够提高网络性能。因此 Sandler M 等人<sup>[35]</sup>在 MobileNet 基础上结合残差网络结构提出了 MobileNet V2, 其标准卷积模块首先使用  $1 \times 1$  卷积核对输入数据特征通道数扩张, 其次用  $3 \times 3$  卷积核进行深度卷积提取单通道特征信息, 再用  $1 \times 1$  卷积对通道降维, 最后与残差模块进行通道拼接输出。MobileNet V2 还取消使用 Relu 函数对深度卷积以及通道拼接输出激活, 用于防止数据信息的丢失, 该网络在 ImageNet 竞赛数据集的分类预测精度达到 71%, CPU 运行时长 80ms, 相比于 V1 70% 和 123ms 的测试结果有效证明了 V2 网络保持轻量化的同时性能也有所升。Howard 等人<sup>[36]</sup>提出 MobileNet V3 卷积网络, 该网络将 SENet 提出的特征注意力机制融入 MobileNet V2, 还对部分网络结构进行调整, 设计新型非线性激活函数, 用神经网络架构搜索得到各层卷积核个数, 使得网络的性能进一步提高。Zhang X 等人<sup>[37]</sup>提出一种基于通道混排、分组卷积以及深度卷积的轻量化网络 ShuffleNet。ShuffleNet 标准卷积模块首先对输入数据进行分组卷积, 然后对结果进行通道混排, 最后将混排结果经深度卷积以及分离卷积输出。网络结构参照 ResNet, 使用分组卷积以及深度可分离卷积有效降低模型存储要求以及算力消耗, 通过对多通道数据实现重排清洗加强信息交互, 充分保证网络性能。ShuffleNet 与 MobileNet 相比复杂度更低且精确度更好, 在 ImageNet 相对误差达到 2.2%。Ma N 等人<sup>[38]</sup>针对轻量化模型设计提出四条原则: (1) 维度相同时内存访问达到最小; (2) 避免过多使用分组卷积; (3) 结构碎片化影响计算并行度; (4) 注意元素级操作。根据上述原则 Ma 等人基于 ShuffleNet 提出 ShuffleNet V2, 该网络创新提出通道分割(Channel split)用于取消 ShuffleNet V1 中分组卷积方法。该方法将输入特征图一部分维度的数据直接向后传输, 另一部分采用深度可分离卷积操作, 而后再将两种结果直接通道拼接, 最后对拼接结果通道混排输出。在 COCO 目标检测数据集, ShuffleNet V2 对比 ShuffleNet V1 与 MobileNetV2, 网络预测精度较优且在 GPU 运行速度提高 40% 与 16%。Han K 等人<sup>[39]</sup>提出轻量化模型 GhostNet, 该网络通过对特征图进行简单线性操作获得更多特征图, 以少量计算代价实现网络特征信息丰富, 在 ImageNet 分类数据集上得到 75.7% 的 Top-1 准确率。

### 1.2.3 目标检测算法

目标检测任务要求不仅判断输入数据包含物体种类，还要定位物体位置并用矩形框框出。现如今目标检测算法可分成两种：（1）传统目标检测算法，（2）基于深度卷积神经网络的算法。传统目标检测法由区域选择、特征信息提取<sup>[40]</sup>以及分类器构成。区域选择较多使用基于滑动窗口<sup>[41]</sup>的方法，特征信息主要涉及颜色、边缘、尺度不变特征（SIFT）<sup>[42]</sup>以及方向梯度直方图（HOG）特征<sup>[43]</sup>，分类器则有支持向量机<sup>[44,45]</sup>和 AdaBoost<sup>[46]</sup>。传统算法在区域选择以及特征提取上时间复杂度较高，最终分类准确率也较低。随着深度卷积神经网络不断开拓创新，将其融合到目标检测算法成为研究的热点。现如今，基于深度卷积神经网络目标检测算法<sup>[47]</sup>可分成两种：一种是基于候选区域的 Two-Stage 检测算法，另一种为基于回归的 One-Stage 检测算法。

#### （1）基于候选区域的 Two-Stage 检测算法

Grishick R 等人<sup>[48]</sup>于 2014 年提出 R-CNN，第一次将深度卷积网络融入到目标检测领域。该算法框架首先采用选择搜索算法（Selective Search, SS）<sup>[49]</sup>生成 2000 个候选框，并将不同尺寸候选框缩放至  $227 \times 227$ ；然后算法使用 AlexNet 卷积网络对候选框进行特征信息提取；最后使用 SVM 算法确定目标的类别，使用回归器对候选框进行位置矫正。He 等人<sup>[50]</sup>提出 SSP-Net 目标检测算法，通过特征金字塔将由 SS 算法得到的候选框统一尺度，避免 R-CNN 中候选框卷积计算量。2015 年 Grishick R<sup>[51]</sup>基于 SSP-Net 思想改进 R-CNN 提出 Fast-RCNN，该算法创新提出 ROI Pooling 层，避免了对候选框缩放操作，有效提高网络计算速度。算法还替换特征提取网络为 VGG16，SVM 分类层替换为 Softmax 函数，采用多任务学习即同时实现目标类别预测与位置回归计算，有效降低模型参数量与计算量，在 Pascal VOC 2007 取得 70% 的 mAP（mean Average Precision），但由于 Fast R-CNN 仍采用 SS 方法选择候选框，导致算法无法实现低延时检测。Ren S 等人<sup>[52]</sup>提出了 Faster-RCNN 目标检测算法，该算法提出候选框生成网络（RPN），降低候选区域选择带来的复杂计算量。Faster-RCNN 将候选框生成、特征信息提取、目标分类和位置回归融合进单个算法框架，实现目标检测算法端到端训练，网络检测精度以及速度得到较高提升，在 VOC2007 测试集 mAP 达到 73.2%。；2016 年，Dai J 等人<sup>[53]</sup>提出了 R-FCN 目标检测算法，该算法采用残差网络模块作为特征提取网络，使用卷积神经网络代替 ROI Pooling 层之后的全连接层，有效减少模型参数量。R-FCN 还加入位置感受得分图解决卷积平移不变性，将目标位置信息添加进 ROI Pooling 层中，在 Pascal VOC2007 测试集检测精度提升到 80.5%。但是该模型计算复杂，检测实时性较低。

#### （2）基于回归的 One-Stage 检测算法

2016 年, Redmon J 等人<sup>[54]</sup>首次将目标检测看待为一种回归问题, 提出单阶段目标检测算法 YOLO V1, 输入图像通过单次卷积就能得到图像中物体分类与所在位置结果。YOLO V1 相较于两阶段目标检测器在目标检测速度实现大幅提升, 但算法对于小目标、密集分布物体预测精度较低, 在 VOC2007 上 mAP 仅为 66.4%。2017 年, Redmon J 等人<sup>[55]</sup>在 YOLO V1 基础上提出 YOLO 9000, 该网络对卷积层输出都采用批量归一化处理, 同时采用了锚框机制并设计 Darknet-19 作为 YOLO 9000 的特征提取网络, 有效提升了目标检测算法准确度, 在 VOC2007 数据集上的 mAP 达到 78.6%;2018 年, Redmon J 等人<sup>[56]</sup>基于 YOLO 9000 提出 YOLO V3, 它使用了特征提取效果更好的 Darknet-53, 并且还采用多尺度分类预测方法以及锚框机制, 使得网络在保证检测速度同时, 预测精度也有效提高。2020 年, Bochkovskiy A 等人<sup>[57]</sup>提出了 YOLO V4 算法, 该算法是在原有 YOLO 系列目标检测算法基础上, 采用了近些年 CNN 领域中最优秀的优化策略, 从数据预处理、主干网络设计、网络训练方法、激活函数、损失函数等各个方面着手对算法进行优化处理, 在 COCO 数据集上, 可达 43.5%AP, 速度达到 65FPS。Liu W 等人<sup>[58]</sup>提出 SSD 目标检测框架, 该方法融合 YOLO 以及 Faster-RCNN 中锚框机制, 采用 VGG16 部分卷积层用于特征提取, 并新增 6 个卷积层获得更多特征信息, 使用多尺度检测实现对不同大小物体精确检测, 在 VOC2007 上测试得到 75.1%的 mAP, 检测速度达到 58 FPS。

### 1.3 论文研究内容与章节安排

#### 1.3.1 研究内容

针对传统基于云计算范式的人工视频监控系统存在的性能缺陷, 论文提出一种基于边缘计算的视频监控系统架构方案, 通过对煤层气井站监控视频数据检验, 有效证明设计方案具备可行性。论文主要研究内容包含以下三点:

(1) 提出基于边缘计算的视频监控系统架构方案。该系统由边缘设备、边缘服务器以及云端共同组成。边缘设备实现对现场数据实时采集、分析与处理; 边缘服务器实现对边缘设备采集的视频数据存储以及检测算法参数文件缓存; 云端则执行检测算法参数地迭代更新并定时推送到边缘服务器端。

(2) 设计面向边缘计算的深度卷积神经网络, 该网络保留 MobileNet 以及 SENet 结构理念的同时, 采用多尺度卷积核提升网络性能, 并结合 GhostNet 进一步实现网络轻量化。通过与其他深度卷积模型进行对比实验, 验证设计网络的有效性。

(3) 将本文设计的深度卷积网络结合到目标检测算法中,通过对比试验验证本文设计网络性能。选择综合指标较好的改进算法用于本文设计的视频监控系统架构方案中,实现对煤气气井站监控数据实时准确检测。

### 1.3.2 章节安排

此次论文共计七个章节,图 1.1 为章节结构示意图,具体内容如下:

第一章绪论。首先介绍论文研究背景和意义,其次介绍边缘计算国内外研究现状,再介绍了深度卷积神经网络以及轻量化网络实现的国内外研究现状,然后分析基于深度卷积神经网络的目标检测算法研究现状,最后介绍论文研究内容以及章节安排。

第二章基于边缘计算的视频监控系统设计。首先介绍边缘计算相关理论,指出边缘计算相较于云计算的发展优势,其次介绍边缘计算应用场景,再介绍用于设计采用的边缘计算设备 NVIDIA TX2 的规格参数,最后介绍设计的基于边缘计算的视频监测系统架构方案。

第三章深度卷积神经网络算法分析。首先介绍深度卷积神经网络基础知识,指出传统卷积网络存在的问题并提出人工设计轻量化网络的技术路线,然后以 MobileNet V1、V2 以及 V3 作为研究切入点,对三种网络结构进行介绍分析,为下一章设计新型轻量化深度卷积网络做好理论铺垫。

第四章面向边缘计算的深度卷积网络设计。首先阐述本文提出的深度卷积网络结构设计理念,其次具体介绍网络组成,再介绍实验使用数据集、超参数设计以及网络性能评价指标,最后通过与其他深度卷积网络进行对比实验来验证网络性能。

第五章基于 MMGNet 的目标检测算法设计。首先介绍 Faster-RCNN、SSD 以及 YOLOV4-tiny 算法网络结构、损失函数组成等相关知识,其次介绍用于对比实验的数据集和评价指标,最后将本文设计网络融入到三种算法得到对比实验结果分析,综合对比各类指标选择分类表现最好的改进算法运用到本文设计的视频监控系统架构方案。

第六章煤层气井站异常目标检测。首先对设计视频监控架构方案所应用场景进行简要介绍,再介绍用于实验验证的煤层气井站监控数据集,其次介绍实验所需评价指标,最后得到算法在嵌入式设备上的实验结果分析。

第七章结论。对本文研究内容做出全面的分析总结,指出存在的不足并给出未来研究展望。

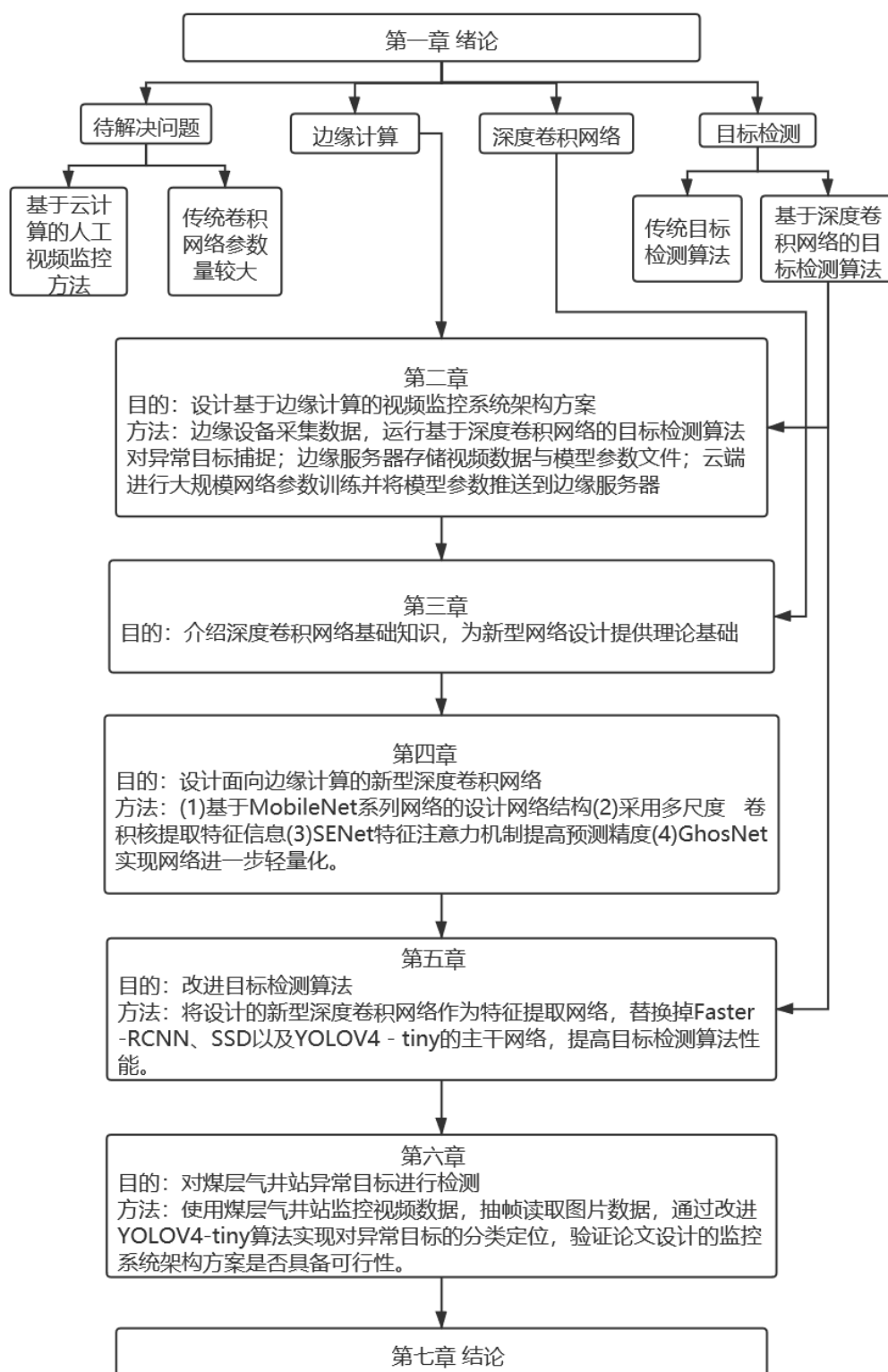


图 1.1 论文章节结构图

Fig. 1.1 Chapter structure diagram of the thesis

## 2 基于边缘计算的视频监控系统设计

### 2.1 边缘计算概念及特点

随着科技不断发展进步,可联网智能设备被广泛运用到现代社会的生产生活中。这些设备产生的数据流量呈爆炸式增长,周俊等人<sup>[59]</sup>指出直至 2020 年共计 500 亿台设备联网,预计 2025 年接入互联网设备将达到 5000 亿<sup>[60,61]</sup>。而智能设备越来越多的接入网络,传统云计算范式云端负载能力、数据存储以及带宽资源等方面将面临巨大挑战,使得云计算范式在诸如车联网、VR 游戏以及智能监控等低延迟、高带宽需要的应用场景存在系统缺陷。为解决云计算范式面对的诸多挑战,边缘计算范式被提出且被国内外学者广泛研究,也逐渐在万物互联时代发挥重要作用。边缘计算范式通过将云计算中心的部分算力下放到接近数据产生源头,通过具备一定计算能力的边缘设备实现对采集数据的实时分析处理,在一些低延时要求的应用场景优势明显。施威松<sup>[8]</sup>指出边缘计算是一种在网络的边缘实现数据的采集、处理与分析的全新计算模型。被计算的物体不但来自于物联网数据还包括来自云端的数据,边缘则是从数据源到云计算中心之间任何网络资源节点。相较于传统云计算范式,边缘计算具有如下三大特点:

(1) 就近计算提高实时性。传统云计算要求前端设备将采集数据由网络回传至云端,而随着计算节点增加势必会导致网络负载增加以及带宽资源不足,进而导致系统响应延时,这在高实时性要求的万物互联时代是致命缺陷。边缘计算则是将云中心部分计算任务卸载到网络边缘侧,有效的降低网络带宽的损耗,提高系统响应实时性。

