基于改进 YOLOv3 算法在道路目标检测中的应用

谭芳喜[1],肖世德[1,2],周亮君[1],李晟尧[1]
(1. 西南交通大学 机械工程学院,四川 成都 610031;
2. 轨道交通运维技术与装备四川省重点实验室,四川 成都 610031)

摘 要:为了提高道路环境下目标检测的准确率和实时性,提出一种基于 YOLOv3 的改进检测算法。通过深度可分离卷积模块减少模型计算量,提高模型的实时性;采用 K-Means++聚类算法代替原来的 K-Means 算法生成本数据集所需的先验锚点框,解决 K-Means 算法受初始点选取的影响较大,聚类结果不稳定的问题;在 YOLOv3 的多尺度预测网络中引入 SENet(squeeze-and-excitation networks),加强网络对特征的学习能力;改进位置损失函数,解决使用 IoU(intersection over union)度量时存在无法反映预测框与真实框重合度大小、无法优化 IoU 为零等问题;利用 DIoU-NMS(基于 Distance-IoU 的非极大值抑制)去除冗余框,减少错误抑制,提高检测精度。实验结果表明,改进算法相对于原算法在检测耗时降低的同时,对 5 类常见目标检测的准确率均有提升。

关键词:目标检测;YOLOv3;深度可分离卷积;SENet;DIoU-NMS 算法
中图分类号:TP391 **文献标识码**:A **文章编号**:1673-629X(2021)08-0118-06
doi:10.3969/j.issn.1673-629X.2021.08.020

Application in Road Target Detection Based on Improved YOLOV3 Algorithm

TAN Fang-xi[1], XIAO Shi-de[1,2], ZHOU Liang-jun[1], LI Sheng-yao[1]
(1. School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China;
2. Key Laboratory of Sichuan Province for Rail Transit Operation and Maintenance Technology and Equipment, Chengdu 610031, China)

Abstract: In order to improve the accuracy and real-time performance of target detection in the road environment, an improved detection algorithm based on YOLOv3 is proposed. The deep separable convolution module is used to reduce the computational load of the model and improve the real-time performance of the model. K-Means++ clustering algorithm is used to replace the original K-Means algorithm to generate the a priori anchor point box, which solves the problem that the K-Means algorithm is greatly affected by the initial point selection and the clustering result is unstable. SENet is introduced into the multi-scale prediction network of YOLOv3 to strengthen the feature learning ability of the network. The location loss function is improved to solve the problems that the intersection over union measurement cannot reflect the intersection degree between the predicted box and the real box, and the IOU cannot be optimized to zero. The DIoU-NMS is used to remove redundant frames, so as to reduce error suppression and improve detection accuracy. The experiment shows that compared with the original algorithm, the proposed algorithm has improved the accuracy of the detection of five types of targets while reducing the detection time.

Key words: target detection; YOLOv3; deep separable convolution; SENet; DIoU-NMS

0 引 言

目标检测作为计算机视觉研究领域的热点,一直受到许多学者的关注。传统的目标检测算法主要是基于设计好的手工特征来进行检测,检测的效果主要与设计的特征质量有关。对于不同的目标而言,没有一种通用的特征,只能根据检测对象的特点有针对性地设计特征,所以鲁棒性很差[1-2]。近年来,基于深度学习的目标检测技术取得了长足的进步,无论是在检测效率还是准确率上都取得了很好的效果[3-4]。

目前基于深度学习的目标检测算法大致可以分为两种方式:Two-stage 目标检测算法与 One-stage 目标检测算法。Two-stage 系列算法在目标检测过程中分

收稿日期:2020-09-18 **修回日期**:2021-01-18
基金项目:四川省应用基础计划项目(2014JY0212)
作者简介:谭芳喜(1995-),男,硕士,研究方向为数字图像处理;肖世德,教授,研究方向为图像测控与智能机器人(车)

两步完成,首先通过算法生成一系列的候选区域,然后再通过卷积神经网络进行分类,代表性的算法有 RCNN[5]、Fast RCNN[6]、Faster RCNN[7]等,这一系列算法在检测的准确率和定位精度上相对要好一些,但在检测速度上无法满足实时性的需求。One-stage 系列算法将目标检测问题直接当作回归问题来处理,不产生候选区域,直接预测出目标位置与类别;代表性的算法有 YOLO[8]、YOLOv2[9]、YOLOv3[10]、SSD[11-12]、RetinaNet[13]等,One-stage 系列的算法精度上相对于 Two-stage 系列算法要低一些,但是检测速度要快很多。

该文以 One-stage 系列算法中的 YOLOv3 为基础,针对道路环境下的常见目标,提出了一种改进算法以增强 YOLOv3 的检测能力。通过引入深度可分离卷积模块减少参数计算量;采用 K-Means++算法确定数据集的先验锚点框;在多尺度预测网络结构中嵌入 SENet 模块,增强网络提取特征的能力;引入 DIoU 作为边界框位置损失函数,解决 IoU 无法反映预测框与真实框重合度大小、无法优化 IoU 为零等问题;最后利用 DIoU-NMS 去除冗余框,实现准确定位。

