计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.20200986
59(4):882-893,2022

# 基于迭代稀疏训练的轻量化无人机目标检测算法

侯 鑫[1] 曲国远[2] 魏大洲[2] 张佳程[3]
[1](中国科学院计算技术研究所 北京 100190)
[2](中国航空无线电电子研究所 上海 200241)
[3](北京邮电大学信息与通信工程学院 北京 100876)
(houxin@ict.ac.cn)

# A Lightweight UAV Object Detection Algorithm Based on Iterative Sparse Training

Hou Xin[1], Qu Guoyuan[2], Wei Dazhou[2], and Zhang Jiacheng[3]
[1](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)
[2](*Chinese Aeronautical Radio Electronics Research Institute, Shanghai* 200241)
[3](*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing* 100876)

**Abstract** With the maturity of UAV (unmanned aerial vehicle) technology, vehicles equipped with cameras are widely used in various fields, such as security and surveillance, aerial photography and infrastructure inspection. It is important to automatically and efficiently analyze and understand the visual data collected from vehicles. The object detection algorithm based on deep convolutional neural network has made amazing achievements in many practical applications, but it is often accompanied by great resource consumption and memory occupation. Thus, it is challenging to run deep convolutional neural networks directly on embedded devices with limited computing power carried by vehicles, which leads to high latency. In order to meet these challenges, a novel pruning algorithm based on iterative sparse training is proposed to improve the computational effectiveness of the classic object detection network YOLOv3 (you only look once). At the same time, different data enhancement methods and related optimization means are combined to ensure that the precision error of the detector before and after compression is within an acceptable range. Experimental results indicate that the pruning scheme based on iterative sparse training proposed in this paper achieves a considerable compression rate of YOLOv3 within slightly decline in precision. The original YOLOv3 model contains 61.57 MB weights and requires 139.77GFLOPS(floating-point operations). With 98.72% weights and 90.03% FLOPS reduced, our model still maintains a decent accuracy, with only 2.0% *mAP*(mean average precision) loss, which provides support for real-time application of UAV object detection.

**Key words** YOLOv3; model compression; iterative sparse training; data enhancement; low precision loss

**收稿日期**:2020-11-30;**修回日期**:2021-04-14

**基金项目**:国家重点研发计划项目(2018YFC0809300,2107YFB0202105,2016YFB0200803,2017YFB0202302);国家自然科学基金项目(61972376);北京市自然科学基金项目(L182053)
This work was supported by the National Key Basic Research and Development Program of China (2018YFC0809300, 2107YFB0202105, 2016YFB0200803, 2017YFB0202302), the National Natural Science Foundation of China (61972376), and the Beijing Natural Science Foundation (L182053).

**摘　要**　随着无人机技术的成熟,配备摄像头的无人机被广泛应用于各个领域,自动高效地分析和理解从无人机收集的视觉数据非常重要.基于深度卷积神经网络的目标检测算法在许多实际应用上取得了惊人的成绩,但往往伴随着巨大的资源消耗和内存占用.因此,对于无人机上携带的计算能力受限的嵌入式设备来说,直接运行深度卷积神经网络非常具有挑战性.为了应对这些挑战,以经典的目标检测方法 YOLOv3(you only look once)为例,基于迭代稀疏训练的剪枝方式可以实现有效的模型压缩,同时通过组合不同数据增强方式与相关优化手段保证压缩前后检测器精度误差在可接受范围内.实验结果证明,基于迭代稀疏训练的剪枝方法在 YOLOv3 上取得了非常可观的压缩效果,并且将精度误差控制在了 2%以内,为无人机目标检测实时应用提供了支持.

**关键词**　YOLOv3 算法;模型压缩;迭代稀疏训练;数据增强;精度误差小

**中图法分类号**　TP391

随着无人机(unmanned aerial vehicle, UAV)技术的成熟,配备摄像头和嵌入式系统的无人机已被广泛应用于各个领域,包括农业、安防与监视[1]、航空摄影[2]和基础设施检查[3]等.这些应用要求无人机平台能够感知环境,理解分析场景并作出相应的反应,其中最基础的功能就是目标自动高效检测.基于深度网络的目标检测器[4-8]通过卷积神经网络自动提取图像特征,大大提高了目标检测的性能.按照有没有利用候选区域,目标检测主要分为两大阵营:基于候选区域检测阵营与不基于候选区域检测阵营(如 CornerNet[9],CenterNet[10]等).基于候选区域检测阵营根据候选框处理手段又可以分为 2 阶段目标检测(如 R-CNN(region convolutional neural network)[11-13],R-FCN(region-based fully convolutional network)[14]等)和 1 阶段目标检测(如 YOLO(you only look once)[15-17],SSD(single shot multibox detector)[18],RetinaNet[19]等).

检测性能提高的同时也带来了巨大的资源消耗和内存占用,这对计算能力受限的无人机处理平台来说是不友好的:1)模型大小.深度卷积神经网络强大的特征表达能力得益于数百万个可训练的参数,如 VGG-16 网络,参数数量 1.3 亿多,需要占用 500 MB 空间,这对嵌入式设备来说是很大的资源负担.2)计算量.深度卷积神经网络前向推理的过程需要执行大量的浮点运算,VGG-16(visual geometry group)网络完成一次图像识别任务需要 309 亿次浮点运算,这对于计算资源有效的嵌入式设备来说是巨大的挑战.3)推理时间.对于一张高分辨率的输入图像来说,嵌入式平台执行深度卷积神经网络前向推理过程是非常耗时的,可能要几分钟才能处理一张图像,这对于实时应用来说是不可接受的.

因此,很多研究者通过模型压缩方式减小模型复杂度,减少模型参数量与计算量,加快模型前向推理速度.其中比较经典的方法包括模型结构优化[20-23]、低秩分解[24-25]、模型剪枝[26-27]、模型量化[26,28]、知识蒸馏[29]等.模型压缩方法在降低模型参数量和计算量的同时,势必会带来一定的精度损失.对于无人机场景目标检测来说,如何在保持精度基本无损的情况下,进行最大限度的模型压缩是关键的.

针对上述问题,本文提出了一种基于迭代稀疏训练的模型压缩方法,其主要贡献有 3 个方面:

1) 通过迭代稀疏训练的方式,对经典目标检测网络 YOLOv3 进行通道和层协同剪枝,可以在保持精度基本无损的情况下实现最大限度的模型压缩;

2) 通过组合不同数据增强方式增加了数据集分布的复杂性和多样性,结合一些优化手段(Tricks),提高了无人机场景下目标检测网络 YOLOv3 的泛化性能;

3) 实验证明,该方法对于无人机场景下目标检测网络 YOLOv3 模型压缩效果明显,在保证精度基本无损的情况下可以实现最大限度的模型压缩,而且可以极大地加速模型的前向推理过程,使得模型实时应用成为可能.