(2) 隐私数据高效保护。万物互联网时代用户产生的数据与生活密切联系,全部上传至云端会增加隐私数据泄密风险。边缘计算是网络边缘对采集的数据进行实时分析,无需将数据全盘回传至云端,重要隐私数据也就不会在网络流通,从而降低了数据泄露的风险。同时,边缘计算作为分布式架构计算范式,对于网络攻击也具备一定的防御能力。

(3) 节省云端数据存储资源。随着可接入网络的智能设备增加,产生的数据流量也爆炸式增多,云计算中心数据存储、传输以及计算面临的压力也愈大。边缘计算范式将边缘设备产生的数据流量存储在边缘服务器,对边缘设备与云端实现解耦,有效地降低带宽资源的消耗以及云端计算负载,也一定程度降低了能耗。

需要指出的是边缘计算与云计算之间的关系是相辅相成。云计算可以通过边缘计算缓解云中心负载压力,有效节约网络带宽资源,提高系统响应速度;边缘计算也无法替代云计算完成大规模数据深度计算与储存。



## 2.2 边缘计算应用场景

随着信息技术的不断开拓创新,诸如移动手机、摄像头、汽车等设备逐渐在人类日常生活发挥重要作用,它们不但有较强的算力且能够进行网络通信,普及程度为万物互联时代提供有力的技术基础。随着对边缘计算技术研究不断深入,愈来愈多基于边缘计算的应用出现在大众视野,本节列出边缘计算如何对智慧城市、智能家居、车联网以及智能制造四种场景实现有效赋能<sup>[62-65]</sup>。

(1) 智慧城市:智慧城市是将城市当中各种传感器以及智能设备纳入到物网中实现整合运用,通过边缘计算为其构建高效网络计算范式,将智能设备采集的不同领域数据综合分析处理,实现信息查询、预测、异常信号检测等功能,为城市管理者提供全面的、有利于决策的实用信息,实现城市管理优化提升,也为广大用户提供低延时的高质量服务。

(2) 智慧家居:物联网快速发展让家居智能化,比如市场大量出现的扫地机器人、智能窗帘以及智能冰箱。传统云计算范式是家居通过 WIFI 将数据传输到云中心,存在一定的系统延时以及网络资源消耗。通过边缘计算,给家居赋以更强的计算能力,配合边缘计算低延时的特性,使得家居更加智能化。

(3) 车联网:随着嵌入式设备发展,车载计算机系统性能逐渐提高。将车载计算机系统结合人工智能技术,适用于边缘计算的应用场景逐渐被发掘,诸如自动驾驶技术、智慧交通等。车联网将车辆上传感器采集到的实时数据,通过边缘网络为用户提供实时交互,比如实时交通信息、智能路径导航、自动驾驶等。

(4) 智能制造:边缘计算与工业领域有着密不可分的联系。在工业 4.0 时代,工业制造势必实现智能化,设备的专业性以及复杂程度也会进一步提高,生产环节产生的实时数据也会愈加复杂,实时性控制要求也会越高。边缘计算运用到智能制造设备中,实现对设备的实时监控,结合云端强大的计算存储能力,更好的实现制造产业结构优化以及制造智能化。

## 2.3 嵌入式边缘计算设备

随着嵌入式设备的快速发展,尤其是 GPU,TPU 以及 NPU 的出现,边缘设备计算能力突飞猛进,能够实现对采集数据快速处理分析,为边缘计算发展提供硬件基础。NVIDIA TX2(如图 2.1 所示)是一款优秀的边缘计算设备,已在诸多领域得到运用<sup>[66]</sup>。TX2 具备非常强大的 AI 性能,搭载有 4GB PASCAL 架构 GPU 以及 256 个 CUDA 处理核心,并配备多种接口用于个性化功能开发,设备具体参数规格如表 2.1 所示。本文便是选择 TX2 作为视频监控系统架构方案的边缘设备端。

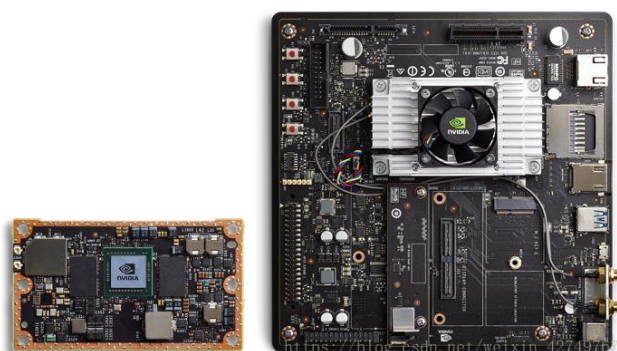


图 2.1 NVIDIA TX2 硬件示意图

Fig. 2.1 Hardware schematic diagram of NVIDIA TX2

表 2.1 NVIDIA TX2 技术规格

Tab. 2.1 Technical specifications of NVIDIA TX2

硬件名称	参数规格
GPU	NVIDIA Pascal™、256 颗 CUDA 核心
CPU	HMP Dual 2/2MB L2+Quad ARM@A57/2MB L2
内存	8 GB 128 位 LPDDR4
数据存储	32GB EMMC、SDIO、SATA
视频	4K×2K 60Hz 编码 (HEVC)、 4K×2K 60Hz 解码 (12 位)
显示器	2 个 DSI 接口、2 个 DP1.2 接口/HDMI 2.0 接口/eDP 1.4 接口
PCIE	Gen 2 1×4+1×1 或 1×2+2×1
USB	USB 3.0+USB2.0
连接	1 千兆以太网、802.11ac WLAN、蓝牙
其他	CAN、UART、SPI、I2C、I2S、GPIO

## 2.4 面向边缘计算的视频监控系统架构方案

在传统安全防护领域，业界多采用基于云计算范式的人工视频监控方法。该方法将前端现场采集视频数据传输到云端，通过云端计算得到数据处理结果，再由网络传输到客户端，监控人员通过客户端实现对异常目标捕捉。现如今监控系统所部署前端摄像头规模可能上千，视频数据传输带来的网络负载压力激增，云端数据存储以及算力需求增加，导致系统存在响应延时，不利于视频监控这一类高实时性要求应用场景。同时人工视频监控模式受限于监控人员职业素质影响，容易出现人为疏忽导致异常信号漏检、误

检,从而无法立即对检测到的异常目标做出处理决策。本文设计的系统架构方案便是利用边缘计算将云端部分计算任务卸载到网络边缘端,在数据源头通过边缘设备实现对现场数据实时采集、分析,有效的降低了数据传输延时。边缘计算设备通过分析结果自动执行控制策略,摆脱人为因素对监控质量的影响,做到以事件触发的决策响应,提高系统响应实时性。整体系统架构方案如下图 2.2 所示。

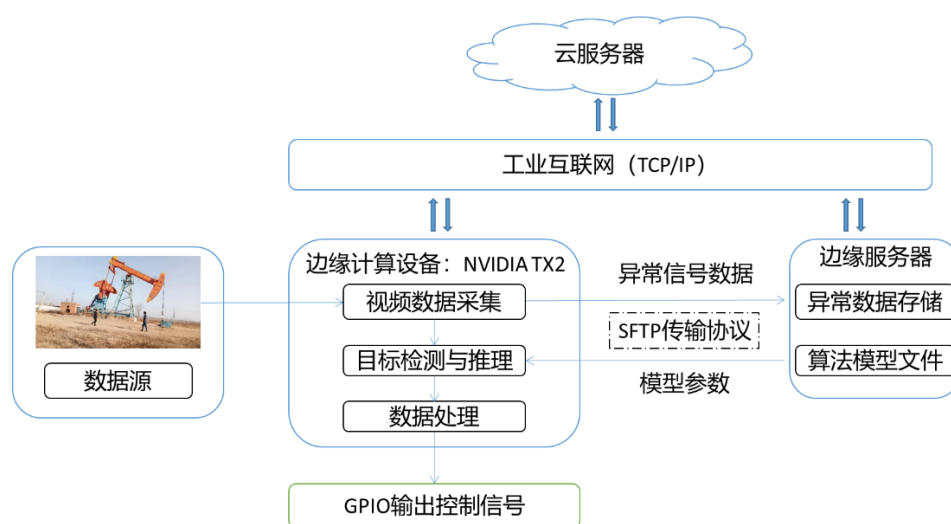


图 2.2 面向边缘计算的视频监控系统架构

Fig. 2.2 Video surveillance system architecture for edge computing

本文设计监控系统架构方案硬件由边缘计算设备、边缘服务器以及云端共同组成。其中边缘计算设备采用 NVIDIA TX2 开发套件；边缘服务器采用 PC 电脑；智星云服务器作为云计算中心。PC 以及云端服务器参数规格如表 2.2 所示。

表 2.2 云服务器以及边缘服务器参数规格

Tab. 2.2 Parameter specifications of cloud server and edge server

名称	作用	GPU	CPU	内存	硬盘
智星云 GPU	云端服务器	GTX 1080Ti 11GB	Inter Xeon E5	32GB	200GB
PC	边缘服务器	GTX 1050Ti 4GB	Inter Core i5	8GB	1TB

设计方案运行过程如下：

(1) 边缘计算设备 TX2 通过 OpenCV 算法程序打开板载摄像头并实时采集前端数据,对采集到的每一帧图像数据运行基于深度卷积网络的目标检测算法实现异常目标

检测，当识别到对应异常信号时，操作 GPIO 接口输出高电平用作后续控制决策的输入信号。TX2 对采集的图片数据可选择性上传至 PC 电脑缓存，方便操作人员进行历史数据查找校验。

(2) 采用 PC 电脑接受云端下发的算法模型参数并定时推送到 TX2 进行边缘计算设备更新。同时，PC 还存储 TX2 采集的异常目标数据实现备份，并规定时间将数据上传至云端用作模型训练。

(3) 智星云云端服务器是接受来自 PC 端的异常数据用于大规模的模型训练，训练完成后还需将模型参数文件下发至 PC 端。云端与 PC、TX2 之间数据传输协议均采用 SFTP。

## 2.5 本章小结

本章节首先介绍边缘计算的相关概念以及对比云计算的优势特点，其次介绍边缘计算四种实际应用场景体现出边缘计算具有重要研究与应用价值，再介绍本文使用的边缘计算设备 NVIDIA TX2 相关信息，最后具体给出基于边缘计算的视频监控系统设计方案。设计方案采用了云边协同工作的架构方案，TX2 作为边缘设备端，1TB 存储空间 PC 电脑作为边缘服务器，智星云服务器作为云端。方案中对异常目标检测方法采取基于深度卷积神经网络的目标检测算法，因此下章节将介绍深度卷积神经网络基本原理，为后续研究提供理论基础。

### 3 深度卷积神经网络算法分析

#### 3.1 网络基本结构

深度卷积神经网络是属于人工神经网络的一种网络结构，能够高效实现对输入图片颜色、边缘以及更复杂特征信息提取，对完成各类计算机视觉任务起到重要的作用<sup>[67-69]</sup>。在结构组成上，一个完备的卷积网络通常有输入层、卷积层、非线性激活层、池化层以及全连接层。

##### (1) 输入层

卷积神经网络的输入数据通常为多维数据，以张量的形式进行数值计算。比如输入一张 RGB 格式的图片对应的就是一个  $W \times H \times 3$  的三通道二维矩阵，其中  $W \times H$  为图片的分辨率大小。该矩阵每个位置数值等于图片的 RGB 数值，也就表示当前数据包含三种特征信息。相较于传统人工神经网络需要在训练数据前人工提取特征信息，卷积神经网络在特征选择与提取上更为简洁高效。

##### (2) 卷积层

卷积层通过卷积运算实现了网络逐层映射关系，是卷积神经网络提取特征信息的重要组成。改变用于卷积运算的卷积核大小以及通道数可获得不同感受野以及局部信息提取结果。卷积核大小一般小于或等于输入特征数据大小，以滑动窗口的表现形式在特征数据上由左至右、由上至下，并以固定步长交替运算，实现对整个输入数据特征提取。图 3.1 展示了步长 (stride) 为 1 的二维卷积流程图。

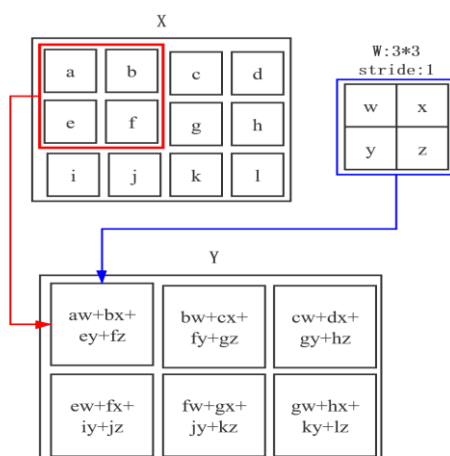


图 3.1 二维卷积计算流程示例

Fig. 3.1 An example of a two-dimensional convolution computation process

图 3.1 输入数据  $X$  为单通道的二维特征数据，卷积核  $W$  大小同样为单通道的二维矩阵，卷积计算数学表示为式 3.1 所示。

$$y(i, j) = \sum_m \sum_n X(i+m, j+n) \cdot W(m, n) \quad (3.1)$$

多通道特征数据对应相同通道数的卷积核，计算表示如式 3.2 所示。

$$y(i, j) = \sum_m \sum_n \sum_k X(i+m, j+n, k) \cdot W(m, n, k) \quad (3.2)$$

其中  $m, n$  表示输入数据尺度， $k$  表示通道数。

卷积运算的本质是实现了稀疏交互与权值共享。全连接神经网络通过权重矩阵实现前后网络层中神经元交互关系表达，该结构中后层网络神经元都与前一层任意神经元连接。如下图 3.2 所示。

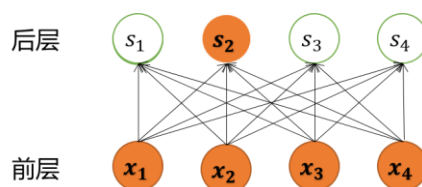


图 3.2 全连接神经网络结构简图

Fig. 3.2 Structure diagram of fully connected neural network

不同于全连接神经网络，卷积神经网络中由于卷积核大小一般是小于输入数据尺度，这样使得卷积核只能与部分输入数据进行交互，输出神经元只与上层特定区域神经元存在权重连接，这就是卷积运算的稀疏交互性，如图 3.3 所示。

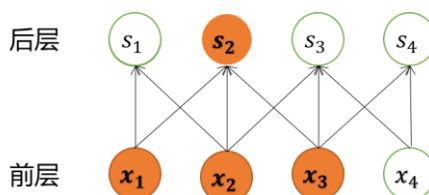


图 3.3 卷积层连接示意图

Fig. 3.3 Schematic diagram of convolution layer connection

稀疏交互在物理上实现了对输入数据局部特征信息的有效提取，并将提取到的局部特征融合可形成更为复杂抽象的特征。图 3.4 展示了人脸识别中间卷积层可视化结果，可以看出底层网络提取出边缘纹理等特征信息；中间层网络对底层信息组合得到五官复杂信息；上层网络将中间层特征信息组合得到人脸特征。

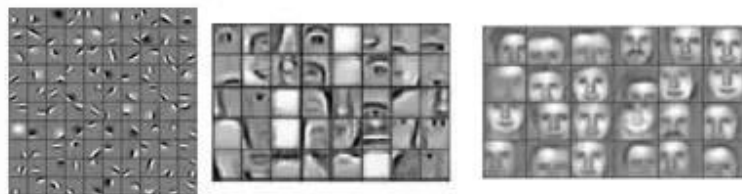


图 3.4 人脸识别中部分卷积层可视化结果

Fig. 3.4 Visualization results of face recognition in the middle part

参数共享指在模型的不同模块采用相同参数。全连接神经网络的权重值只能与输入数据的某一维度元素单独作用，而卷积运算中的卷积核采用在输入数据上滑动计算，使得每个权重参数都与输入数据的部分区域进行交互。因此卷积网络不需要针对每个输入位置学习特定的参数值，只需用一组数据即可适用于整体，大大减少模型参数量以及存储资源消耗。同时参数共享机制让卷积实现了平移等变性，即同一类物体在图片的不同位置都会得到相同识别结果。

### (3) 非线性激活层

由公式 3.1 可以知，卷积运算得到的是输入输出的线性关系。激活函数则改变输出与输入为非线性关系，有效防止多层网络推理等效于单层线性函数，提高了多层网络的拟合能力。常见用于非线性激活层的函数有 Relu, Sigmoid, Hard Sigmoid, Leaky Relu 等。Relu 函数如式 3.3 所示，曲线图如 3.5 所示。

$$f(z) = \begin{cases} z, & z \geq 0 \\ 0, & z < 0 \end{cases} \quad (3.3)$$

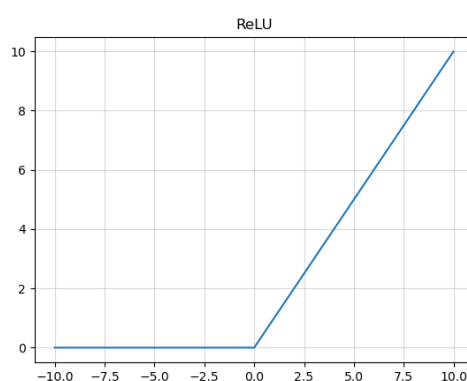


图 3.5 Relu 函数曲线图

Fig. 3.5 Graph of Relu function

Relu 函数计算过程简单，其非饱和性能够有效解决梯度弥散问题，提高网络的稀疏性。但是 Relu 函数在负数直接置 0 的特点使得网络在训练过程中，易造成神经元“死亡”，

使其无法被任何输入数据激活。这通常是由学习率设置过大导致权重以及偏置参数无法正常更新，最终训练失败。为缓解这一问题，研究人员提出了 LeakyRelu，其表达式如式 3.4 所示。

$$f(z) = \begin{cases} z, & z \geq 0 \\ az, & z < 0 \end{cases} \quad (3.4)$$

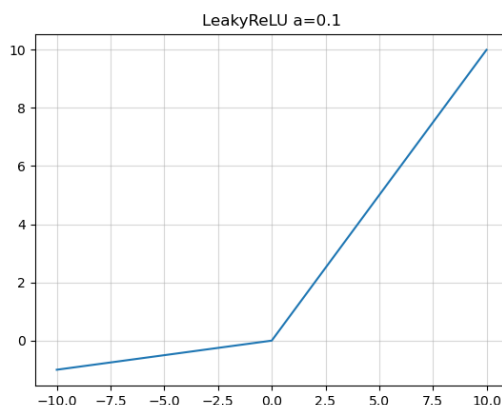


图 3.6 LeakyRelu 在  $a=0.1$  取时函数曲线

Fig. 3.6 The function curve of LeakyRelu when  $a$  takes 0.1

LeakyRelu 对比 Relu 函数，当输入  $z < 0$  时输出值为一个斜率为  $a$  的线性函数，该斜率值通常取一个很小的正数，如此便解决神经元“死亡”问题，防止负梯度的信息缺失。

Hard sigmoid 是 Sigmoid 的分段线性逼近函数，Sigmoid 函数如式 3.5 所示，Hard Sigmoid 函数如式 3.6 所示，可以看出 Hard sigmoid 简化激活函数在前向推理以及反向求导计算量，但都存在梯度消失问题。

$$f(z) = \frac{1}{1 + e^{-z}} \quad (3.5)$$

$$f(z) = \begin{cases} 0, & x < -\frac{b}{a} \\ az + b, & -\frac{b}{a} \leq z \leq \frac{1-b}{a} \\ 1, & z > \frac{1-b}{a} \end{cases} \quad (3.6)$$

两种函数图像如图 3.7 所示：



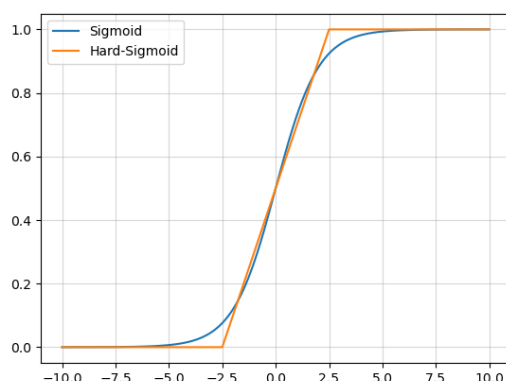


图 3.7 Sigmoid 与 Hard Sigmoid 函数曲线图

Fig. 3.7 The function graph of Sigmoid and Hard Sigmoid

#### (4) 池化层

池化层通常用来对卷积后维度较大的特征图进行降维处理。池化方法通常使用以下两种：一个是最大池化（Max Pooling），另一个是平均池化（Average Pooling），如图 3.8 所示。最大池化输出池化区域的最大值，该方式能够对由于网络参数误差导致的估计值均值偏移现象起到抑制作用，对边缘纹理等信息有较好提取能力；平均池化则输出池化区域所有数值平均值，有利于抑制由于相邻区域数据范围差距过大导致估值方差激增，实现对背景数据信息保存。

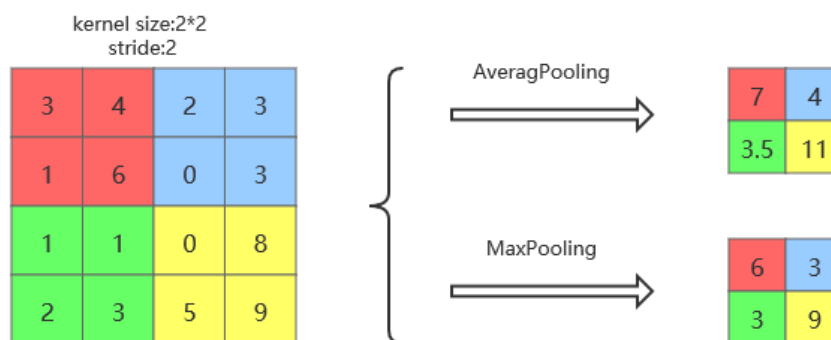


图 3.8 两种常用池化方式实例

Fig. 3.8 Examples of two common pooling methods

此外还有特殊的池化操作，比如相邻重叠部分池化以及金字塔池化。相邻重叠区域池化是使用远小于输入数据大小的池化步长滑动操作，这使得相邻两次的池化存在重叠部分。金塔池化是采用多尺度描述，将多种不同大小的矩阵对输入数据池化后，对结果

进行通道拼接作为下一个网络层的输入。池化操作实现输入数据降采样过程，能够使网络参数量较大幅度减少，并且可以保持平移、旋转、伸缩等操作的不变性。

#### （5）全连接层

全连接层（Fully Connected Layer, FC）一般在卷积网络末端使用，接受经过卷积以及池化操作后的特征数据，铺平展开（Flatten）成一维特征数据，然后通过线性组合以及非线性操作（激活函数）实现卷积网络特征组合，最后输出分类结果。全连接层结构如图 3.9 所示。

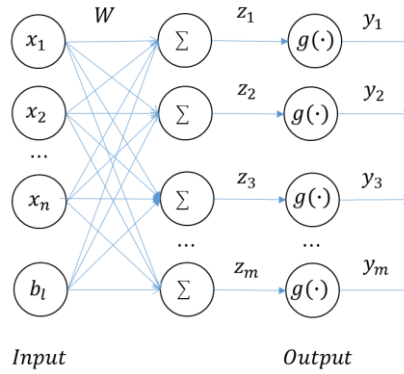


图 3.9 全连接层结构图

Fig. 3.9 Structure diagram of full connection layer

其中输入数据  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ ，偏置值  $\mathbf{b} = [b_1, b_2, \dots, b_m]^T$ ，输出数据  $\mathbf{y} = [y_1, y_2, \dots, y_m]^T$ ， $W$  为  $m \times n$  大小的矩阵， $g(\cdot)$  为非线性激活函数。全连接层计算公式如式 3.7 以及式 3.8 所示。

$$y_m = g(z_m) \quad (3.7)$$

$$z_m = \sum_{i=1}^n w_{mi} \cdot x_i + b_m \quad (3.8)$$

在多分类任务中，全连接网络输出通常还需要经过 Softmax 函数。Softmax 函数通过归一化数据处理将全连接层输出值映射到值范围在 (0, 1) 的概率分布，预测时将最大输出概率值对应的索引作为预测分类标签值。计算如式 3.9 所示。

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{i=0}^{N-1} e^{x_i}} \quad (3.9)$$

其中  $N$  为分类样本类别总数， $x_i$  为第  $i$  类全连接层输出值。

上述网络模块通过堆叠形成深度卷积网络，在图像数据分类预测以及目标检测发挥出不错的效果。各路学者为追寻更加出众的分类结果，网络的深度以及宽度越来越大，

导致基于深度卷积网络的目标检测算法特征提取网络参数增大、计算资源占用高。移动设备以及边缘嵌入式设备可能无法达到检测算法正常运行所需的存储要求以及算力要求，在低延时要求应用场景难以运行。因此，深度卷积网络的轻量化实现成为研究的热点。工业领域以及学术领域对轻量化模型的研究主要涵盖如下方面：(1)人工构造轻量化深度卷积网络<sup>[33-38]</sup> (2)基于神经网络架构搜索的自动化设计神经网络<sup>[70]</sup> (3)卷积神经网络压缩策略<sup>[71,72]</sup> (4) Auto ML 模型压缩方法<sup>[73]</sup>。本文基于现有的实验条件,将选择人工构造轻量化网络的方法，以 MobileNet 系列轻量化卷积神经网络作为研究切入点，介绍算法实现原理，为新型网络设计做好理论铺垫。

## 3.2 MobileNet 系列轻量化神经网络

### 3.2.1 MobileNet V1

2017 年, Howard 等人提出基于深度可分离卷积 (Depthwise Separable Convolution) 的轻量化深度卷积网络 MobileNet。深度可分离卷积由深度卷积 (Depthwise Convolution, DW) 与点卷积 (Pointwise Convolution, PW) 两部分组成, 如图 3.10 所示。深度卷积 (左图) 计算方式是将特征图的每个通道单独用二维卷积核进行特征信息提取, 再将结果按照原始通道顺序拼接; 点卷积实现方式 (右图) 与传统三维卷积形式相同, 只是将卷积核大小固定成  $1 \times 1$ , 实现对深度卷积结果特征信息线性组合进而得到全新特征。

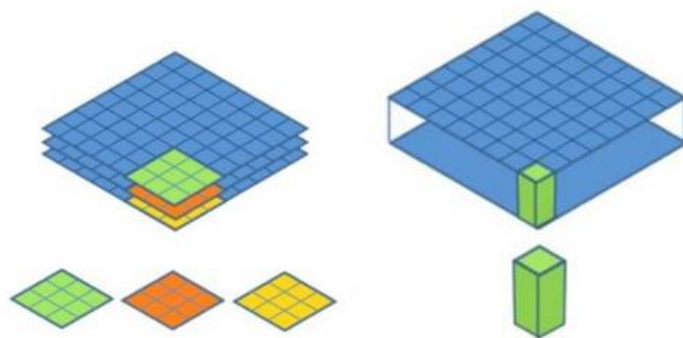


图 3.10 深度卷积与点卷积示意图

Fig. 3.10 Schematic diagram of DW and PW

对比传统标准卷积, 深度可分离卷积模式可有效降低算法运行与存储资源消耗, 实现了网络模型的轻量化处理。不考虑偏置, 当输入特征数据大小为  $M \times M \times H$ , 输出特征为  $M \times M \times Q$ , 其中  $H, Q$  为通道数, 传统三维卷积标准卷积核可表示为  $N \times N \times H$ 。传统三维卷积乘法计算量  $G$  以及参数量  $P$  为:

$$G = M \times M \times H \times Q \times N \times N \quad (3.10)$$

$$P = N \times N \times H \times Q \quad (3.11)$$

深度可分离卷积乘法计算量以及参数量分别为：

$$G' = M \times M \times H \times N \times N + H \times Q \times M \times M \quad (3.12)$$

$$P' = N \times N \times H + Q \times H \quad (3.13)$$

深度可分离卷积与标准卷积相比，乘法计算量以及参数量分别减少了：

$$\frac{G'}{G} = \frac{M \times M \times H \times N \times N + H \times Q \times M \times M}{M \times M \times H \times Q \times N \times N} = \frac{1}{Q} + \frac{1}{N^2} \quad (3.14)$$

$$\frac{P'}{P} = \frac{N \times N \times H + Q \times H}{N \times N \times H \times Q} = \frac{1}{Q} + \frac{1}{N^2} \quad (3.15)$$

MobileNet V1 对网络结构也进行轻量化设计。首先将 VGG 网络中标准卷积模块进行如图 3.11 所示的改变，其中 BN 层是对数据进行归一化处理。同时对 VGG 末端的 3 个全连接层替换为  $7 \times 7$  平均池化层（Average Pooling）以及一层  $1024 \times 1000$  全连接层。去除 VGG 中用于数据降维的最大池化层，用步长为 2 的深度卷积替换。

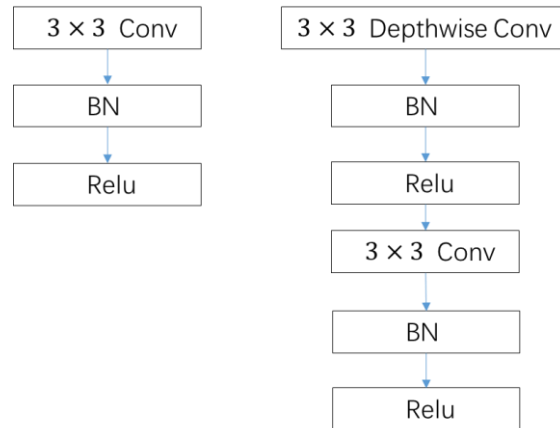


图 3.11 传统卷积流程与深度可分离卷积流程

Fig. 3.11 Traditional convolution process and depth separable convolution process

除此之外，V1 还设置宽度因子以及分辨率因子两个超参数，分别用于改变输入输出特征数据的通道数以及输出特征尺度，在保持一定预测精度下，有效降低模型的计算量以及参数量。表 3.1 摘自 MobileNet 原文中对比 VGG 以及 GoogleNet 在 ImageNet 竞赛数据集表现，可以看出在分类性能上三种网络相差较小，但在模型参数量以及计算量上 MobileNetV1 拥有更为优异表现。

表 3.1 MobileNet、GoogleNet 以及 VGG16 在 ImageNet 数据集分类表现

Tab. 3.1 Classification performance of MobileNet, GoogleNet and VGG16 in ImageNet dataset

Model	ImagNet Accuracy	Million Mult-Adds	Million Params
MobileNet	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG16	71.5%	15300	138

### 3.2.2 MobileNet V2

MobileNet V2 在 V1 基础上引入 ResNet 网络设计的“瓶颈式”（bottleneck）网络结构，如图 3.12 所示。bottleneck 结构先用点卷积压缩降低模型参数，在通过大小传统三维卷积提取特征信息，最后再由点卷积扩张到目标维度。V2 借鉴 bottleneck 结构将其改编为“橄榄球型”（如图 3.13 所示），即先扩张特征通道，最后压缩通道输出，同时还将传统卷积替换为深度卷积，降低了卷积计算量。

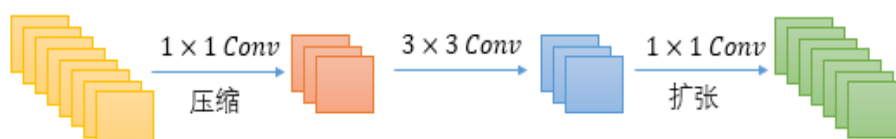


图 3.12 残差网络”瓶颈式”结构

Fig. 3.12 "Bottleneck" Structure of Residual Network

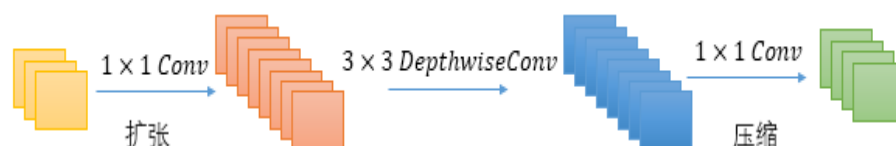


图 3.13 MobileNetV2 “橄榄球”结构

Fig. 3.13 "Rugby" structure of MobileNet V2

V2 通过点卷积实现输入数据通道扩张，丰富了网络特征信息，有利于抑制模型的过拟合。网络还取消了压缩通道输出后进行 Relu 激活操作，而是采用线性激活(Linear)，这是由于 Relu 函数对于负值输出节点直接置零，导致部分神经元无法进行梯度更新，造成特征信息的缺失。V2 标准卷积模块结构如图 3.14 所示。

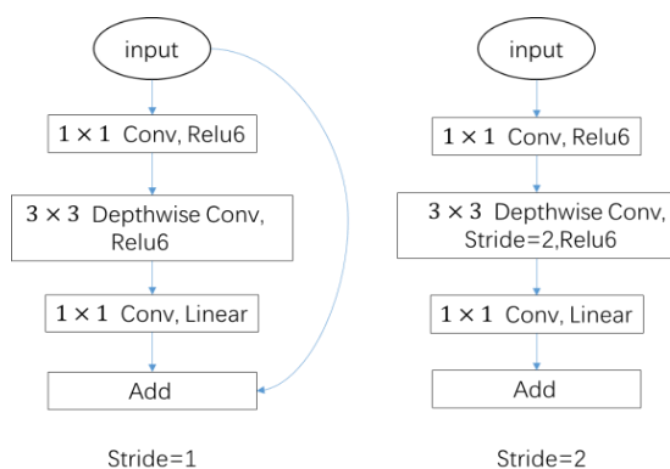


图 3.14 不同深度卷积步长对应的标准卷积结构

Fig. 3.14 Standard convolution structures corresponding to different depthwise convolution strides