## 1　相关工作

### 1.1　目标检测

2013 年以前,目标检测问题通过手工设计的特征算子与滑动窗口方式解决,处理速度慢、效果不鲁棒,难以满足实际工业需求.2014 年,Girshick 等人[11]提出 R-CNN 算法,使用深度卷积神经网络完成对输入图像的特征提取工作,紧接着一系列基于深度卷积神经网络的检测算法出现,包括基于 2 阶段的检测算法 Fast R-CNN[12],Faster R-CNN[13],Mask R-CNN[6],R-FCN[14]等,以及 1 阶段的检测算法

YOLO[15-17], SSD[18], RetinaNet[19]等.随着检测算法的发展,研究者们逐渐发现基于候选区域的目标检测存在一定的缺陷,因此出现了不基于候选区域的目标检测器,典型的有 CornerNet[9], CenterNet[10]等算法.

1.1.1 基于 2 阶段的目标检测器

基于 2 阶段的目标检测过程主要由 2 部分组成,首先生成包含感兴趣区域的高质量候选框,然后通过进一步分类与回归获得检测结果.R-CNN 是 2 阶段检测器的基础,首先使用深度网络提取图像特征,然后基于选择搜索[30]生成候选区域,最后通过支持向量机(support vector machine, SVM)分类器对候选区特征进行分类从而得到最终的检测结果.R-CNN 不能进行端到端训练,随着网络的发展,逐渐形成了较为成熟的 Faster R-CNN 检测算法,后续的改进和提升基本都是基于 Faster R-CNN 进行的.以 R-CNN 算法为代表的 2 阶段检测器经过不断的发展和改善,尤其是加入 RPN(region proposal network)[13]结构以后,检测精度越来越高,计算复杂度也越来较高,检测速度较慢,难以满足实时应用的需求.因此一般计算能力受限的嵌入式平台不会使用 2 阶段检测器进行部署应用.

1.1.2 基于 1 阶段的目标检测器

不同于 2 阶段检测器,1 阶段检测器不包含候选区域生成步骤,直接对预定义锚框对应的特征进行分类和回归以得到最终的检测结果.这种方法相比 2 阶段检测器来说性能有一定的下降,但速度有明显提升.YOLO[15]首先对训练样本聚类得到候选框,使用这些预定义的候选框密集地覆盖整个图像空间位置,然后提取输入图像的特征并对预定义的候选框进行分类和回归.SSD[18]通过长宽及长宽比来预定义锚框,为提升 1 阶段检测器的性能瓶颈,SSD 从多个尺度特征层出发,同时对预定义的候选框进行类别概率学习以及坐标位置回归,有效提升了检测性能.后续提出的 RetinaNet[19]引入 focal loss 函数来解决类别不平衡问题,1 阶段检测器性能有了进一步的提升.

1 阶段检测器中 YOLO 系列算法运行速度非常快,可以达到实时应用级别,但精度差强人意.后续 YOLOv2[16], YOLOv3[17]的出现,将精度进行了大幅提升.特别是 YOLOv3,借鉴了 ResNet(residual network)[31]的残差思想和 FPN(feature pyramid network)[32]的多尺度检测思想,在保持速度优势的前提下,进一步提升了检测精度,尤其加强了对小目标的识别能力.在实际部署中,由于非常好的速度/精度均衡性和高度的集成性、灵活性,YOLOv3 成为工业界最受欢迎的模型之一.

1.1.3 不基于候选区域的目标检测

2 阶段检测器与 1 阶段检测器都基于候选框进行分类与回归,实际应用中,很多研究者发现如果候选框设置不合适,可能影响检测器的性能.因此,不基于候选区域的目标检测方法被提出,这些方法将目标检测任务转换为关键点检测与尺寸估计.CornerNet[9]将目标检测任务转换为左上角和右下角的关键点检测任务;ExtremeNet[33]将目标检测任务转换为检测目标的 4 个极值点;CenterNet[10]将左上角、右下角和中心点结合成为三元组进行目标框的判断;FoveaBox[34]则借鉴了语义分割思想,对目标上每个点都预测一个分类结果,物体边界框通过预测偏移量得到.不基于候选区域的目标检测器的成功主要得因于 FPN 和 focal loss 结构,但与 2 阶段检测器和级联方法的检测精度仍然有差距,在灵活性和检测速度上也没有明显优势.因此,目前工业部署应用不是特别广泛.当然,作为一种新的检测思路,随着方法本身的发展,相信未来将会有更多的应用.

## 1.2 模型压缩

在资源有限的设备上部署深度网络模型时,模型压缩是非常有效的工具,常用的模型压缩方法主要包括模型结构优化[20-23]、低秩分解[24-25]、模型剪枝[26-27]、模型量化[26,28]、知识蒸馏[29]等.模型结构优化是指通过使用轻量化的网络结构来降低模型计算量,但这样会导致模型表达能力下降,从而造成性能下降.低秩分解将原始网络权值矩阵当作满秩矩阵,用若干个低秩矩阵的组合来替换原来的满秩矩阵,低秩矩阵又可以分解为小规模矩阵的乘积,从而实现模型压缩和加速.模型量化主要利用 32 b 表示权重数据存在的冗余信息,使用更少位数表示权重参数,从而实现模型压缩和加速.上述 2 种方法在压缩率较高时,都面临精度损失大的问题.知识蒸馏通过学生网络对老师网络的拟合得到一个更加紧凑的网络结构,以再现大型网络的输出结果.学生网络的选择与设计,以及学习老师网络的哪些信息依然是一个值得考虑的问题.模型剪枝指对一个已训练好的高精度复杂模型,通过一种有效的评判手段来判断参数的重要性,将不重要的连接进行裁剪以减少参数冗余,从而实现模型压缩和加速.

Denil 等人在文献[35]中提出很多深度卷积神经网络中存在显著冗余,仅仅使用很少一部分(5%)权值就足以预测剩余的权值,因此模型剪枝可以实现非常可观的压缩率.针对无人机场景数据集,本文选择速度/精度均衡的 YOLOv3 作为基础检测模型,通过模型剪枝来进行压缩和加速,以获得可在算力有限的嵌入式平台部署应用的快速准确检测模型.

### 1.3　数据增强

数据增强的目标是增加输入图像的可变性,以使模型对从不同环境获得的图像具有更高的鲁棒性.常用的 2 种数据增强方式包括空间几何变换与像素颜色变换[36-39],其中图像平移、旋转、缩放以及图像亮度、对比度、饱和度是比较常用的.上述提到的数据增强方法都是基于像素级别的,对于无人机场景下的稠密小目标检测任务来说提升作用有限.Kisantal 等人在文献[40]中通过剪切粘贴方式来增加小目标数量,该种方法对稠密分布的小目标效果不明显.此外,一些研究员提出使用多张图像一起执行数据增强的方法.Mixup[41]对任意 2 张训练图像进行像素混合从而起到正则化效果;Mosaic[42]任意混合 4 张训练图像作为一张新的图像参与训练.这 2 种方法混合了不同的上下文信息,有效提升了数据集的场景复杂性,从而提升了模型的检测性能.本文除了使用基础的空间几何变换与像素颜色变换增强方式外,巧妙结合了 Mixup 和 Mosaic 两种增强方式,进一步提升了 YOLOv3 在无人机场景下的检测精度.