### 3.2.3 MobileNet V3

MobileNet V3 在 V2 的基础上采取了四点改进措施：（1）在结构上引入了 SENet 中通道注意力机制。首先将单层特征图进行全局平均池化，其次经过全连接层得到每一层通道的权重值，通过权重值大小评估出每个通道所提取的特征重要性，最后权重值与特征层对应位置相乘，对特征信息提取结果影响较大通道予以加强，影响程度较小通道予以抑制。加入 SENet 的单层结构后，网络标准卷积模块如图 3.15 所示，通过实验验证网络性能得到有效提高，且 SENet 结构并未对网络计算效率造成损害。（2）对卷积核数

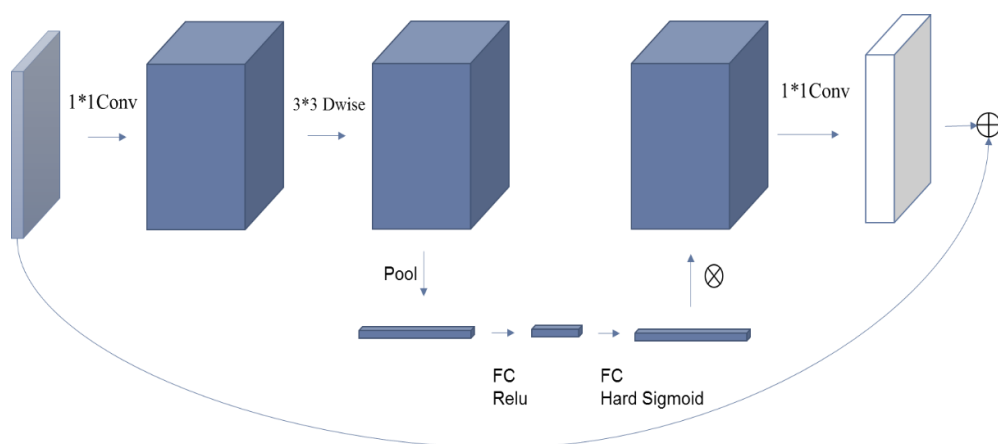


图 3.15 MobileNet V3 标准卷积块

Fig. 3.15 Standard convolution block of MobileNet V3

量采用神经网络架构搜索方法确定；（3）对 V2 末端网络结构做出调整，将平均池化

层替换为 $3 \times 3$ 深度卷积以及 $1 \times 1$ 点卷积，使得模型保持一定精度的同时有效降低模型计算量；（4）将 V2 中的 Swish（式 3.16）非线性激活函数替换为全新的非线性激活函数 Hard-swish（式 3.17），函数图像如图 3.16 所示，可以看到 Hard-swish 函数同 Swish 有相似的效果，但是 Relu6 函数计算复杂度远小于 Sigmoid 函数，有效降低模型前向推理以及梯度更新时计算消耗。

$$\text{Swish}(x) = x \cdot \text{Sigmoid}(x) \quad (3.16)$$

$$\text{Hardswish}(x) = x \cdot \frac{\text{Relu6}(x+3)}{6} \quad (3.17)$$

$$\text{Relu6}(x) = \min(\max(0, x), 6) \quad (3.18)$$

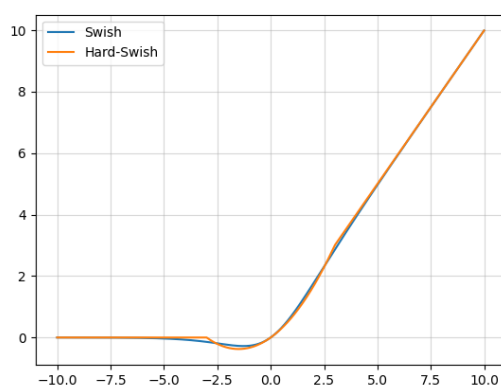


图 3.16 Swish 与 Hard-swish 函数图像

Fig. 3.16 Function graph of Swish and Hard-swish

### 3.3 本章小结

本章节首先介绍深度卷积网络基本构成，指出深度卷积网络由于存在大量参数冗余导致网络模型巨大，在嵌入式边缘设备难以有效部署。然后阐明采用轻量化卷积网络的研究思路，通过将轻量化网络 MobileNet V1、V2 以及 V3 作为研究切入点，对三种网络结构原理进行介绍分析，为下章节新型网络设计提供理论基础。



## 4 面向边缘计算的深度卷积网络设计

### 4.1 MMGNet 深度卷积网络

边缘计算设备存储空间、运行内存以及算力相较于云端较小，基于深度卷积网络的目标检测算法由于特征提取网络参数较大，计算资源占用高，导致算法难以在边缘设备实时运行。因此本文采用人工设计轻量化网络方式提出一种适用于边缘设备的深度卷积网络 MMGNet (Multi-scale convolution network based on MobileNet and GhostNet) 用于目标检测，该网络是基于 MobileNet 系列与 GhostNet 网络的、采用多尺度核提取特征信息的深度卷积神经网络。对该网络的结构设计包括如下四个方面：

(1) 网络基本结构上采用 MobileNet V2 中的“先扩张，后压缩”方法用于提升网络信息提取能力，同时也采用 SENet 的通道注意力机制进一步提升网络精度。

(2) 构建 MCblock (Multiscale Convolution Block), 该模块基于深度可分离卷积有效降低参数量与计算量，采用多种尺度卷积核（论文使用  $3\times 3$ 、 $5\times 5$  与  $7\times 7$ ）对点卷积输出数据进行深度卷积，用于提取更多的特征信息，结合 SENet 注意力机制对多种尺度卷积核进行重要性评估，然后将不同的分离卷积结果进行通道拼接，这样做既以较小参数量实现了通道的扩张也模仿人通过不同视觉观察事物的特性。

(3) 在 GhostNet<sup>[39]</sup>的研究中发现，卷积网络训练结束过后的特征图中存在着信息的冗余。因此可以先对输入数据进行较小维度卷积输出，再对输出通过线性运算(直接进行通道拼接)生成最终特征图，有效减少网络参数以及运行计算量。结构如图 4.1 所示。

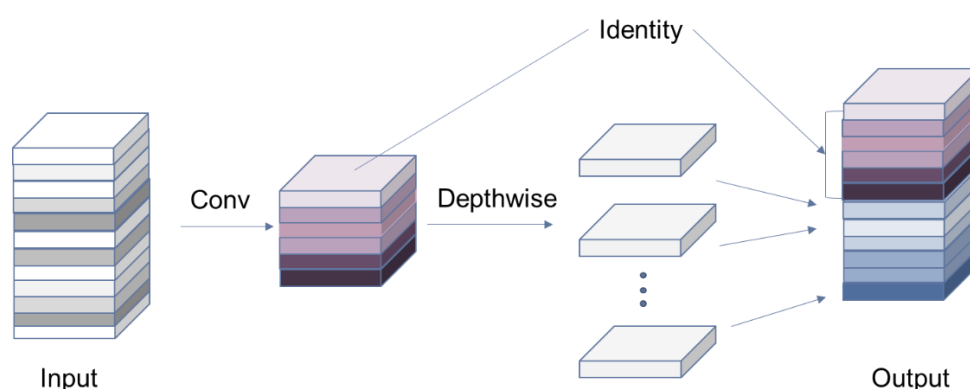


图 4.1 GhostNet 标准模块  
Fig. 4.1 Standard module of GhostNet



(4) 在网络结构末端采用点卷积替代全连接层, 避免全连接层过多参数量导致模型大小以及计算复杂度增加, 同时有效避免训练阶段参数过拟合。

综上所述, MMGNet 的标准卷积模块 Mbotleneck 如图 4.2 所示。

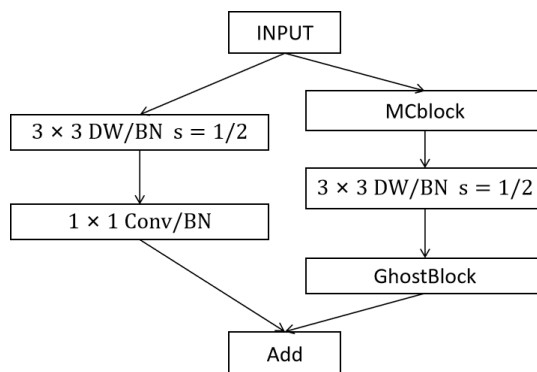


图 4.2 Mbotleneck 模块结构图

Fig. 4.2 Structural drawing of Mbotleneck module

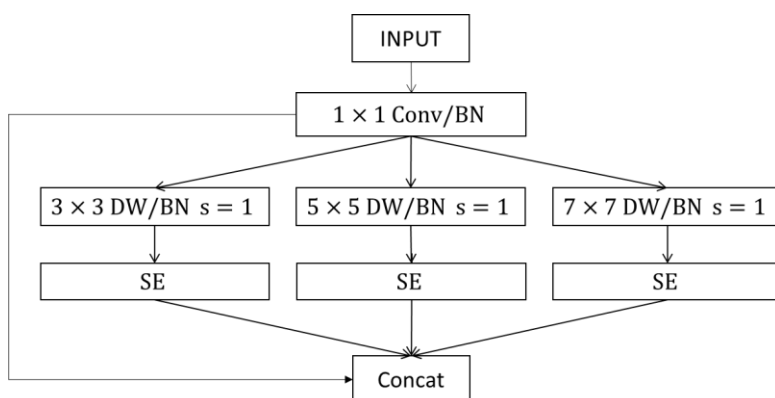


图 4.3 MCblock 结构图

Fig. 4.3 Structural drawing of MCblock

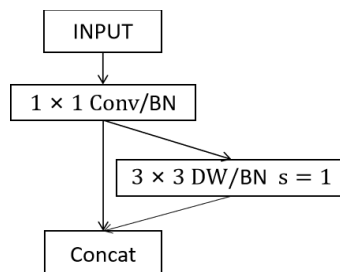


图 4.4 GhostBlock 结构图

Fig. 4.4 Structural drawing of GhostBlock

其中DW/BN表示对数据首先采用步长为 1 深度卷积，然后批归一化处理（Batch Normalization）； $s$  表示卷积步长， $s=1$  时输入特征图尺寸不变， $s=2$  时尺寸缩小一半。标准卷积模块堆叠形成深度卷积网络。当输入数据为大小 $112 \times 112 \times 3$ 的图片设计网络结构如表 4.1 所示：

表 4.1 输入数据 $112 \times 112 \times 3$ 时设计的轻量化网络结构Tab. 4.1 Lightweight network structure designed when the input data dimension is  $112 \times 112 \times 3$ 

Input	LayerName	Exp	Out_channels	SE	Stride
$112 \times 112 \times 3$	Conv2D_5	—	16	—	2
$56 \times 56 \times 16$	Mbottleneck	48	24	—	1
$56 \times 56 \times 24$	Mbottleneck	64	24	—	1
$56 \times 56 \times 24$	Mbottleneck	64	24	—	1
$56 \times 56 \times 24$	Mbottleneck	144	64	1	2
$28 \times 28 \times 64$	Mbottleneck	200	64	—	1
$28 \times 28 \times 64$	Mbottleneck	200	64	—	1
$28 \times 28 \times 64$	Mbottleneck	224	96	1	2
$14 \times 14 \times 96$	Mbottleneck	256	96	—	1
$14 \times 14 \times 96$	Mbottleneck	256	96	1	1
$14 \times 14 \times 96$	Mbottleneck	256	96	—	1
$14 \times 14 \times 96$	Mbottleneck	256	120	1	1
$14 \times 14 \times 120$	Mbottleneck	256	120	1	1
$14 \times 14 \times 120$	Mbottleneck	288	160	1	2
$7 \times 7 \times 160$	Mbottleneck	320	160	—	1
$7 \times 7 \times 160$	Mbottleneck	320	160	1	1
$7 \times 7 \times 160$	Mbottleneck	320	160	—	1
$7 \times 7 \times 160$	Conv2D_1	—	720	—	1
$7 \times 7 \times 720$	AvgPool	—	720	—	1
$1 \times 1 \times 720$	Conv2D_1	—	100	—	1
$1 \times 1 \times 100$	Flatten/Classifier	—	—	—	—

在表中，Input 表示输入特征图维度大小，Exp 表示通道扩张数，Out\_channels 表示输出特征图通道数；SE=1 表示加入通道注意力模块；Stride 表示卷积运算步长，若取值为 1，表示经过该网络层数据分辨率不变，若取值为 2 表示数据分辨率减半；Conv2D <sub>$n$</sub>  表示卷积核大小为  $n$  的卷积运算；AvgPool 表示对输入数据采取全局平均池化处理；

*Flatten/Classifier* 表示对卷积网络输出结果展开成一维向量,通过 *Softmax* 函数求取每个标签概率值,取最大概率值索引作为预测标签值。

网络在接近输入端使用一个卷积实现分辨率缩小 1/2,有效降低网络运算量。降低特征图尺寸后,使用多个不变分辨率的 Mbotleneck 进行特征提取,然后通过带降采样的 Mbotleneck 降低数据分辨率。网络中通道扩展数取值为输入数据通道数的 2-3 倍,当输入数据分辨率高但通道数低时,选择较大扩张倍数丰富特征信息;当数据分辨率较低但通道数较高时选择较低倍数扩张,避免高通道扩张倍数导致参数量与显存消耗增加。网络末端使用全局池化和点卷积实现输出分辨率为 1 的特征数据,其通道数等于分类类别总数,经过 Flatten 得到一维向量,再由 softmax 函数预测最终类别,对比全连接层可有效降低带来参数量与显存消耗增加,如下表 4.2 所示。

表 4.2 全局池化/点卷积的末端结构同全连接层在参数量与显存消耗对比结果

Tab. 4.2 Comparison between the end structure of global pooling/coupon product and the full connection layers in terms of parameter quantity and memory consumption

结构名称	参数量	显存消耗
Structure name	Params (MB)	Memory consumption (MB)
全局池化/点卷积	10.0	84.9
全连接层	94.4	295.6

## 4.2 实验结果与分析

### 4.2.1 实验环境

本次实验采用智星云服务器,相关参数规格如下表 4.3 所示:

表 4.3 实验环境硬件参数规格

Tab. 4.3 Hardware parameter specification of experimental environment

CPU	内存	GPU	硬盘
Inter Xeon E5	32GB	NVIDIA GTX1080ti 11GB	200GB

软件编写环境为 Windows Python 3.7, CUDA10.0, 深度学习框架为 keras 以及 TensorFlow\_GPU\_1.15, 另外还使用到 OpenCV、Matplotlib 等框架用于图片数据处理以及实验结果可视化。

### 4.2.2 数据集

对设计网络进行训练测试的数据集为 CIFAR100 数据集。该数据集涉及 100 个物体类，每个类涵盖 600 张图片，合计 60000 张。CIFAR100 的训练集与测试集划分比例为 5:1。

由于深度卷积网络含有大量参数，训练过程需要大规模数据集。通过数据增强技术可增强单张图片数据信息丰富程度，实现低成本提高训练数据集数据量，有效增加算法泛化能力，抑制在训练过程中网络过拟合现象。实验中采取对训练数据集进行随机旋转，区域裁剪，添加高斯噪声以及增加图片对比度、亮度、饱和度等方法，数据增强效果如图 4.5 所示。

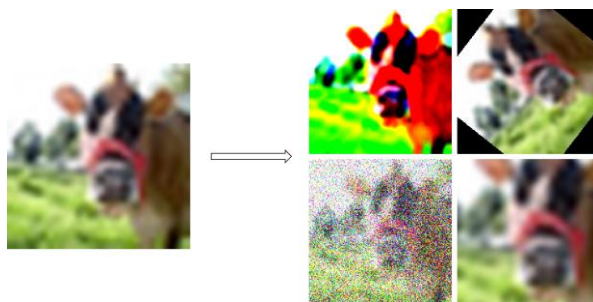


图 4.5 数据增强效果示意图

Fig. 4.5 Schematic diagram of data enhancement effect

### 4.2.3 评价指标

对网络模型的评价指标为模型参数量 (Params)、显存消耗 (Memory consumption, Mc) 以及 Top-1 准确率 (Accuracy, Ac)。其中，模型参数量大小以及显存消耗用于衡量模型轻量化程度，Top-1 准确率判断网络分类性能。

#### (1) 模型参数量

卷积神经网络的参数量 (Params) 由卷积核权重值以及偏置共同构成。不同的卷积形式对应不同参数量计算方式。对于传统三维卷积，单个卷积层参数量计算如式 4.1 所示。

$$Params_{conv} = C_{out} \times (K_h \times K_w \times C_{in} + 1) \quad (4.1)$$

其中  $K_w$ ,  $K_h$  代表卷积核的长与宽； $C_{in}$  表示输入特征图维度； $C_{out}$  表示输出特征图维度；加 1 表示偏置量。

深度可分离卷积由深度卷积与分离卷积构成，深度卷积的参数计量如式 4.2，分离卷积参数量计算如式 4.3。

$$Params_{PW} = C_{out} \times (1 + C_{in}) \quad (4.2)$$

$$Params_{DW} = C_{in} \times (K_h \times K_w + C_{out}) \quad (4.3)$$

全连接神经网络参数量计算：

$$Params_{FC} = N_{out} \times (N_{in} + 1) \quad (4.4)$$

其中  $N_{in}$  通常是将卷积运算结果经 Flatten 操作得到的一维向量长度；加 1 同样表示偏置； $N_{out}$  表示全连接层神经元个数。综上所述，网络模型参数总量为所有网络层参数量之和。

### (2) 显存消耗

显存是专门用于存储将被处理或是处理过的图像数据，卷积网络运行过程中对显存的消耗可以有效衡量网络性能。该指标由模型参数量以及前向推理过程中所有输出特征图大小共同决定，计算式如式 4.5 所示。

$$Mc = Params + \sum W_{out} \times H_{out} \times C_{out} \quad (4.5)$$

其中  $W_{out} \times H_{out}$  表示每一层输出特征图分辨率大小。

### (3) Top-1 准确率

Top-1 准确率表示预测类别标签与实际类别相等的样本数占总样本数比例，计算过程如式 4.6 与 4.7 所示：

$$Ac = \frac{1}{N} \sum_{i=0}^N I_i \quad (4.6)$$

$$I_i = \begin{cases} 0, & l_i^{pre} \neq l_i^{true} \\ 1, & Others \end{cases} \quad (4.7)$$

其中  $N$  为样本总数目， $l_i^{pre}$  与  $l_i^{true}$  分别表示预测标签值与实际标签值。

## 4.2.4 实验结果与分析

实验过程中对网络参数使用随机梯度下降算法进行迭代更新，初始学习率设置为  $1 \times 10^{-2}$ ；批量训练的样本数目为 64；损失函数采用多分类交叉熵损失函数，计算过程如式 4.8 所示。

$$loss = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} y_{ik} \log(p_{i,k}) \quad (4.8)$$

其中  $N$  为样本总数； $K$  为多分类任务中类别总数； $y_{ik}$  表示第  $i$  个样本对类别真实标签值是否等于  $k$ ，相等取值为 1 反之为 0； $p_{i,k}$  表示第  $i$  个样本对预测类别为  $k$  的概率值。

训练过程中当迭代到一定次数，损失函数可能出现稳定不变情况，模型参数无法进一步有效更新，需要对学习率参数进行修改。变化示意图如图 4.6 所示。

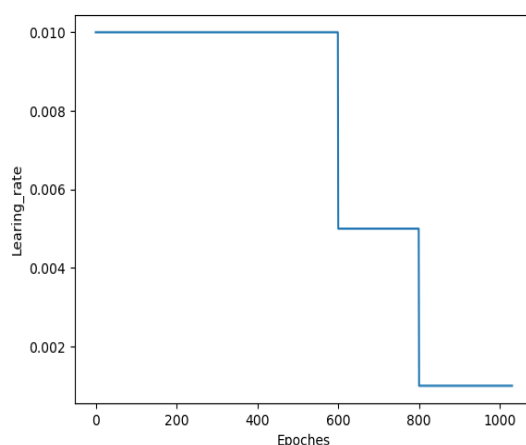
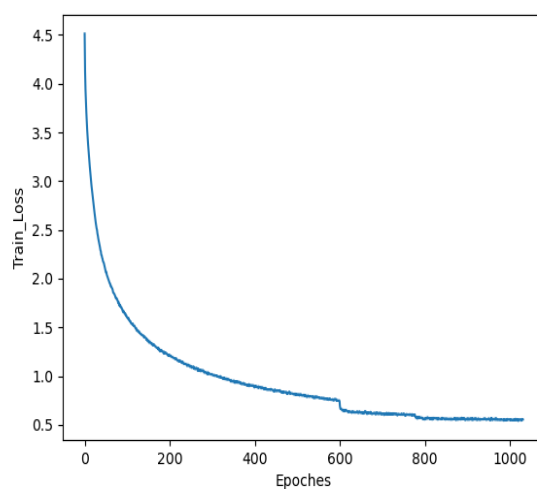


图 4.6 学习率曲线

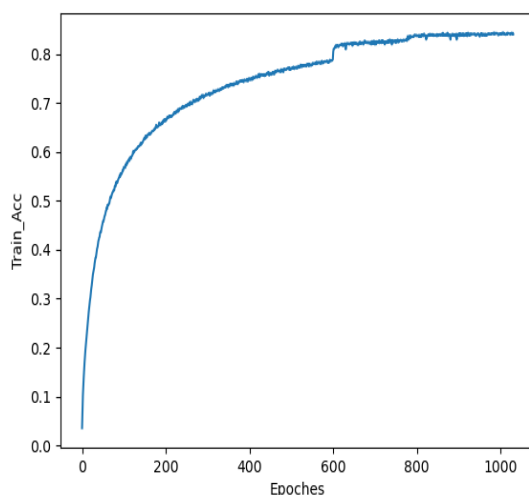
Fig. 4.6 The curve of learning rate

图 4.7 表示的训练集在 1000 个 Epoches 后 Loss 以及准确率值变化曲线，可知模型基本收敛。



(a) 训练集 loss 变化曲线

(a) Loss curve of training set



(b) 训练集准确率变化曲线

(b) Accuracy curve of training set

图 4.7 训练集 loss 以及准确率值变化曲线

Fig. 4.7 Accuracy and loss value change curve of training set

在 CIFAR100 测试集上检测，得到了 71.52% 的 Top-1 准确率。将 MMGNet 对比其他经典深度卷积神经网络，在参数量、运行显存消耗以及 Top-1 准确率表现如表 4.4 所示：

表 4.4 MMGNet 与部分卷积神经网络在参数量、显存消耗以及 Top-1 准确率对比

Tab. 4.4 Comparison of parameters , running memory and top-1 Accuracy between MMGNet and partial convolution neural network

模型名称	参数量	显存消耗	准确率
Model Name	Params (MB)	Mc (MB)	Top-1 Ac (%)
RCNN-160[74]	—	—	68.25
MIM[75]	—	—	70.8
VGG19[23]	228	243.72	72.3
ResNet50[27]	97.8	133.64	77.4
Inception V3[26]	22.74	103.83	77.2
MobileNet[34]	4.05	32.23	68
MobileNet V2[35]	2.36	34.8	69.1
<b>MMGNet</b>	<b>10.0</b>	<b>84.9</b>	<b>71.52</b>

通过表 4.4 实验结果可知, 对比 RCNN-160 以及 MIM 传统深度卷积网络, 论文设计的 MMGNet 在 CIFAR100 取得的 71.52% 预测准确率高于两种对比网络; 对比 VGG19、ResNet50 、InceptionV3 深度卷积网络, MMGNet 在预测准确率参数略低, 但是在参数量与显存消耗上有较大幅度的降低, 其中网络对比 VGG 模型参数量减少了 95%, 运行显存消耗缩小 65.1%, 证明网络实现一定轻量化处理; 对比 MobileNet 与 MobileNet V2 两种轻量化网络, 本文设计网络虽然在参数量以及显存消耗上有所增加, 但在预测准确率上高于两种算法。综上所述, MMGNet 深度卷积神经网络实现轻量化同时具备较好的预测精度。

### 4.3 本章小结

本章节首先介绍 MMGNet 设计理念, 再具体展示网络结构; 然后介绍实验使用数据集基本信息、数据增强方法、网络训练策略以及相关评价指标; 最后通过对比试验验证了本文设计的 MMGNet 实现深度卷积网络轻量化并具备一定的分类精度。下一章将 MMGNet 结合 Faster-RCNN、SSD、YOLOV4-tiny 目标检测算法进一步验证网络性能与实用性, 为后续监控系统架构中实现异常目标检测提供可靠的方案支持。



## 5 基于 MMGNet 的目标检测算法设计

### 5.1 目标检测算法

#### 5.1.1 Faster-RCNN 算法

##### (1) 算法结构

Faster-RCNN 是 Two-Stage 目标检测算法的代表作之一，算法结构如图 5.1 所示。

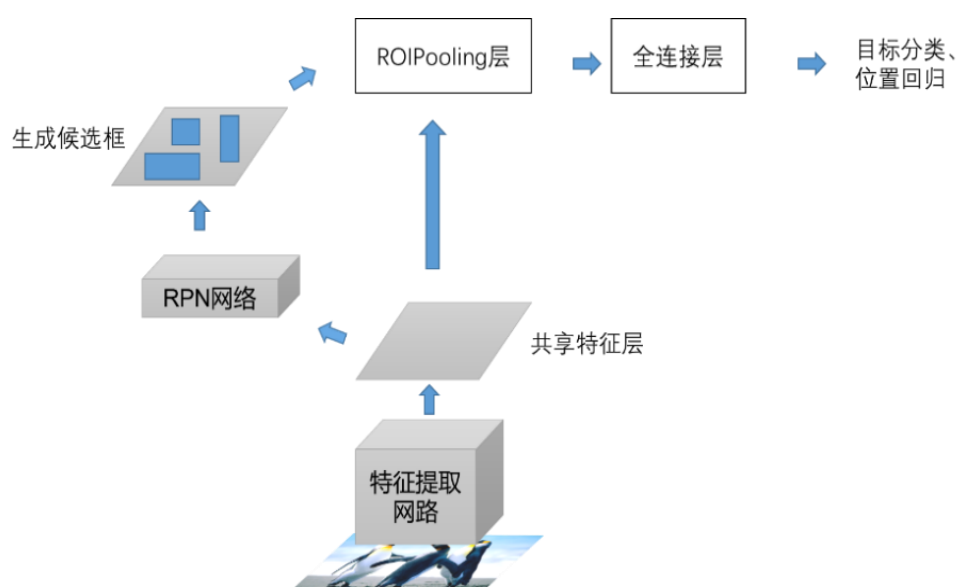


图 5.1 Faster-RCNN 网络结构图

Fig. 5.1 Network structure diagram of Faster-RCNN

该算法通过设计一个如图 5.2 所示的候选框生成网络（RPN）<sup>[52]</sup>，有效缓解以往基于 SS（Selective Search）算法在候选框生成耗时巨大以及提取的候选区域包含大量背景信息导致候选框质量较差等问题发生。RPN 网络首先对特征提取网络输出数据采用  $3 \times 3$  标准卷积，然后并行运算两次点卷积分别输出维度为  $2k$  以及  $4k$  特征数据，前者经过 Softmax 函数实现前景框与背景框的二分类预测，后者用于预测框与设置锚框（Anchor）的位置回归校正。RPN 网络结构如图 5.2 所示。

经过 RPN 得到的候选框并非全部作为后续网络输入，这是由于候选框不仅数量众多且大部分并未包含有目标物，正负样本严重不均衡，全部使用会降低分类准确度。因此 Faster-RCNN 首先比较目标真实框与候选框之间的交并比（IoU），然后把每个框的 IoU 值与设定阈值比较，筛选出符合规则的正负样本。IoU 结构如图 5.3 所示。

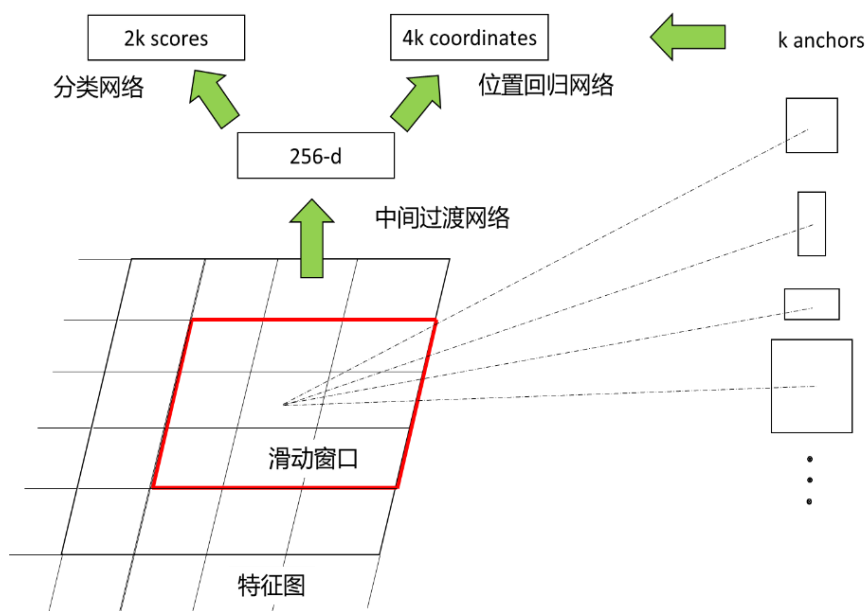


图 5.2 RPN 网络原理示意图<sup>[52]</sup>

Fig. 5.2 Schematic diagram of RPN network

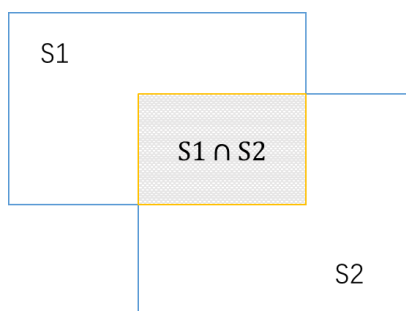


图 5.3 IoU 结构图

Fig. 5.3 Structure diagram of IoU

$s_1$  与  $s_2$  分别代表目标框面积与候选框面积， $S_1 \cap S_2$  代表交叉面积，则：

$$IoU = \frac{S_1 \cap S_2}{S_1 \cup S_2} \quad (5.1)$$

其中  $S_1 \cup S_2 = S_1 + S_2 - S_1 \cap S_2$ 。Faster-RCNN 通过 IoU 值筛选正负样本框的准则如下：

①若该候选框与目标框得到的 IoU 为最大值，则判断该候选框包含目标类（前景框），为正样本。

②若该候选框在任何目标框得到的 IoU 大于 0.7，则该目标框包含目标，即为正样本。

③若该候选框在任何目标框得到的 IOU 小于 0.3，则该目标框不包含目标(背景框)，即为负样本。

④位置信息超出边界或者 IoU 值在 0.3 到 0.7 之间的候选框不纳入计算。

由于同类别候选框可能存在大量冗余，因此在选出正负样本后还需经过非极大抑制算法（NMS）进行筛选。NMS 算法首先对预测结果为相同类候选框按照 IoU 值排序，将 IoU 值最大的候选框作为对比框；然后计算剩余候选框与对比框 IoU 值，通过与设定阈值比较排除低于阈值的候选框，选择最大 IoU 值候选框作为新对比框，循环往复直到候选框个数满足要求或者没有剩余。经 NMS 得到的候选框按照类别随机选择实现正负样本均衡，筛选结果将输入至 ROI Pooling 层。

ROI Pooling 层根据经过筛选的候选框所包含的位置信息，映射到特征提取网络输出层对应位置，再对映射特征区域进行网格划分（水平与垂直方向网格数相等），然后每个网格进行最大池化操作，从而实现不同尺度的候选框固定到相同大小。池化结果将作为后续卷积网络与全连接层的输入，用于目标分类预测以及位置回归。

## （2）损失函数

Faster-RCNN 算法的损失函数如式 5.2 所示：

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} p_i^* \sum_i L_{reg}(t_i, t_i^*) \quad (5.2)$$

在式 5.2 中  $N_{cls}$  表示均衡后的用分类回归计算的候选框数目； $N_{reg}$  表示锚框个数； $i$  代表锚框索引，若锚框为正样本则  $p_i^* = 1$ ，负样本  $p_i^* = 0$ ； $p_i$  是锚框对应预测框的前景背景二分类结果；由于  $N_{cls}$  和  $N_{reg}$  在数值上存在较大差距，因此设置  $\lambda$  为平衡权重值； $t_i$  表示预测框位置向量； $t_i^*$  是锚框对应的目标真实框位置向量，表示为  $t_i^* = [t_x^*, t_y^*, t_w^*, t_h^*]$ ，各个分量由如下计算公式所得。

$$t_x^* = \frac{x^* - x_a}{w_a} \quad (5.3)$$

$$t_y^* = \frac{y^* - y_a}{h_a} \quad (5.4)$$

$$t_w^* = \log\left(\frac{w^*}{w_a}\right) \quad (5.5)$$

$$t_h^* = \log\left(\frac{h^*}{h_a}\right) \quad (5.6)$$

其中  $x_a, y_a, w_a, h_a$  分别代表锚框中心点横纵坐标以及宽高;  $x^*, y^*, w^*, h^*$  表示目标真实框的中心点横纵坐标以及宽高。

分类损失函数  $L_{cls}$  计算如下:

$$L_{cls}(p_i, p_i^*) = -\log(p_i^* p_i + (1 - p_i^*)(1 - p_i)) \quad (5.7)$$

位置回归损失函数  $L_{reg}$  计算如下:

$$L_{reg}(t_i, t_i^*) = \sum_{i \in \{x, y, w, h\}} smooth_{L1}(t_i - t_i^*) \quad (5.8)$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & \text{Otherwise} \end{cases} \quad (5.9)$$