## 2　基于迭代稀疏训练的模型剪枝

深度卷积神经网络主要模块包括卷积、池化、激活,这是一个标准的非线性变换模块.网络层数加深,意味着更好的非线性表达能力,可以学习更加复杂的变换从而拟合更加复杂的特征输入.因此,对于同一场景数据集来说,网络越复杂,通常意味拟合能力越强,效果越好.而复杂模型参数量大,推理速度慢,对于嵌入式平台部署应用来说是一个很大的挑战.文献[35]还提出这些剩下的权值甚至可以直接不用被学习,也就是说,仅仅训练一小部分原来的权值参数就有可能达到和原来网络相近甚至超过原来网络的性能.受此启发,本文对一个特征表达能力强的复杂模型通过迭代稀疏训练的方式,巧妙结合层剪枝和通道剪枝,在保持检测精度的同时,实现了模型极大比例的压缩,剪枝流程如图 1 所示:

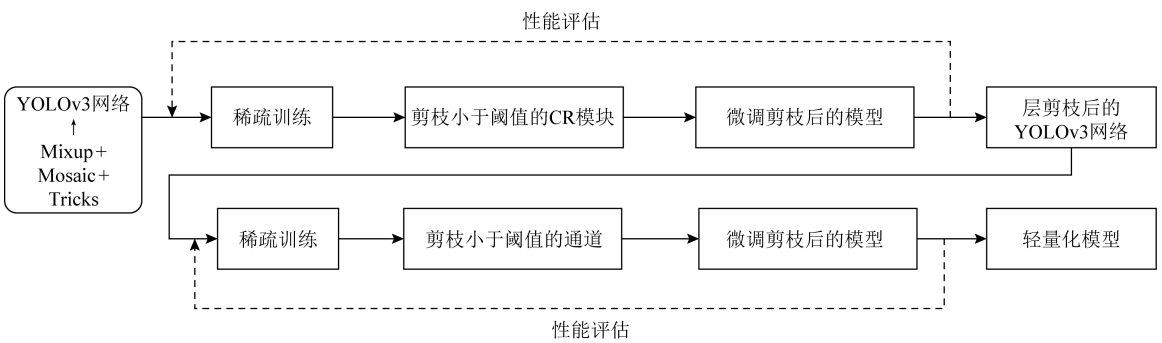



Fig. 1　The process of iterative sparse training

图 1　迭代稀疏训练流程图

### 2.1　网络结构

在实际部署应用中,由于非常好的速度、精度均衡和高度的集成性、灵活性,YOLO 系列模型成为最受欢迎的模型之一.YOLOv3 借鉴了 ResNet[31]的残差思想和 FPN[32]的多尺度检测思想,在保持速度优势的前提下,进一步提升了检测精度,尤其加强了对小目标的识别能力.如图 2 所示为 YOLOv3 网络结构,该结构由大量 1×1 卷积、3×3 卷积与残差结构组成,我们把每个这样的结构作为一个单元称为 CR 块.在检测部分,YOLOv3 引入了特征金字塔的思想,采用 3 个不同尺度的特征图进行检测,以应对目标尺度变化大的问题.本文基础模型基于 YOLOv3 进行,相比 Zhang 等人在文献[43]中基于 YOLOv3-SPP3(spatial pyramid pooling)开展的工作,同样剪枝率下本文模型参数量更少,推理速度更快,精度更高.

### 2.2　数据增强

数据增强是目标检测中常用的不需要增加太多训练成本但可以有效提升模型性能的方法.Mixup[41]是一种新的数据增强策略,通过对任意 2 张训练图像进行像素混合来对神经网络起到正则化的效果,可以提高网络对空间扰动的泛化能力,有助于提升

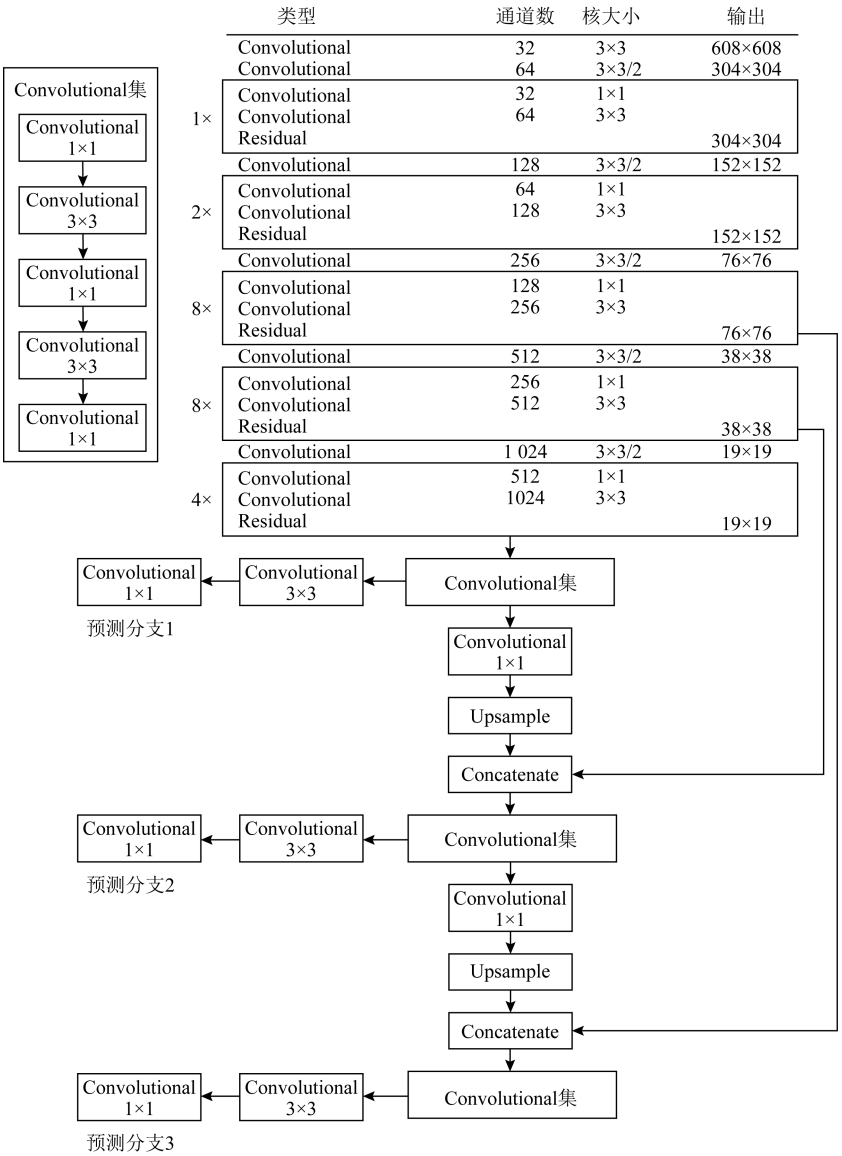



Fig. 2 The network structure of YOLOv3

图 2 YOLOv3 网络结构

检测精准率.Rosenfeld 等人在文献[44]中做过一个非常有趣的实验,称为“房间里的大象”,将包含大象的补丁图像随机放置在自然图像上,使用现有的物体检测模型进行检测,发现效果并不理想.Zhang 等人在文献[45]方案中将 Mixup 应用于目标检测,有效提升了模型对“房间里的大象”现象的鲁棒性.如式(1)所示为 YOLOv3 损失函数:

$$L_{\text{loss}}(b,\hat{b},c,\hat{c},o,\hat{o})=\frac{1}{N}(\alpha L_{\text{loc}}(b,\hat{b})+\beta L_{\text{cls}}(c,\hat{c})+\gamma L_{\text{obj}}(o,\hat{o})),\quad(1)$$

其中,$N$ 表示匹配到的默认框数目,$b,c,o$ 分别表示与标注框匹配到的候选框的坐标、类别置信度和目标置信度,$\hat{b},\hat{c},\hat{o}$ 分别表示标注框的坐标、类别与目标信息.