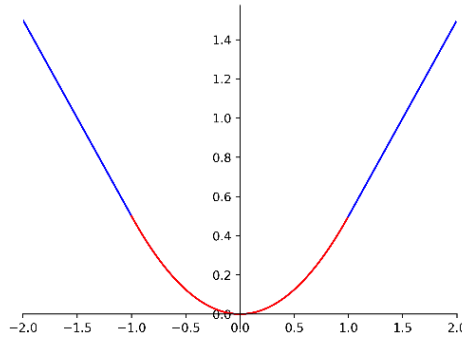


图 5.4 Smooth\_L1 函数曲线图

Fig. 5.4 The function graph of Smooth\_L1

Faster-RCNN 相较于 R-CNN<sup>[48]</sup>以及 Fast-RCNN<sup>[51]</sup>算法在检测速度以及精度上有了较好的提升。但由于算法采用先生成候选区域、后对每个区域进行分类回归的计算流程,以及 VGG 和全连接神经网络的设计结构,导致算法参数存储与计算消耗较大,无法有效应用于实时检测。后续学者提出 SSD、YOLO<sup>[54-58]</sup>等基于回归的 One-Stage 检测算法,让目标检测在保证高精度的同时,速度大幅提升。

### 5.1.2 SSD 算法

#### (1) 算法结构

SSD 算法是基于回归的目标检测算法代表之一,网络结构如图 5.5 所示。

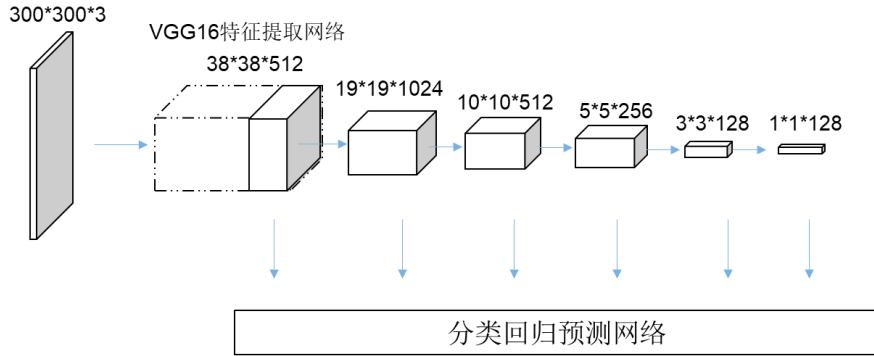


图 5.5 SSD 算法网络结构

Fig. 5.5 Schematic diagram of SSD

SSD 融合了 Faster-RCNN 与 YOLO 算法中先验框设置机制，并通过将不同卷积层提取的特征信息进行融合，有效提高网络检测精度也提升模型推理速度。原始 SSD 算法输入数据大小为  $300 \times 300 \times 3$ ，特征提取网络采用了 VGG16 前 4 个卷积层以及新增添的 6 个卷积层，同时摒弃 VGG 末端全连接神经网络，将其替换为两个卷积块。网络输出  $38 \times 38, 19 \times 19, 10 \times 10, 5 \times 5, 3 \times 3, 1 \times 1$  这 6 种不同分辨率大小特征图，浅层网络提取到的小尺度特征图用于检测小型目标，深层网络提取的大尺度特征图用于检测大的目标。

SSD 对 6 个尺度特征层预先设定了先验框个数依次为 4,6,6,6,4,4。每一层预设的先验框大小符合线性增长原则，即随着特征层尺寸减小，先验框的尺寸增大。先验框相对于原始图片比例计算方法如式 5.10 所示。

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m-1} (k-1), k \in [1, m] \quad (5.10)$$

其中  $m$  为特征图个数取值为 5。由于第一个特征层( $k=0$ )对应先验框事先以确定为 30，因此  $k$  取 1 至 5 正整数；文献[69]中  $s_{max}$  与  $s_{min}$  设定的为 0.9 与 0.2，经过计算得到先验框标准大小  $S_k \in \{30, 60, 111, 162, 213, 264, 315\}$ ，其余先验框的长宽比  $a_r \in \{1, 2, 3, 1/2, 1/3\}$ 。先验框宽高计算方式为：

$$w_k^a = S_k \sqrt{a_r} \quad (5.11)$$

$$h_k^a = S_k / \sqrt{a_r} \quad (5.12)$$

默认情况下每个特征图至少设置两个长宽比为 1 但尺度为  $S_k$  和  $S_k' = \sqrt{S_k \times S_{k+1}}$  的先验框。当特征层数目为 4 则  $a_r = \{1, 2, 1/2\}$ ，若特征层数目为 6 则  $a_r = \{1, 2, 1/2, 3, 1/3\}$ 。6 个特征层总计产生 8732 个先验框。SSD 网络输出结果用于先验框位置微调，再将校正后的先验框同目标真实框计算 IoU 值，最后通过非极大抑制算法获得最优分类结果。

## (2) 损失函数

SSD 的损失函数由两部分组成：一部分是目标框位置损失函数  $L_{reg}$ ，另一部分是多分类交叉熵损失函数  $L_{cls}$ 。计算式如下所示：

$$Loss = \frac{1}{N} (L_{reg} + \alpha L_{cls}) \quad (5.13)$$

其中  $N$  为先验框总数，由于网络训练过程中正负样本的分布不平衡，容易导致损失函数计算差异过大，因此正样本与负样本数量上保持 1:3 进行选取； $\alpha$  为平衡权重； $L_{reg}$  与  $L_{cls}$  计算如下所示：

$$L_{cls} = - \sum_{i \in Pos} I_{ij}^p \log(c_i^p) - \sum_{i \in Neg} \log(c_i^n) \quad (5.14)$$

$$c_i^p = \frac{\exp(c_i^p)}{\sum_{Pos} \exp(c_i^p)}, c_i^n = \frac{\exp(c_i^n)}{\sum_{Neg} \exp(c_i^n)} \quad (5.15)$$

$$L_{reg} = \sum_{i \in Pos} \sum_{m \in [x, y, w, h]} I_{ij}^p smooth_{L1}(l_i^m - g_j^m) \quad (5.16)$$

其中当第  $i$  个预测框与第  $j$  个目标真实框类别一致  $I_{ij}^p$  取 1，反之为 0； $c_i^p$  表示第  $i$  个预测框为前景框置信度； $c_i^n$  表示第  $i$  个预测框为背景框置信度； $l_i^m$  表示第  $i$  个预测框坐标向量值； $g_j^m$  表示第  $j$  个目标真实框坐标向量值。

### 5.1.3 YOLOV4-tiny 算法

#### (1) 算法结构

YOLOV4 是 YOLO 系列代表算法之一。YOLOV4-tiny 是 YOLOV4 网络结构做简化的结果，该算法在 COCO 目标检测数据集达到 40.2% 的 mAP。网络结构如下图 5.6 所示，图中括号数字组合表示输出特征图分辨率与通道数。

用于特征信息提取的主干网络是 CSPDarkNet53-tiny，该网络在 DarkNet53 的每一个残差模块加入 CSPNet<sup>[76]</sup>模块。CSPNet 是一种加强卷积神经网络学习能力的网络结构，它在保持算法准确率的同时也实现网络的轻量化，有效降低计算内存的消耗。网络结构是将输入的特征图进行通道拆分处理（split）分成两个部分，一部分用于卷积等操作，剩余的部分直接与上一部分处理的结果进行通道拼接。在文献中也验证了网络在目标检测时对主干网络性能的提升非常明显。

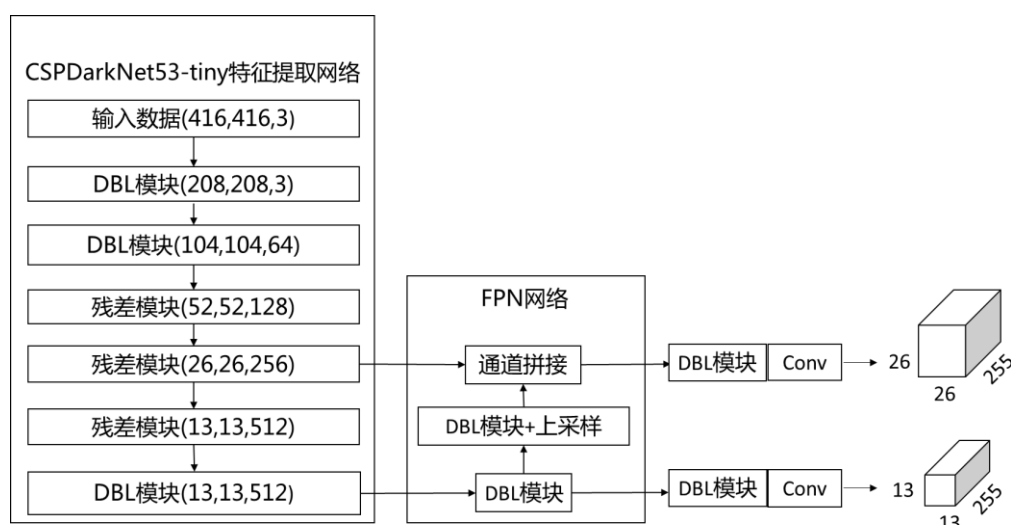


图 5.6 YOLOV4-tiny 网络结构示意图  
Fig. 5.6 Schematic diagram of YOLOV4-tiny

CSPDarkNet53-tiny 特征网络由 DBL(DarknetConv2D\_BN\_LeakyRelu)模块和残差模块(Resblock\_body)共同组成。前者计算流程如图 5.7 左侧所示，后者卷积计算流程如图 5.7 右侧所示。

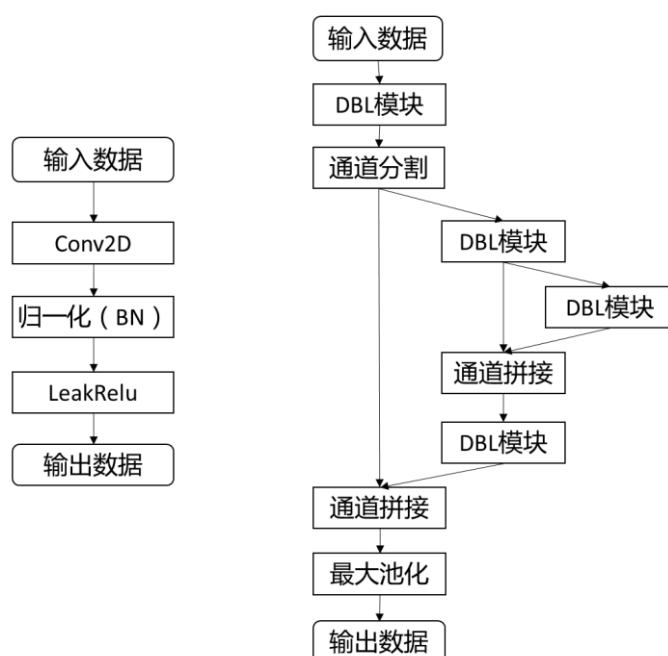


图 5.7 DBL 模块（左）与残差块（右）计算流程图  
Fig. 5.7 Calculation flow chart of DBL module (left) and residual block (right)

YOLOV4-tiny 采用特征金字塔结构 (FPN)<sup>[77]</sup>用于处理多尺度变化问题。FPN 将主干网络最后一层提取的  $13 \times 13$  特征图进行一次卷积后上采样转换为  $26 \times 26$ ，实现不同尺度特征层直接堆叠形成全新特征层。YOLO Head 接收两种尺度的特征信息再输入到卷积神经网络输出最终目标分类与位置回归。原始图片单个通道对应输出数据包含了预测框的位置信息、置信度以及目标分类结果，如图 5.8 所示。

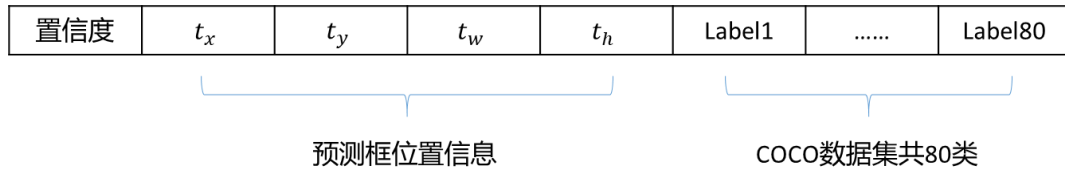


图 5.8 输出数据结构

Fig. 5.8 Structure of output data

其中位置信息由预测框中心点横纵坐标相对偏移量以及长宽构成，在实际预测时对原图需要按照 YOLOHead 输出特征尺度大小进行网格划分，每个网格坐标结合预测位置信息便得到预测框实际位置。校正后的预测框按照类别执行非极大抑制算法，筛选得到最终预测结果。

## (2) 损失函数

YOLOV4-tiny 损失函数包含三个部分：

- ① 边框位置回归损失函数  $L_{CioU}$ 。
- ② 预测框前景、背景置信度损失函数  $L_{conf}$ 。
- ③ 目标分类交叉熵损失函数  $L_{cls}$ 。

整体损失函数计算如式 5.17 所示：

$$Loss = L_{CioU} + L_{conf} + L_{cls} \quad (5.17)$$

$L_{CioU}$  在 IoU 的基础上，进一步融合了边框间的重合度、中心点距离以及边框宽高比因素。计算如式 5.18 所示：

$$L_{CioU} = \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} [1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + av] \quad (5.18)$$

其中  $K \times K$  为 YOLO Head 输出特征图尺度，特征图上每个点对应生成  $M$  个预测框，共计产生  $K \times K \times M$  个。 $I_{ij}^{obj}$ 、 $I_{ij}^{noobj}$  代表第  $i$  个网格的第  $j$  个预测框中存在物体分别置 1 和 0，不存在目标则置 0 和 1。 $\rho^2(b, b^{gt})$  代表预测框与真实框的关于中心点之间的欧氏距离， $c$  为同时涵盖目标框和预测框的最小矩形框对角距离，如图 5.9 所示。



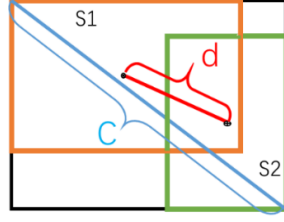


图 5.9 CIoU 结构图

Fig. 5.9 Schematic diagram of CIoU

在式 5.18 中对  $a, v$  计算如式 5.19 与 5.20 所示：

$$a = \frac{v}{1 - IoU + v} \quad (5.19)$$

$$v = \frac{4}{\pi^2} \left( \arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (5.20)$$

$w^{gt} / h^{gt}$  表示目标框的宽高比， $w / h$  表示预测框宽高比。对于  $L_{conf}$  以及  $L_{cls}$  计算如式 5.21 与 5.22 示：

$$L_{conf} = - \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{obj} [C_i^j \log(C_i^j) + (1 - C_i^j) \log(1 - C_i^j)] - \lambda_{noobj} \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{noobj} [C_i^j \log(C_i^j) + (1 - C_i^j) \log(1 - C_i^j)] \quad (5.21)$$

$$L_{cls} = - \sum_{i=0}^{K \times K} I_{ij}^{obj} \sum_{c \in classes} [p_i^j(c) \log(p_i^j(c)) + (1 - p_i^j(c)) \log(1 - p_i^j(c))] \quad (5.22)$$

其中  $\lambda_{noobj}$  为平衡权重，减少不涵盖目标的预测结果影响力。 $C_i^j$ 、 $C_i^j$  代表目标框与预测框置信度； $p_i^j(c)$ 、 $p_i^j(c)$  表示目标框与预测框类别为  $c$  概率值。

## 5.2 实验结果与分析

### 5.2.1 数据集

实验数据集采用 PASCAL VOC2007 数据集以及 VOC 2012 数据集。由于煤层气井站现常采集异常信号多由人以及车辆组成，因此合并 VOC2007+2012 数据集并从中提取出人与车辆共计 5 类分类目标用于训练测试。合并后的训练集以及测试集组成如表 5.1 所示，其中单张图片可能包含有多种类别，所以 5 类总数与 Total 会存在差异。

表 5.1 训练集与测试集组成结构

Tab. 5.1 Composition structure of training set and test set

类别	训练集/验证集	测试集
Person	6095	2007
Car	1874	721
Bicycle	795	239
Motorbike	771	222
Bus	607	174
Total	8111	2734

### 5.2.2 评价指标

#### (1) 模型参数量

计算详细描述见 4.2.3 小节。

#### (2) 平均精度均值

对于二分类问题存在的正负两类样本，规定以下四个指标描述：（1）**TP**，即被正确预测的正类样本总数；（2）**FP**，即被预测为正样本的负类样本总数；（3）**TN**，即被正确预测为负类样本的负样本总数；（4）**FN**，即被预测成负样本的正类样本总数。因此，准确率（**Precision**）定义为被正确预测的正类样本数目占总样本数目比例，如式 5.23 所示；召回率（**Recall**）定义为被正确预测的正样本数目占正样本总数的比例，如式 5.24 所示。

$$Precision = \frac{TP}{TP + FP} \quad (5.23)$$

$$Recall = \frac{TP}{TP + FN} \quad (5.24)$$

以 **Recall** 作为横坐标、**Precision** 作为纵坐标构建坐标系得到的曲线图即为 **PR** 曲线图，该曲线与坐标轴围成的面积为平均精度（**Average Precision, AP**）。在多分类应用场景下，对每一个类别求得 **AP** 后再取均值即得到平均精度均值(**mean Average Precision, mAP**)。

#### (3) 检测时长

模型运行速度是一项重要的评价指标，关系到算法能否运用到嵌入式平台投入到实际应用场景。检测时长为单张图片检测时间，检测时长越小代表算法运行速度越快。

### 5.2.3 实验内容与结果分析

实验内容如下：

(1) 将论文设计的轻量化神经网络 MMGNet 用于 Faster-RCNN、SSD 以及 YOLO V4-tiny 主干特征提取网络

(2) 调整 Faster-RCNN 末端分类回归的网络结构, 将全连接神经网络结构用卷积代替, 进一步减少模型参数量以及计算量; RPN 网络生成的候选框存在正负样本不均, 因此训练过程中对候选框进行 IoU 排序后, 从中选择 256 个候选框 (正负样本各 128) 用于后续网络训练, 保证训练正负样本均衡。

(3) 改进 SSD 算法选择输出分辨率大小为 19,10,5,3,2,1 的六个特征层用于目标分类与位置回归, 有效降低模型参数量与计算量。

(4) 实验中为了增加数据集的丰富性, 防止参数的过拟合, 在训练过程中采取数据增强操作, 主要包含缩放, 翻转以及修改图片饱和度等处理。

(5) 实验训练环境同表 4.3 所示, 优化器均采用 Adam 优化器, 批次训练样本数为 8 (受训练设备显存影响, 防止溢出), 初期为加快训练速度对网络模型冻结训练; 初始学习率为  $1 \times 10^{-3}$ , 同时为防止在迭代一定次数后, 损失函数变化稳定导致参数无法有效更新, 人为在一定步骤后对学习率减半, 衰减为  $1 \times 10^{-5}$  后保持不变。

MMGNet 融合三种目标检测算法在 GTX 1050ti 实验结果如下表所示:

表 5.2 3 种目标检测算法及改进算法在 GTX 1050ti 实验结果

Tab. 5.2 Experimental results of three target detection algorithms and improved algorithms in GTX 1050ti

模型名称	参数量	平均精度均值	检测时长
Model Name	Params (MB)	mAP (%)	Time (s)
Faster-RCNN	108.9	69.8	0.88
Faster-RCNN(new)	42.4	66.4	0.32
SSD	100.3	74.0	0.05
SSD(new)	26.5	71.0	0.034
YOLOV4_tiny	22.5	79.0	0.021
YOLOV4_tiny(new)	<b>5.9</b>	<b>74.8</b>	<b>0.023</b>

从表 5.2 中测试结果可知 MMGNet 作为 Faster-RCNN 目标检测算法的特征提取网络, 取得了较好的实验效果。算法在保持一定 mAP 的基础上, 参数量缩减近 61%, 检测速度提高 48%; MMGNet 结合 SSD 在模型参数量上缩减 73.6%, 虽然在 mAP 下降幅度为 3%, 单张数据检测时长降低至 0.034s, 证明模型参数减少同时保持了较好的运行速率。MMGNet 结合 YOLO V4-tiny 算法在参数量减少了 74%, 但是 mAP 减少 4.2% 且检测时长有所增加, 说明设计网络对于 YOLOV4-tiny 算法改进空间较小。

综上所述，论文设计的面向边缘计算设备的卷积神经网络结合经典目标检测算法，都能够实现对原有算法模型在参数量的优化，并还能保持一定的分类准确率。在模型运行速率上，设计网络对 Faster-RCNN 与 SSD 有较好提升，对 YOLOV4-tiny 效果提升效果较小。

对比改进后的三种算法模型，基于 YOLO V4-Tiny 的网络参数量最小，仅为 5.9MB，mAP 最高达到 74.8%，单张数据 0.023s 的检测时长虽然高于 YOLOV4-tiny 的 0.021s，但也满足实际需要。综合考虑三种指标后，选择基于 MMGNet 的 YOLO V4-Tiny 算法运用到本文基于边缘计算的视频监控系统设计方案中，通过对实际的煤气气井站异常信号进行检测，检验改进算法的实用性。

### 5.3 本章小结

本章节首先介绍了一种 Two-stage 目标检测算法 Faster-RCNN 以及两种 One-stage 目标检测算法 SSD 与 YOLO V4-tiny，通过阐述三种算法原理，为后续改进实验做好理论铺垫。针对 3 种算法特征提取网络存在参数量较大缺陷，提出将 MMGNet 深度卷积神经网络模型与目标检测算法结合。通过在公开数据集 Pascal VOC2007 以及 VOC2012 对比实验，验证了本文设计的 MMGNet 深度神经网络结合目标检测算法具有可行性。最后对比三种改进算法各项指标结果，选择改进 YOLO V4-tiny 目标检测算法融入到设计的视频监控系统架构方案，在下一章节中实际运用到煤层气井站异常信号检测，验证本文设计方案的可行性。

## 6 煤层气井站异常目标检测

### 6.1 应用场景介绍

煤层气是一种存储在煤层当中的可燃烧清洁气体，是我国重要的化石能源。煤层气采集作业现场为煤气气井站，多分布于偏远的山区且现场多为无人值守，因此，为保证煤层气安全开采，需要对煤层气井站实现视频监控。对煤层气井站安全防护多采用基于云计算的人工监控方法，该方案较依赖监控人员主观判断，容易因人为疏忽导致异常目标错检漏检。同时煤层气井站监控系统涉及的摄像头数以千计，产生的数据流量较大，对网络传输负载以及云中心存储计算会产生巨额消耗。因此，采用基于边缘计算的视频监控对煤层气井站异常目标实现检测，具有较高的实用价值。而本文设计的面向边缘计算的视频监控系统，通过边缘计算设备的板载摄像头采集数据，同时运行目标检测算法实现异常目标高效捕捉并同时给出异常决策响应，有效提高系统响应速度，缓解网络负载压力以及人为因素对监控质量干扰，系统检测原理如图 6.1 所示。本章节使用某煤层气井站监控视频数据作为测试数据，通过在论文设计的视频监控系统上运行目标检测算法，对该测试数据进行检验来验证方案的可行性。

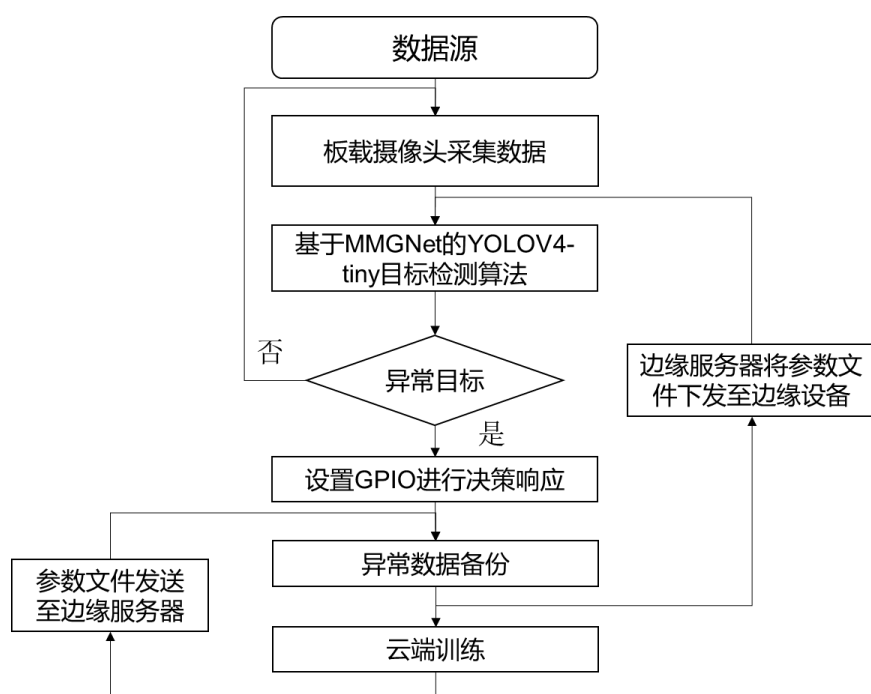


图 6.1 视频监控系统检测流程图

Fig. 6.1 Detection flow chart of video surveillance system

6.2 数据集

实验数据来自某煤层气井站视频监控数据集，视频数据信息如表 6.1 所示。

表 6.1 煤层气井站视频数据集  
Tab. 6.1 video data set of coalbed methane well station

视频名称	分辨率	帧率(帧/s)	时长(s)
Vedio1	640*480	24	54
Vedio2	1280*720	30	241
Vedio3	1280*720	30	307
Vedio4	1280*720	30	386

实验前须构建目标检测数据集。首先对视频数据集进行抽帧读取图片数据，再由目标检测标注工具 labelImg（如图 6.2 所示）进行人工标注，最后将检测目标框出后保存为 Pascal VOC 数据集格式，由 labelImg 自动生成图片对应的 xml 文件，文件包括目标框类别、位置等数据信息。标注后的目标检测数据集如表 6.2 所示。

表 6.2 煤层气井站目标检测数据集  
Tab. 6.2 Target detection data set of coalbed methane well station

类别	数量(张)
Person	637
Car	335

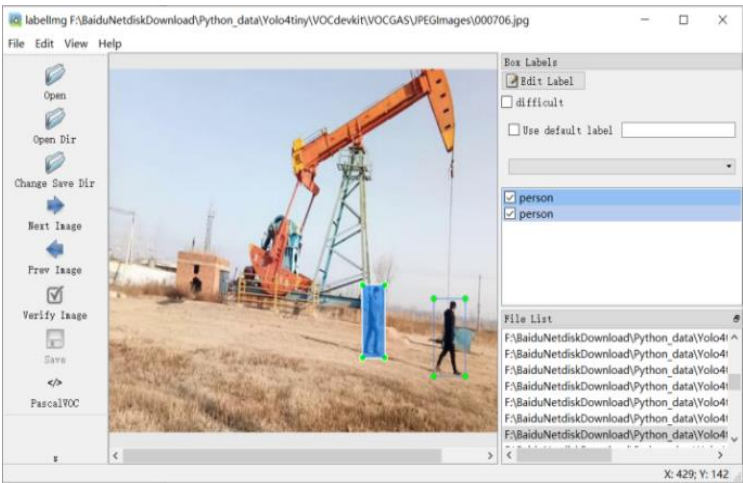


图 6.2 labelImg 运行截图  
Fig. 6.2 The running screenshot of labelImg

### 6.3 评价指标

本章节实验所需的评价指标包括平均精度 AP、平均精度均值 mAP 以及检测时长 Time。指标描述及计算形式见第五章 5.2.2 小节。

### 6.4 实验结果分析

针对车、行人两类预测结果绘制 PR 曲线如图 6.3 所示。分别计算两类 AP 并求均值得到 mAP，再将算法载入到 NVIDIA TX2 平台得到实际单张图片检测时长，实验结果如表 6.3 所示。

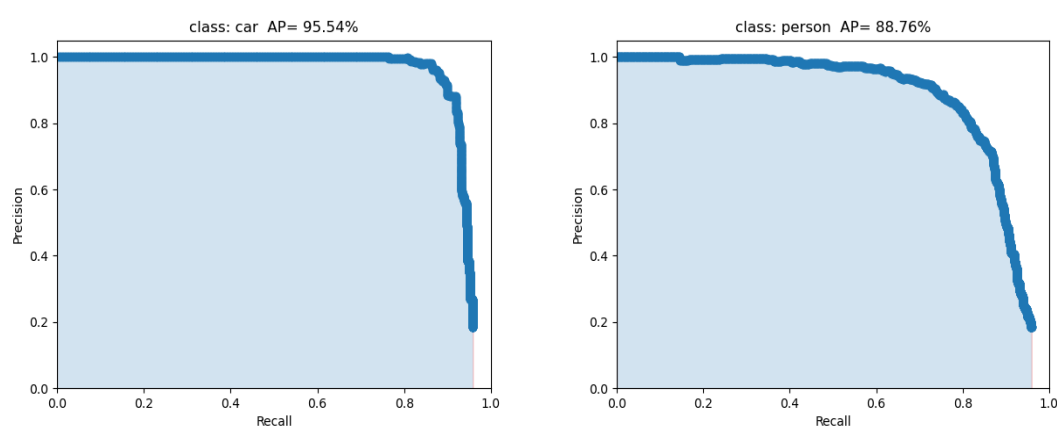


图 6.3 对车辆、行人检测的 PR 曲线图

Fig. 6.3 PR graph of vehicle and pedestrian detection

表 6.3 煤层气井站测试集实验结果

Tab. 6.3 Experimental results of coalbed methane well station test set

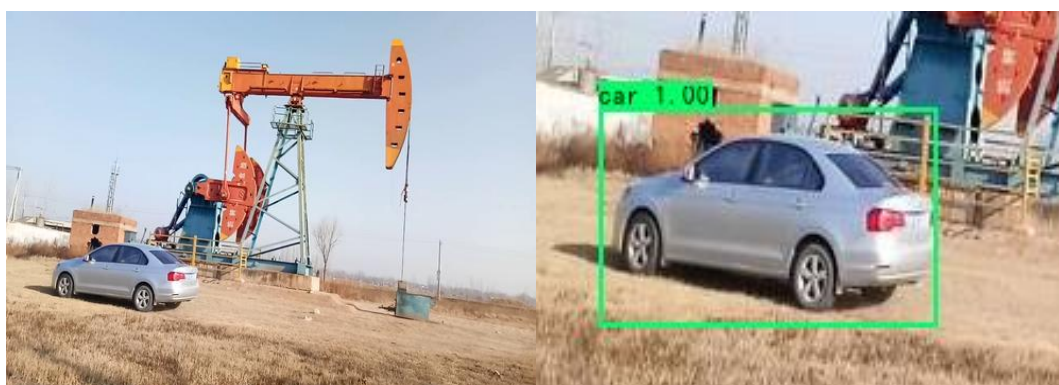
模型 Model Name	平均精度 AP (%)		平均精度均值 mAP (%)	检测时长 Time (s)
YOLOV4-tiny(new)	Car	Person	92.15	0.102
	95.54	88.76		

从图 6.3 与表 6.3 测试结果中可以看出,基于 MMGNet 的 YOLOV4-tiny 算法对车、行人两类目标均有着较好的召回率以及精确率,两类平均精度均值 92.15%证明算法有较强的预测精度,在 TX2 上运行时单张图片检测时长为 0.102 s,有效证明设计算法符合实际需要。部分检测结果放大后如图 6.4 所示。



(a) 车辆检测效果图 1

(a) Car detection renderings 1



(b) 车辆检测效果图 2

(b) Car detection renderings 2



(c) 行人检测效果图 1

(c) Person detection renderings 1





(d) 行人检测效果图 2

(d) Person detection renderings 2

图 6.4 煤层气井站异常检测部分效果图

Fig. 6.4 Partial effect diagram of anomaly detection in coalbed methane well station

## 6.5 本章小结

本章节将基于 MMGNet 的 YOLOV4-tiny 目标检测算法移植到边缘计算设备 NVIDIA TX2。对煤层气井站现场采集的视频数据检测，mAP 达到 92.15%，单张图片数据检测时长为 0.102s，证明目标检测算法具备高检测精度以及低运行速度，满足本文监控系统架构方案需要。

## 结 论

传统基于云计算范式的人工视频监控方法，在对低概率出现的异常目标监测时，容易产生大量冗余视频数据，也会由于数据传输延时以及人为因素导致监控质量降低。边缘计算范式通过在网络边缘端实时采集、分析以及处理数据，有效降低系统响应延时，缓解云端负载压力，减少数据在存储与传输中的资源消耗。因此探索基于边缘计算的视频监控系統架构方案具有重要研究意义与应用价值。与此同时，对异常目标的检测采用深度学习技术可有效排除人为因素导致的监控质量下降问题，降低人力资源消耗，提高系统决策响应实时性。本文围绕基于边缘计算的视频监控系統架构开展研究，运用深度卷积神经网络、目标检测算法等深度学习技术实现异常目标实时检测，在煤层气井站应用场景下取得了较好的验证结果。