增加了 Mixup 数据增广策略之后 YOLOv3 的损失函数为:

$$L_{\text{loss}}(b,\tilde{b},c,\tilde{c},o,\tilde{o})=\rho L_{\text{loss}}(b,\hat{b}_1,c,\rho\hat{c}_1,o,\rho\hat{o}_1)+(1-\rho)L_{\text{loss}}(b,\hat{b}_2,c,(1-\rho)\hat{c}_2,o,(1-\rho)\hat{o}_2),\quad(2)$$

$$\rho\sim U[0,1),$$

$$\tilde{\boldsymbol{x}}=\rho\boldsymbol{x}_1+(1-\rho)\boldsymbol{x}_2,$$

$$\tilde{b}=\hat{b}_1\bigcup\hat{b}_2,$$

$$\tilde{c}=\rho\hat{c}_1\bigcup(1-\rho)\hat{c}_2,$$

$$\tilde{o}=\rho\hat{o}_1\bigcup(1-\rho)\hat{o}_2,$$

其中,$\boldsymbol{x}_i$ 表示原始图像的坐标向量,$\rho$ 表示混合因子,服从均匀分布 $U[0,1)$,$\tilde{b},\tilde{c},\tilde{o}$ 分别表示经过 Mixup 混合后的标注框的坐标、类别与目标信息.

Mosaic[42] 是另一种有效的多张图像混合的数据

增强方法,通过混合 4 张不同图像的上下文信息,有效提升数据集的场景复杂性,有助于提升检测召回率.假设混合后的图像分辨率为 $S\times S$,首先创建一个 $2S\times 2S$ 的大图,则混合图像的中心点坐标 $x_c$,$y_c\sim U[S\times 0.5, S\times 1.5]$,随机从训练集中选取 4 张图像,原图像坐标换算得到混合图像,然后中心裁剪得到最终的增强图像.混合图像左上、右上、左下和右下 4 张图像坐标范围为

左上角坐标:

$$x_{\text{img1\_start}}, y_{\text{img1\_start}}, x_{\text{img1\_end}}, y_{\text{img1\_end}} = \max(x_c - w_{\text{img1}}, 0), \max(y_c - h_{\text{img1}}, 0), x_c, y_c;$$

右上角坐标:

$$x_{\text{img2\_start}}, y_{\text{img2\_start}}, x_{\text{img2\_end}}, y_{\text{img2\_end}} = x_c, \max(y_c - h_{\text{img2}}, 0), \min(x_c + w_{\text{img2}}, 2\times S), y_c;$$

左下角坐标:

$$x_{\text{img3\_start}}, y_{\text{img3\_start}}, x_{\text{img3\_end}}, y_{\text{img3\_end}} = \max(x_c - w_{\text{img3}}, 0), y_c, x_c, \min(y_c + h_{\text{img3}}, 2\times S);$$

右下角坐标:

$$x_{\text{img4\_start}}, y_{\text{img4\_start}}, x_{\text{img4\_end}}, y_{\text{img4\_end}} = x_c, y_c, \min(x_c + w_{\text{img4}}, 2\times S), \min(y_c + h_{\text{img4}}, 2\times S). \tag{3}$$

其中,$x$,$y$,$w$,$h$ 分别表示图像的横纵坐标、宽和高.

Mixup 通过任意 2 张图像的像素混合可以有效提升检测精准率,Mosaic 通过任意 4 张图像拼接混合可以有效提升检测召回率,本文将 Mixup 和 Mosaic 这 2 种数据增强策略进行巧妙结合,获得了检测精准率和召回率的大幅提升.在训练时,首先加载 *batchsize* 张训练图像,然后对每张图像按照一定概率选择数据增强策略,其中包含只进行 Mixup 增强、只进行 Mosaic 增强、同时进行 Mixup 和 Mosaic 增强与 Mixup 和 Mosaic 增强都不执行这 4 种方式.需要特别注意的是在选择与其他图像结合时,需要从 *batch* 以外的图像中选择.对 *batch* 中的每张图像都执行上述操作,直至最后得到新生成的 *batchsize* 张增强图像参与模型训练,上述组合增强策略为

$$dataaug = a_1 mixup + a_2 mosaic + a_3(mixup + mosaic) + a_4 None. \tag{4}$$

## 2.3　模型剪枝

模型剪枝通过对高精度复杂模型进行冗余参数修剪,从而得到计算量和参数量有效降低的轻量化模型.除了 2.2 节提到的数据增强策略外,本文参考 He 等人在文献[46]方案中的设置,在 YOLOv3 训练过程中加入了相关优化方法,其中包括多尺度训练、大的批处理尺寸、热启动、余弦学习率衰减法(cos lr decay)、标签平滑(label smooth)等,有效提高了模型的精度.将该模型作为剪枝基准模型,依次通过迭代稀疏、层剪枝、微调与迭代稀疏、通道剪枝、微调(finetune)的过程,最终得到了不同剪枝比例的可部署模型,剪枝流程如图 1 所示.

### 2.3.1　稀疏训练

模型剪枝需要选择可评估权重重要性的指标来剪枝不重要的结构.本文采用批归一化层(batch normalization, BN)中的缩放因子 $\gamma$ 的绝对值度量权重的重要程度.BN 层的标准化公式为

$$y = \gamma \times \frac{x-\mu}{\sqrt{\sigma^2+\varepsilon}} + \beta. \tag{5}$$

其中,$\mu$ 和 $\sigma^2$ 为统计参数,分别表示批处理中数据的均值和方差;$\gamma$ 和 $\beta$ 为可训练参数,分别对归一化分布进行缩放与平移,从而保留原始学习到的特征.基于 BN 层原理,缩放系数 $\gamma$ 主要用来保留归一化操作前的特征,因此可以作为评估权重重要程度的指标因子.

如果权重分布不稀疏将不利于模型剪枝,需要通过外力强制稀疏化,L1 正则化约束是常用的稀疏化方法.加入 L1 正则后 YOLOv3 的损失函数为:

$$L_{\text{sparsity}} = L_{\text{loss}}(b, \hat{b}, c, \hat{c}, o, \hat{o}) + \partial \sum_{\gamma\in\Gamma} g(\gamma), \quad g(\gamma) = |\gamma|. \tag{6}$$

其中,$\partial$ 为正则化系数,平衡 2 个损失项之间的关系.由于添加的 L1 正则项梯度不是处处存在的(在 0 点导数不存在),因此我们使用次梯度下降[27]作为 L1 正则项的优化方法.

### 2.3.2　层剪枝

如图 2 所示为 YOLOv3 的标准网络结构,特征提取部分由大量 $1\times 1$ 卷积、$3\times 3$ 卷积与残差结构构成,我们把每个这样的结构作为一个单元称为 CR 块.实际剪枝操作过程中,为保证 YOLOv3 结构完整,CR 块作为一个整体参与.YOLOv3 结构中一共有 23 个 CR 单元,权重重要程度统计时,我们选择 CR 单元的均值作为评估剪枝与否的指标,剪掉均值较低的单元.层剪枝是一种非常粗糙的剪枝方法,按照单元裁剪的方式可能会带来较大的精度损失,因此实际剪枝过程中我们通过观察 $\gamma$ 的变化趋势设置需要剪枝的 CR 单元比例,通常不会一次剪掉太多,每次迭代按照 50% 的概率进行剪枝.通过稀疏训练-层剪枝-微调的迭代过程,保证每次剪掉一些 CR 单元后,其精度损失可以通过微调补偿回来.图 3 为层剪枝后的 YOLOv3 网络结构图,具体剪枝参数设置见 3.3 节.

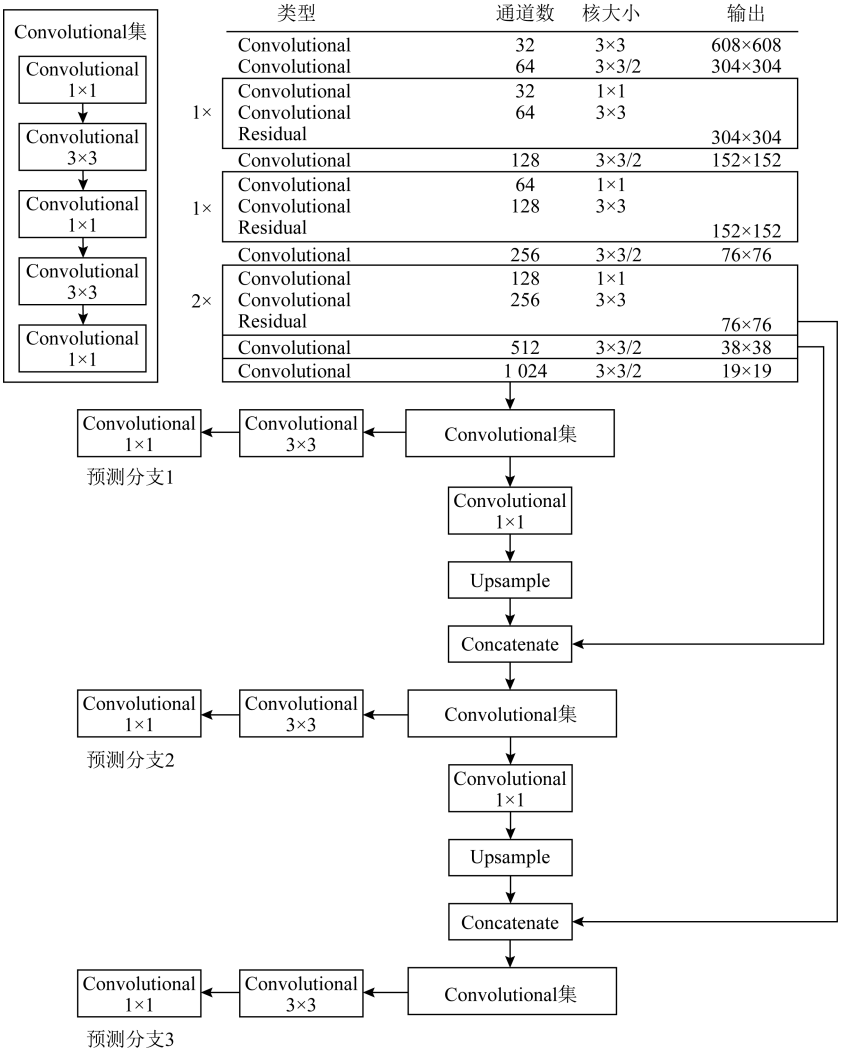



Fig. 3 The network structure of YOLOv3 after pruning

图 3 YOLOv3 剪枝后的网络结构

2.3.3 通道剪枝

层剪枝之后我们得到了相对稳定的网络结构，此时通过通道剪枝对其进行进一步修剪.首先需要定义全局剪枝率 $\lambda$，表示需要剪枝的通道数占总通道数的比例，同时可以换算出全局剪枝率 $\lambda$ 对应 $\gamma$ 的阈值 $\pi_1$.为防止通道剪枝时某些层被全部剪掉，需要另外设置一个保护阈值 $\beta$，表示每层中至少需要保留多少比例的通道，同理也可以换算出每层中对应 $\gamma$ 的阈值 $\pi_2$.当且仅当 $\gamma<\pi_1$，$\gamma<\pi_2$ 同时满足时，该通道可以剪枝.直接使用较大的剪枝率会损失较多的精度，因此通道剪枝时我们通过分析稀疏训练后 $\gamma$ 的分布，按照 30%的比例设置参数量剪枝率并迭代执行稀疏、通道剪枝、微调的过程，使得由于剪枝带来的精度下降可以通过微调补偿回来，从而得到精度损失较小的轻量化模型.

2.3.4 微 调

在执行完一次层剪枝或通道剪枝后，或多或少都会带来精度损失，一定程度的精度损失可以通过后续的微调补偿回来.本文我们是通过迭代稀疏训练进行剪枝的，因此在每一次剪枝之后必须通过微调来补偿剪枝带来的精度损失，才能不影响下一次迭代的稀疏训练.

## 3 实验结果与分析

本文提出了一种基于迭代稀疏训练的模型剪枝方法.实验证明，该方法可以获得精度和速度兼备的检测模型.其中，模型训练环境均基于 Pytorch 框架，服务器配置为 Intel® Xeon® CPU E5-2640 v4@2.40 GHz，128GRAM，4 块 GTX 1080 Ti.

### 3.1　数据集和评价指标

VisDrone-DET2019[47]是一个典型的无人机目标检测应用数据集.该数据集包含 8 599 张来自无人机平台从不同高度、不同光照条件下收集的图像,涉及包含人、车在内的 10 个类别(pedestrian,person,car,van,bus,truck,motor,bicycle,awning-tricycle,tricycle),是一个非常有挑战的数据集.其中训练集 6 471 张、验证集 548 张、测试集 1 580 张,由于测试集标注未开放下载,因此本文中模型都是基于训练集训练、验证集评估的.