本文研究内容具体如下：

1. 设计一种面向边缘计算的视频监控系統架构方案，解决传统基于云计算范式的人工视频监控系統存在云端负载压力大、系統延时以及人力资源消耗等问题。该方案采取边缘计算设备、边缘服务器以及云端协同工作的架构方案，通过将云端前向推理计算卸载到边缘计算设备实现对采集数据的实时分析处理，降低因视频数据传输导致系統响应延时，同时也缓解人为因素导致监控质量降低；将视频数据以及检测算法参数文件放在边缘服务器有利于缓解云端数据存储资源的消耗；利用云端强大算力实现模型参数训练并定时向边缘服务器传输参数文件。通过对煤层气井站监控数据进行检验，证明设计方案具备可行性。

2. 提出一种基于多尺度卷积核的轻量化深度卷积神经网络，能有效应用于边缘嵌入式设备。传统深度卷积网络由于模型参数量较大、计算资源占比高，在较小存储空间以及运行内存的嵌入式设备难以实时运行。本文基于 MobileNet 系列网络以及 GhostNet 提出全新架构的 MMGNet，对比传统卷积网络实现参数量和内存占比有效降低，网络检测精度也维持在较好水准。

3. 利用基于 MMGNet 的目标检测算法，实现异常目标实时准确监测。针对传统三种目标检测算法在特征提取网络存在参数量较大缺陷，使用论文所设计轻量化深度卷积网络作为特征提取网络，有效降低算法模型参数量，并保持较好的检测精度，运行速度满足实际需要。将算法运行于边缘计算设备，在煤层气井站视频数据集上对车辆、行人异常检测的分类平均精度均值达到 92.15%，单张图片检测时长为 0.102s，证明改进目标检测算法可实现异常信号检测并有效融入到本文设计的视频监控系統架构方案。

本文设计算法能够取得较好的测试效果，首先得益于深度卷积网络采用多尺度卷积核能够提取到更多的特征信息，并通过通道注意力机制选取出较好的特征图，有效提高网络预测精度；其次将多尺度卷积结果直接通道拼接的结构思路维持了“先扩张”的结构策略，并且更加丰富了高维空间蕴含的特征信息；最后通过 GhostNet 结构实现“后压缩”思路且进一步实现网络轻量化。虽然本文设计的基于边缘计算的深度卷积网络在算法性能上符合实际需要，但仍存在提升改进空间：

1. 本文设计的深度卷积网络在轻量化表现上，相比于 SqueezeNet、MobileNet 等著名轻量化卷积网络存在差距。同时该网络对 YOLOV4-tiny 算法提升空间有限，检测速度存在差距。这是由于模型采用多尺度卷积一定程度增加了计算量。在未来研究中可以采取模型剪枝、量化等压缩策略探讨网络进一步轻量化实现,使用 TensorRT 技术对网络加速或是设计更加轻量化的深度卷积网络。

2. 在对煤层气井站视频数据进行实时检测时，本文采用直接对视频数据进行抽帧读取图片，该方法容易出现由于前后帧图片差距较小导致的冗余检测，未来可研究针对视频数据流处理算法进一步提高系统检测速度。

3. 本文监控异常目标涉及行人车辆，异常目标总类以及监测环境复杂度较低，为提高监控系统泛化性，可探索将更多不同种类的异常目标以及应用环境纳入检测范围。

## 参 考 文 献

- [1] 姜栋瀚, 刘晓平, 吴作栋. 云计算技术发展现状研究综述[J]. 信息与脑, 2019, (8):170-171.
- [2] 赵明. 边缘计算技术及应用综述[J]. 计算机科学, 2020, 47(z1):268-272, 282. DOI:10.11896/jsj.kx.190600115.
- [3] 葛畅, 白光伟, 沈航, 等. 基于边缘计算的视频监控框架[J]. 计算机工程与设计, 2019, 40(1):32-39. DOI:10.16208/j.issn1000-7024.2019.01.006.
- [4] 余京蕾. 浅谈计算机视觉技术进展及其新兴应用[J]. 北京联合大学学报(自然版), 2020, 34(1):63-69. DOI:10.16255/j.cnki.lidxbz.2020.01.010.
- [5] Voulodimos A, Doulamis N, Doulamis A, et al. Deep Learning for Computer Vision: A Brief Review[J]. Computational Intelligence and Neuroscience, 2018, 2018:1-13.
- [6] 李子姝, 谢人超, 孙礼, 等. 移动边缘计算综述[J]. 电信科学, 2018, 34(001):87-101.
- [7] 计春雷, 杨志和, 谢致邦. 服务计算新模式:雾计算[J]. 上海电机学院学报, 2012, 015(005):337-341.
- [8] Laghari A A, Jumani A K, Laghari R A. Review and State of Art of Fog Computing[J]. Archives of Computational Methods in Engineering, 2021(5).
- [9] 孙利民, 沈杰, 朱红松. 从云计算到海计算:论物联网的体系结构[J]. 中兴通讯技术, 2011(01):8-12.
- [10] Shi W, Sun H, Cao J, et al. Edge Computing: An Emerging Computing Model for the Internet of Everything Era[J]. Journal of Computer Research and Development, 2017, 54(5):907-924.
- [11] Satyanarayanan M. The Emergence of Edge Computing[J]. Computer, 2017, 50(1):30-39.
- [12] Jie Lin, Wei Yu, Nan Zhang, et al. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications[J]. IEEE Internet of Things Journal, 2017, 735(4):1125-1142.
- [13] Hossain S K A, Rahman M A, Hossain M A. Edge computing framework for enabling situation awareness in IoT based smart city[J]. Journal of Parallel and Distributed Computing, 2018, 122(DEC.):226-237.
- [14] 杨微星. 5G 通信技术应用场景与关键技术分析[J]. 科学与信息化, 2021(1):60.
- [15] Sun X, Ansari N. EdgeIoT: Mobile Edge Computing for the Internet of Things[J]. IEEE Communications Magazine, 2016, 54(12):22-29.
- [16] 蔡锴, 蔡争耘, 李瑜, 等. 边缘计算在智慧制造领域的应用[J]. 电信科学, 2019(2).
- [17] Wang R, Tsai W T, He J, et al. A Video Surveillance System Based on Permissioned Blockchains and Edge Computing. IEEE, 2019.

- [18]Chen J, Li K , Deng Q , et al. Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing[J].IEEE Transactions on Industrial Informatics, 2019:1-1.
- [19]元慧,穆晓芳,韩素青,等.一种基于边缘计算的仓库视频监控系统:CN110913181A[P].2020.
- [20]葛畅,白光伟,沈航,等.基于边缘计算的视频监控框架[J].计算机工程与设计,2019,040(001):32-39.
- [21]Lecun Y, Bottou L. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.
- [22]Krizhevsky, Alex, Sutskever, et al. ImageNet Classification with Deep Convolutional Neural Networks[J]. Communications of the ACM, 2017,60(6):84-90.
- [23]Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
- [24]Szegedy C, Wei L, Jia Y, et al. Going Deeper with Convolutions[J].IEEE Computer Society, 2014.
- [25]Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. JMLR.org, 2015.
- [26]Xia X, Cui X, Bing N. Inception-v3 for flower classification[C]//2017 2nd International Conference on Image, Vision and Computing (ICIVC). IEEE, 2017.
- [27]He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J].IEEE, 2016.
- [28]He K, Jian S. Convolutional neural networks at constrained time cost[C]//2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015.
- [29]Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning[J]. 2016.
- [30]Jie H, Li S, Gang S, et al.Squeeze-and-Excitation Networks[J].IEEE Transaction on Pattern Analysis and Machine Intelligence, 2017, PP(99).
- [31]Li X, Wang W, Hu X, et al. Selective Kernel Networks[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
- [32]Woo S , Park J , Lee J Y , et al. CBAM: Convolutional Block Attention Model[J]. Springer, Cham, 2018.
- [33]Iandola, Forrest N, Han, et al. SqueezeNet: AlexNet-level accuracy with 50x Fewer parameters and <0.5MB model size[J].2016.
- [34]Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017.
- [35]Sandler M, Howard A, Zhu M, et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks[J].2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

- [36]Howard A, Sandler M, Chen B, et al. Searching for MobileNetV3[C]// 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2020.
- [37]Zhang X, Zhou X, Lin M, et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices[J]. 2017.
- [38]Ma N, Zhang X, Zheng H T, et al. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design[J]. European Conference on Computer Vision, 2018.
- [39]Han K, Wang Y, Tian Q, et al. GhostNet: More Features From Cheap Operation[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR) IEEE, 2020.
- [40]Nixon M S. Feature Extraction and Image Processing[M]. Publishing House of Electronics Industry, 2013.
- [41]Xu Y, D Xu, Lin S, et al. Sliding Window and Regression Based Cup Detection In Digital Fundus Images for Glaucoma Diagnosis[J]. Springer-Verlag, 2011.
- [42]Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [43]刘方园, 王水花, 张煜东. 方向梯度直方图综述[J]. 计算机工程与应用, 2017, 53(19).
- [44]Lin, Zhu, Yang, et al. Large-scale image classification: Fast feature extraction and SVM training. IEEE Computer Society, 2011.
- [45]乔凤娟, 郭红利, 李伟, 等. 基于 SVM 的深度学习分类研究综述 [J]. 齐鲁工业大学学报, 2018, 32(5):39-44.
- [46]张溪樾. 基于 Adaboost 的行人检测综述[J]. 电子制作, 2019, (1):59-61.
- [47]黄健, 张钢. 深度卷积神经网络的目标检测算法综述[J]. 计算机工程与应用, 2020, 56(17):12-23. DOI:10.3778/j.issn.1002-8331.2005-0021.
- [48]Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[J]. IEEE Computer Society, 2013.
- [49]Uijlings, J.R.R., Van De Sande, et al. Selective search for object recognition[J]. International Journal of Computer Vision, 2013, 104(2):154-171.
- [50]He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9):1904-16.
- [51]Girshick R. Fast R-CNN[J]. Computer Science, 2015.
- [52]Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.
- [53]Dai J, Li Y, He K, et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks[J]. Advances in Neural Information Processing Systems, 2016.
- [54]Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// Computer Vision & Pattern Recognition. IEEE, 2016.

- [55]Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[J]. IEEE Conference on Computer Vision & Pattern Recognition, 2017:6517-6525.
- [56]Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. Computer Science, 2018, 4(1):1-6.
- [57]Bochkovskiy A, Wang C Y, Liao H. YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. 2020.
- [58]Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[J]. Springer Cham, 2016.
- [59]周俊, 沈华杰, 林中允, 等. 边缘计算隐私保护研究进展 [J]. 计算机研究与发展, 2020, 57(10):2027-2051. DOI:10.7544/issn1000-1239.2020.20200614.
- [60]Evans D. The Internet of Things: How the Next Evolution of the Internet is Changing Everything[OL]. [2021-4-4].  
[https://www.researchgate.net/publication/313420977\\_The\\_Internet\\_of\\_Things\\_How\\_the\\_Next\\_Evolution\\_of\\_the\\_Internet\\_is\\_Changing\\_Everything](https://www.researchgate.net/publication/313420977_The_Internet_of_Things_How_the_Next_Evolution_of_the_Internet_is_Changing_Everything).
- [61]Camhi J. Former Cisco CEO John Chambers predicts 500 billion connected devices by 2025[OL]. 2015.  
[http://www.businessinsider.com/former-cisco-ceo-500-billion-connected-devices-by-2025-2015-11?utm\\_source=tuicool&utm\\_medium=referral](http://www.businessinsider.com/former-cisco-ceo-500-billion-connected-devices-by-2025-2015-11?utm_source=tuicool&utm_medium=referral).
- [62]吴迪. 边缘计算赋能智慧城市:机遇与挑战[J]. 人民论坛·学术前沿, 2020, 000(009):18-25.
- [63]黄倩怡, 李志洋, 谢文涛, 等. 智能家居中的边缘计算[J]. 计算机研究与发展, 2020, 57(9):1800-1809. DOI:10.7544/issn1000-1239.2020.20200253.
- [64]Liu S, Liu L, Tang J, et al. Edge Computing for Autonomous Driving: Opportunities and Challenges[J]. Proceedings of the IEEE, 2019, PP(99):1-20.
- [65]宋纯贺, 曾鹏, 于海斌. 工业互联网智能制造边缘计算:现状与挑战[J]. 中兴通讯技术, 2019, 25(3):50-57. DOI:10.12142/ZTETJ.201903008.
- [66]齐健. NVIDIA Jetson TX2 平台:加速发展小型化人工智能终端[J]. 智能制造, 2017(5):20-21. DOI:10.3969/j.issn.1671-8186.2017.05.005.
- [67]Lawrence S, Giles C L, AC Tsoi, et al. Face Recognition: A Convolutional Neural Network Approach[J]. IEEE Transactions on Neural Networks, 1997, 8(1): 98-113.
- [68]Perera C, Hung J Y, Chen K W, et al. A Deep Learning Approach to Identify Blepharoptosis by Convolutional Neural Networks[J]. International Journal of Medical Informatics, 2021, 148(3):104402.
- [69]Castellano G, Castiello C, Mencar C, et al. Crowd Counting from Unmanned Aerial Vehicles with Fully-Convolutional Neural Networks[C]//2020 International Joint Conference on Neural Networks (IJCNN). 2020.
- [70]Elsken T, Metzen J H, F Hutter. Neural Architecture Search: A Survey[J]. arXiv, 2018.

- [71]Han S, Mao H, Dally W J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding[J]. Fiber, 2015, 56(4):3-7.
- [72]Song H, Liu X, Mao H, et al. Deep compression and EIE: Efficient inference engine on compressed deep neural network[C]//Hot Chips 28 Symposium. IEEE, 2017.
- [73]He Y, Lin J, Liu Z, et al. AMC: AutoML for Model Compression and Acceleration on Mobile Devices[J]. 2018.
- [74]Ming L, Hu X. Recurrent convolutional neural network for object recognition [C] // IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2015:3367-3375.
- [75]Liao Z , Carneiro G . On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units[C]// 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2016.
- [76]Wang C Y, Liao H, Wu Y H, et al. CSPNet: A New Backbone that can Enhance Learning Capability of CNN[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2020.
- [77]Lin T Y, Dollar P, Girshick R, et al. Feature Pyramid Networks for Object Detection [C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2017.



## 致 谢

光阴似箭，日月如梭，经过六个多月的不懈努力，基于边缘计算的深度卷积神经网络研究与应用论文顺利完成。此次论文设计在杨建华老师与卢伟老师耐心教导逐步完成，无论是在初期毕业论文选题与技术路线选择，还是中期实验过程中疑难问题解答以及后期论文撰写格式修改，都体现了严谨的科学态度以及精益求精科研工作作风。正是有了二位老师细心指导，我才能保质保量地完成此次硕士毕业论文撰写工作。通过此次的毕业论文设计，让我对边缘计算以及前沿深度学习技术有了一个全方位了解，锻炼本人在代码编写以及独立自主查阅相关邻域资料的能力，知道一个完整科研工作展开完成过程。虽然论文设计结果存在着一些不足之处，但是这样一个独立自主分析问题解决问题的过程才是最为重要的意义。完成毕业设计不仅仅是对理论知识的夯实以及实践能力有效提高，更是为自己树立了踏实肯干、勇于创新、实事求是的人生态度，相信这将会在未来的学习工作中扮演至关重要的角色，激励着我砥砺前行。

在大连理工大学的控制工程专业硕士研究生求学历程缓缓走向终点。在两年时间里，我不仅仅收获到丰富的控制工程领域的基础理论知识，更是收获了重要的师生情、同学情以及朋友情。再次诚挚感激杨建华老师以及卢伟老师对我在学习上的严格教导，在衣食住行上的嘘寒问暖，在未来职业规划上方向指点迷津以及在道德品质的严格要求，是他们为我树立了作为科研人员应该具备的踏实肯干、勇于创新、诚实守信等重要科研精神。其次感谢研究生辅导员与助管，本人作为班长在协助辅导员以及助管管理班级事务时，深深感受到他们工作的艰苦。无论是在疫情期间的统筹监管，还是毕业招聘时相关信息准确推送以及论文撰写疑难解答，都切实营造了舒适的科研学习生活环境，正因为有他们辛苦工作才能让我心无旁骛完成此次的毕业论文设计。最后感谢教研室的师兄师姐以及同窗对我在科研工作中所遇到问题的细心指点以及在未来职业规划过程中的诸多建议，帮助我对未来的人生有了清晰的规划；同时也感谢班级同学以及四位室友，正是有了他们的一路陪伴，让我感受到友情的珍贵，这注定成为我人生经历中的又一笔宝贵财富。

最后感谢大连理工大学为广大学子提供优质学习资源以及衣食住行条件，感谢两年中控制工程专业授课老师传道授业解惑。

## 大连理工大学学位论文版权使用授权书

本人完全了解学校有关学位论文知识产权的规定，在校攻读学位期间论文工作的知识产权属于大连理工大学，允许论文被查阅和借阅。学校有权保留论文并向国家有关部门或机构送交论文的复印件和电子版，可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印、或扫描等复制手段保存和汇编本学位论文。

学位论文题目：面向边缘计算的目标检测算法研究与应用

作者签名：张 钊 日期：2021 年 6 月 15 日

导师签名：杨建华 日期：2021 年 6 月 15 日