模型评估标准与文献[43]中保持一致,我们主要评估输入分辨率为 608×608 像素下模型的参数量、计算量(floating-point operations, FLOPS)、交并比(intersection over union, IOU)在 0.5 下的精度度量指标 $F1$-score 和平均精度均值(mean average precision, $mAP$)以及模型压缩中涉及的参数量压缩率、计算量压缩率等指标.为更全面地评估轻量化模型在不同分辨率下的性能,我们选取计算量压缩率为 85%的模型与文献[43]中模型进行对比分析,实验证明我们的算法在达到与文献[43]中同等压缩率的情况下可以获得更高的精度与更快的速度.

### 3.2　训练细节

本文模型训练环境基于 Pytorch 框架展开,采用动量 0.9、权重衰减因子 0.000 5 的随机梯度下降(stochastic gradient descent, SGD)权重更新策略;正常训练迭代次数为 300epoch,初始学习率为 0.002 5,分别在 240epoch 和 270epoch 衰减 10 倍;稀疏训练和微调的迭代次数为 100epoch,初始学习率为 0.001,分别在 80epoch 和 90epoch 衰减 10 倍;批处理大小为 128;损失函数中 $\alpha,\beta,\gamma$ 分别设置为 3.31,52.0,42.4;数据增强策略中 $\alpha_1,\alpha_2,\alpha_3,\alpha_4$ 按照等概率设置为 0.25;多尺度训练设置图像大小变化区间为[416,896];稀疏因子根据 $\gamma$ 分布设置为 0.01～0.08 之间.

### 3.3　实验结果

#### 3.3.1　数据增强

数据增强是一种成本很低却可以有效提升模型性能的方法.除了常规使用的增强方法外,本文结合了 Mixup 和 Mosaic 的优势,有效提升了检测精准率和召回率,增强效果如图 4 所示:

(a) 不作数据增强　(b) Mixup数据增强　(c) Mosaic数据增强　(d) Mixup和Mosaic数据增强

Fig. 4　The effect of data enhancement

图 4　数据增强效果图

为获得好的基准模型,我们通过数据增强策略与优化手段共同对 YOLOv3 标准模型进行了优化.表 1 中前两行分别表示 Zhang 等人在文献[43]中的测试结果,行 3 表示 YOLOv3 标准网络测试结果,

**Table 1　Test Results Comparison of Benchmark Model**

**表 1　基准模型测试结果比较**

| 模型 | 输入/pixel | 参数量/MB | 计算量/GFLOPS | $mAP$/% | $F1$-score/% |
|---|---|---|---|---|---|
| YOLOv3-SPP1[43] | 608 | 62.60 | 140.36 | 22.9 | 36.9 |
| YOLOv3-SPP3[43] | 608 | 63.90 | 151.72 | 23.3 | 37.6 |
| YOLOv3 | 608 | 61.57 | 139.77 | 25.1 | 35.2 |
| YOLOv3+Mixup | 608 | 61.57 | 139.77 | 25.3 | 36.1 |
| YOLOv3+Mosaic | 608 | 61.57 | 139.77 | 25.8 | 36.5 |
| YOLOv3+Mixup+Mosaic | 608 | 61.57 | 139.77 | 26.5 | 37.0 |
| YOLOv3+Mixup+Mosaic+Tricks | 608 | **61.57** | **139.77** | **27.5** | **38.1** |

注:黑体数值表示通过数据增强与相关优化手段结合得到的基准模型测试结果.

行 4～6 分别表示单独加入 Mixup 增强策略、单独加入 Mosaic 增强策略以及 Mixup 和 Mosaic 结合策略的测试结果，最后一行表示加入相关优化手段的测试结果.我们可以看到不论使用 Mixup 策略还是 Mosaic 策略都可以提升 $mAP$ 指标，此外 Mixup 和 Mosaic 随机组合的策略对 $mAP$ 的提升要比单独执行时效果更好.表 1 最后一行不论 $mAP$ 还是 $F1$-score 都有很大的提升，虽然未加 SPP[48] 模块，但通过数据增强策略与相关优化手段可以达到甚至优于 YOLOv3-SPP3 的结果，而且参数量和计算量更少.

### 3.3.2 剪 枝

基准模型获得后，我们通过迭代稀疏训练的方式进行模型剪枝.图 5 为层剪枝稀疏训练过程中 CR 单元 $\gamma$ 均值变化曲线图，其中实线表示基准模型 CR 单元 $\gamma$ 均值曲线，虚线表示稀疏训练后模型 CR 单元 $\gamma$ 均值曲线.从图 5 中可以明显发现，模型越靠前的层 $\gamma$ 均值越大，越靠后的层 $\gamma$ 均值越小.随着迭代的进行，$\gamma$ 均值会逐渐变小，但不改变层之间的相对大小关系，而且越靠后的层越接近 0，说明针对当前数据集来说，越靠后的层对模型性能贡献越小，因此会在层剪枝过程中被剪枝.随着剪枝过程的进行，其稀疏训练过程也会变得越来越难，层剪枝难度增大，直至达到一定的结构平衡，如图 3 所示，完成层剪枝过程.

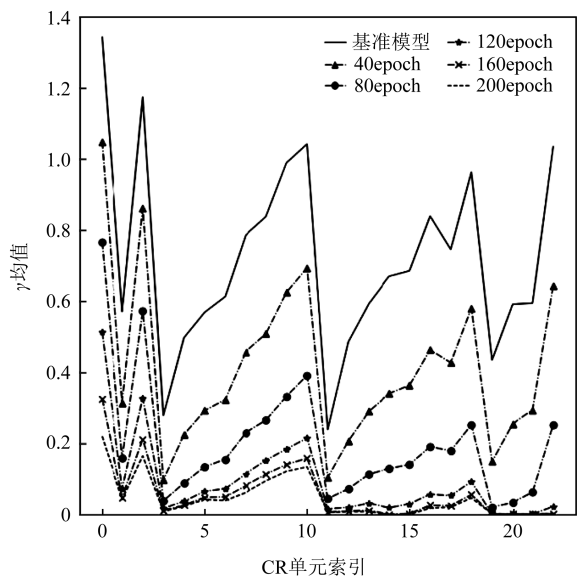
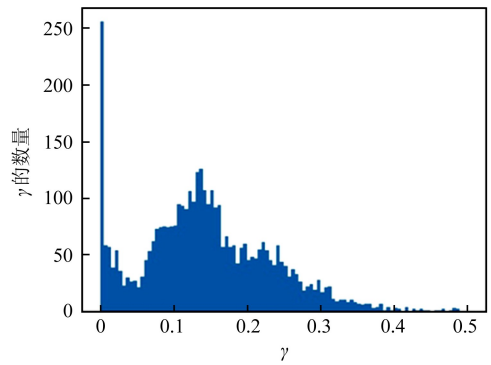


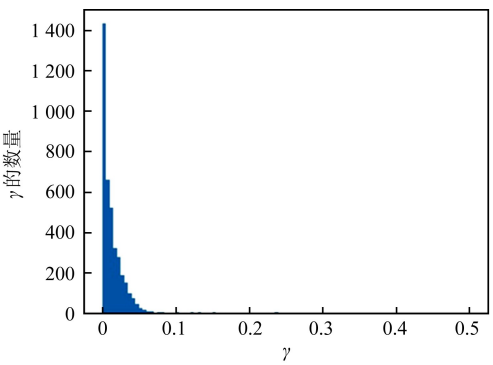

Fig. 5 The scale factor change curve of CR module

图 5 CR 单元缩放因子变化曲线图

标准 YOLOv3 模型一共含有 72 个 BN 层，如图 6 所示为通道剪枝过程中稀疏训练前后 $\gamma$ 统计直方图分布.对比图 6(a)(b)的 2 张图可以明显地看到，稀疏训练后 $\gamma$ 数据分布直方图变得更细更尖了，意味着接近于 0 的 $\gamma$ 值变得更多了，而较大的 $\gamma$ 值变少了，也即 $\gamma$ 数据分布更加稀疏，有利于通道剪枝的实现.




(a) 基准模型$\gamma$统计直方图




(b) 稀疏训练后$\gamma$统计直方图

Fig. 6 The $\gamma$ statistics histogram before and after channel pruning sparse training

图 6 通道剪枝稀疏训练前后 $\gamma$ 统计直方图

表 2 表示不同剪枝率下的模型压缩性能对比，其中 YOLOv3_baseline 表示本文剪枝前的基准模型，YOLOv3_prune_70(85,90)分别表示计算量剪枝率为 70%，85%，90%的模型，与文献[43]工作中的 SlimYOLOv3-SPP3-50(90,95)一一对应.在计算量达到同等剪枝率甚至更高剪枝率的条件下，我们的方法对参数量的压缩率更高，而且精度误差更低.特别是对于计算量压缩率为 85%时，参数量压缩率达到 97.3%，比文献[43]方案中的 87.48%高了将近 10%.此外，$mAP$ 误差只有 0.3，相比文献[43]方案中提升了 2.4 个点.

为了与文献[43]方案中结果进行更公平全面的对比，表 3 针对计算量 85%压缩率水平进行了 416×416，608×608，832×832 这 3 个分辨率的性能对比，不出意料，不论 $mAP$ 指标还是 $F1$-score 指标，

我们的方法都取得了巨大的提升.图 7 展示了 608×608 输入分辨率下剪枝前后的检测效果图,可以看到剪枝前后都能检测到绝大多数感兴趣目标,而且剪枝前后误差很小.

**Table 2　Comparison of Model Compression Performance Under Different Pruning Rates**

**表 2　不同剪枝率下的模型压缩性能对比**

| 模型 | 输入/pixel | 参数量/MB | 参数量剪枝率/% | 计算量/GFLOPS | 计算量剪枝率/% | *mAP*/% | *mAP* 误差/% |
|---|---|---|---|---|---|---|---|
| YOLOv3_baseline | 608 | 61.57 | | 139.77 | | 27.5 | |
| YOLOv3_prune_70 | 608 | 7.40 | 87.98 | 42.93 | 69.29 | 27.5 | 0 |
| YOLOv3_prune_85 | **608** | **1.66** | **97.30** | **21.60** | **84.55** | **27.2** | **−0.3** |
| YOLOv3_prune_90 | 608 | 0.79 | 98.72 | 13.94 | 90.03 | 25.5 | −2.0 |
| SlimYOLOv3-SPP3-50[43] | 608 | 20.80 | 67.45 | 65.17 | 57.05 | 22.6 | −0.7 |
| SlimYOLOv3-SPP3-90[43] | 608 | 8.00 | 87.48 | 21.30 | 85.96 | 20.6 | −2.7 |
| SlimYOLOv3-SPP3-95[43] | 608 | 5.10 | 92.02 | 14.04 | 90.75 | 19.1 | −4.2 |

注:黑体数值表示剪枝率为 85%时综合性能最好.

**Table 3　Comparison of Model Compression Performance Under Different Resolutions**

**表 3　不同分辨率下的模型压缩性能对比**

| 模型 | 输入/pixel | 参数量/MB | 计算量/GFLOPS | *mAP*/% | *F*1-score/% |
|---|---|---|---|---|---|
| YOLOv3_prune_85 | 416 | 1.66 | 10.11 | 17.5 | 27.9 |
| YOLOv3_prune_85 | **608** | **1.66** | **21.6** | **27.2** | **37.5** |
| YOLOv3_prune_85 | 832 | 1.66 | 40.45 | 30.1 | 40.2 |
| SlimYOLOv3-SPP3-90[43] | 416 | 8.0 | 9.97 | 14.5 | 24.4 |
| SlimYOLOv3-SPP3-90[43] | 608 | 8.0 | 21.3 | 20.6 | 32.0 |
| SlimYOLOv3-SPP3-90[43] | 832 | 8.0 | 39.89 | 23.9 | 34.0 |

注:黑体数值表示分辨率为 608 时综合性能最好.

(a) YOLOv3网络　(b) YOLOv3_prune_85网络

Fig. 7　Test result of model before and after pruning

图 7　剪枝前后结果图

## 4　结束语

本文通过数据增强策略与相关优化手段的巧妙结合,得到了一个性能较好的基准模型.通过迭代稀疏训练的模型压缩方法,对上述基准模型进行了极大程度的参数量压缩与计算量压缩.与现有典型方法[43]相比,本文提出的方法在未添加额外计算量(SPP 模块)的情况下,得到了精度更高的基准模型;同时通过迭代稀疏层剪枝与迭代稀疏通道剪枝相结合的方法,在达到同等计算量剪枝率的情况下,参数量更少,精度损失更小,适合在计算资源受限的无人机平台部署应用.模型压缩是生成性能较好的轻量化模型的常用方法,如何保证在一定精度损失的条件下进一步降低计算量和参数量仍然是一个值得深度研究的课题.当前方法只验证了 YOLOv3 网络在

VisDrone-DET2019 数据集上的压缩性能,下一步我们将对其他典型检测网络与数据集做进一步验证和推广,以提高方法的普适性.

**作者贡献声明**:侯鑫参与代码开发、实验测试、数据分析、论文撰写与修改等工作;曲国远、魏大洲参与实验环境搭建、代码开发、实验测试等工作;张佳程参与数据分析等工作.

## 参 考 文 献

[1] Bhaskaranand M, Gibson J D. Low-complexity video encoding for UAV reconnaissance and surveillance [C] //Proc of IEEE 2011—MILCOM. Piscataway, NJ: IEEE, 2011: 1633-1638

[2] Madawalagama S, Munasinghe N, Dampegama S, et al. Low cost aerial mapping with consumer grade drones [J]. Coordinates, 2016, 8(4): 13-18

[3] Sa I, Hrabar S, Corke P. Outdoor flight testing of a pole inspection UAV incorporating high-speed vision [C] //Proc of Field and Service Robotics. Berlin: Springer, 2015: 107-121

[4] Tian Zhi, Shen Chunhua, Chen Hao, et al. Fcos: Fully convolutional one-stage object detection [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 9627-9636

[5] Shen Zhiqiang, Liu Zhuang, Li Jianguo, et al. Dsod: Learning deeply supervised object detectors from scratch [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 1919-1927

[6] He Kaiming, Gkioxari G, Dollár P, et al. Mask R-CNN [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 2961-2969

[7] Tan Mingxing, Pang Ruoming, Le Q V. Efficientdet: Scalable and efficient object detection [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 10781-10790

[8] Zhou Xingyi, Wang Dequan, Krähenbühl P. Objects as points [J]. arXiv preprint, arXiv:1904.07850, 2019

[9] Law H, Deng Jia. CornerNet: Detecting objects as paired keypoints [C] //Proc of the European Conf on Computer Vision (ECCV). Berlin: Springer, 2018: 734-750

[10] Duan Kaiwen, Bai Song, Xie Lingxi, et al. CenterNet: Object detection with keypoint triplets [J]. arXiv preprint, arXiv:1904.08189, 2019

[11] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2014: 580-587

[12] Girshick R. Fast R-CNN [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2015: 1440-1448

[13] Ren Shaoqing, He Kaiming, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks [C] //Proc of the 29th Advances in Neural Information Processing Systems. New York: Curran Associates, 2015: 91-99

[14] Dai Jifeng, Li Yi, He Kaiming, et al. R-FCN: Object detection via region-based fully convolutional networks [C] //Proc of the 30th Advances in Neural Information Processing Systems. New York: Curran Associates, 2016: 379-387

[15] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 779-788

[16] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 7263-7271

[17] Redmon J, Farhadi A. YOLOv3: An incremental improvement [J]. arXiv preprint, arXiv:1804.02767, 2018

[18] Liu Wei, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector [C] //Proc of the European Conf on Computer Vision. Berlin: Springer, 2016: 21-37

[19] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 2980-2988

[20] Howard A G, Zhu Menglong, Chen Bo, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications [J]. arXiv preprint, arXiv:1704.04861, 2017

[21] Sandler M, Howard A, Zhu Menglong, et al. MobileNetV2: Inverted residuals and linear bottlenecks [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 4510-4520

[22] Howard A, Sandler M, Chu G, et al. Searching for MobileNetV3 [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 1314-1324

[23] Zhang Xiangyu, Zhou Xinyu, Lin Mengxiao, et al. ShuffleNet: An extremely efficient convolutional neural network for mobile devices [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 6848-6856

[24] Denton E L, Zaremba W, Bruna J, et al. Exploiting linear structure within convolutional networks for efficient evaluation [C] //Proc of the 28th Advances in Neural Information Processing Systems. New York: Curran Associates, 2014: 1269-1277

[25] Jaderberg M, Vedaldi A, Zisserman A. Speeding up convolutional neural networks with low rank expansions [J]. arXiv preprint, arXiv:1405.3866, 2014

[26] Han Song, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network [C] //Proc of the 29th Advances in Neural Information Processing Systems. New York: Curran Associates, 2015: 1135-1143

[27] Liu Zhuang, Li Jianguo, Shen Zhiqiang, et al. Learning efficient convolutional networks through network slimming [C] //Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 2736-2744

[28] Chen Wenlin, Wilson J, Tyree S, et al. Compressing neural networks with the hashing trick [C] //Proc of the 32nd Int Conf on Machine Learning. New York: ACM, 2015: 2285-2294

[29] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network [J]. arXiv preprint, arXiv:1503.02531, 2015

[30] Uijlings J R R, Van De Sande K E A, Gevers T, et al. Selective search for object recognition [J]. International Journal of Computer Vision, 2013, 104(2): 154-171

[31] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770-778

[32] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 2117-2125

[33] Zhou Xingyi, Zhuo Jiacheng, Krahenbuhl P. Bottom-up object detection by grouping extreme and center points [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 850-859

[34] Kong Tao, Sun Fuchun, Liu Huaping, et al. FoveaBox: Beyond anchor-based object detector [J]. arXiv preprint, arXiv:1904.03797, 2019

[35] Denil M, Shakibi B, Dinh L, et al. Predicting parameters in deep learning [C] //Proc of the 27th Advances in Neural Information Processing Systems. New York: Curran Associates, 2013: 2148-2156

[36] Simard P Y, Steinkraus D, Platt J C. Best practices for convolutional neural networks applied to visual document analysis [C] //Proc of the 7th Int Conf on Document Analysis and Recognition(ICDAR). Los Alamitos, CA: IEEE Computer Society, 2003: 958-962

[37] Ciregan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2012: 3642-3649

[38] Wan Li, Zeiler M, Zhang Sixin, et al. Regularization of neural networks using dropconnect [C] //Proc of the 30th Int Conf on Machine Learning. New York: ACM, 2013: 1058-1066

[39] Sato I, Nishimura H, Yokoi K. Apac: Augmented pattern classification with neural networks [J]. arXiv preprint, arXiv:1505.03229, 2015

[40] Kisantal M, Wojna Z, Murawski J, et al. Augmentation for small object detection [J]. arXiv preprint, arXiv: 1902.07296, 2019

[41] Zhang Hongyi, Cisse M, Dauphin Y N, et al. Mixup: Beyond empirical risk minimization [J]. arXiv preprint, arXiv:1710.09412, 2017

[42] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection [J]. arXiv preprint, arXiv:2004.10934, 2020

[43] Zhang Pengyi, Zhong Yunxin, Li Xiaoqiong. SlimYOLOv3: Narrower, faster and better for real-time UAV applications [C] //Proc of the 2019 IEEE Int Conf on Computer Vision Workshops. Piscataway, NJ: IEEE, 2019: 37-45

[44] Rosenfeld A, Zemel R, Tsotsos J K. The elephant in the room [J]. arXiv preprint, arXiv:1808.03305, 2018

[45] Zhang Zhi, He Tong, Zhang Hang, et al. Bag of freebies for training object detection neural networks [J]. arXiv preprint, arXiv:1902.04103, 2019

[46] He Tong, Zhang Zhi, Zhang Hang, et al. Bag of tricks for image classification with convolutional neural networks [C] //Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 558-567

[47] Du Dawei, Zhu Pengfei, Wen Longyin, et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results [C] //Proc of the 2019 IEEE/CVF Int Conf on Computer Vision Workshop (ICCVW). Piscataway, NJ: IEEE, 2019: 213-226

[48] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904-1916

**Hou Xin**, born in 1993. Master and engineer. Member of CCF. Her main research interests include pattern recognition and computer vision.

**侯　鑫**,1993 年生.硕士,工程师,CCF 会员.主要研究方向为模式识别、计算机视觉.

**Qu Guoyuan**, born in 1983. Master and senior engineer. His main research interests include core computing platform architecture technology and aviation high security architecture technology.

**曲国远**,1983 年生.硕士,高级工程师.主要研究方向为核心计算平台架构技术、航空高安全体系架构技术.

**Wei Dazhou**, born in 1986. Master and engineer. His main research interests include embedded intelligent computing framework technology and multi-source sensor image processing technology.

**魏大洲**,1986 年生.硕士,工程师.主要研究方向为嵌入式智能计算框架技术、多源传感器图像处理技术.

**Zhang Jiacheng**, born in 1997. Master candidate. His main research interests include pattern recognition and computer vision.

**张佳程**,1997 年生.硕士研究生.主要研究方向为模式识别、计算机视觉.