1　YOLOv3 介绍

1.1　YOLOv3 基本思想

YOLOv3 算法作为一种端到端的目标检测模型,只需要在输入端输入图像数据即可在输出端得到一个预测结果,预测结果为边界框的位置信息、置信度以及所属类别。如图 1 所示,其基本思想是将输入图片分割成 $S \times S$ 个网格,如果标注的目标中心坐标落在某个网格中,那么就由该网格来检测这个目标。每一个网格都会预测出 B 个边界框,每个边界框都包含位置信息(x,y,w,h)、置信度(Confidence)以及 C 个类别的概率,对于输出层来说,最终的输出维度为: $S \times S \times B \times (4 + 1 + C)$ 的张量。置信度是指边界框包含目标的可能性 Pr(object)以及包含目标情况下边界框准确度 IoU 的乘积,计算公式如下:

$$\text{Confidence} = \Pr(\text{object}) \times \text{IoU}_{\text{pred}}^{\text{truth}} \tag{1}$$

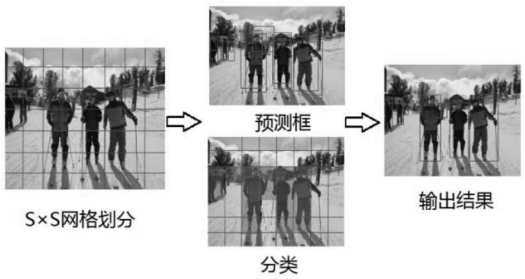

图 1　目标检测流程

上述公式中,当检测目标位于该网格中,Pr(object)= 1,否则为 0。IoU 用于表达真实框与预测框的重叠度,表示的是预测框和真实框的交集与并集的比值。当一个目标被多个检测框所预测时,通过设定阈值,将置信度低于阈值的边界框去除,并且对高于阈值的边界框进行非极大值抑制,去除多余的边框,最终得到最佳边界框。

1.2　YOLOv3 特征提取网络

YOLOv3 主要是由特征提取网络 Darknet-53 和多尺度预测网络两部分组成。其特征提取网络 Darknet-53 在 YOLOv2 的基础上引入了残差模块和跳跃连接构成了全新网络来对输入图像进行特征提取,其网络结构如图 2 所示。该网络主要由 1×1 和 3×3 的卷积层构成,减少了模型的参数量和计算量,在每次卷积之后都会加上一个 BN 层和 LeakyReIU 层,解决梯度爆炸和梯度消失的问题,加快网络的收敛速度。而且该网络中没有池化层和全连接层,特征图尺寸的变化是通过改变卷积核的步长来实现的,减少了特征信息的丢失。相比于 YOLOv2 使用的 Darknet-19,Darknet-53 加入了 5 个残差模块,使网络深度大大增加,虽然模型实时性能有所下降,但模型的检测精度大幅提升。Darknet-53 在 ImageNet 上的实验结果也表明,其分类精度与 ResNet-152 和 ResNet101 相差不大,但是网络层数比它们少,计算速度比它们快,是一种非常优秀的特征提取网络。

	Type	Filter	Size	Output
	Convolution	32	3×3	256×256
	Convolution	64	3×3/2	128×128
1×	Convolution	32	1×1	
	Convolution	64	3×3	
	Residual			128×128
	Convolution	128	3×3/2	64×64
2×	Convolution	64	1×1	
	Convolution	128	3×3	
	Residual			64×64
	Convolution	256	3×3/2	32×32
8×	Convolution	128	1×1	
	Convolution	256	3×3	
	Residual			32×32
	Convolution	512	3×3/2	16×16
8×	Convolution	256	1×1	
	Convolution	512	3×3	
	Residual			16×16
	Convolution	1 024	3×3/2	8×8
4×	Convolution	512	1×1	
	Convolution	1 024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1 000	
	Softmax			

图 2　Darknet-53 网络结构

2　基于 YOLOv3 的改进算法

2.1　先验框聚类

YOLOv3 引入了锚点框机制,锚点框的尺寸大小直接决定了在输出层每一个网格单元上做出的预测框

的大小。如果锚点框的大小一开始就和被检测的物体形状相近,那么训练的收敛速度就会更快,而且整体性能也会更好,反之则可能导致模型难以收敛。

YOLOv3 中选用了 K-Means 算法进行先验框聚类,但是 K-Means 算法受初始点选取的影响较大,从而容易导致聚类结果不稳定。故采用 K-Means++[14] 算法来进行聚类,该算法主要是对初始点的选取进行改进,基本思想就是初始点的聚类中心之间的距离要尽可能得远。算法流程如下:

(1)从数据集当中随机选取一个样本点,作为最初的聚类中心。

(2)计算每个样本点与当前已有聚类中心之间的最小距离(即与最近的一个聚类中心的距离),再计算每个样本点被选为下一个聚类中心的概率 P,按照轮盘赌法选出下一个聚类中心点。

(3)重复步骤(2)直至选出 K 个聚类中心为止。

(4)对数据集中的每个样本点都计算它到 K 个聚类中心点的距离,并将该样本点划分到离它最近的聚类中心点。

(5)对每个类别重新计算聚类中心点。

(6)重复(4)、(5)两步直至聚类中心稳定。

K-Means++算法的距离度量公式为:

$$d(b,c) = 1 - R_{\mathrm{IOU}}(b,c) \tag{2}$$

其中,b 为预测的矩形框,c 为真实的矩形框,表示两个矩形框重叠度。IoU 的定义如图 3 所示。

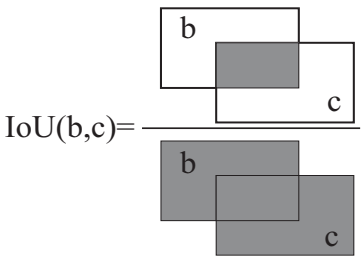

图 3 IoU 定义

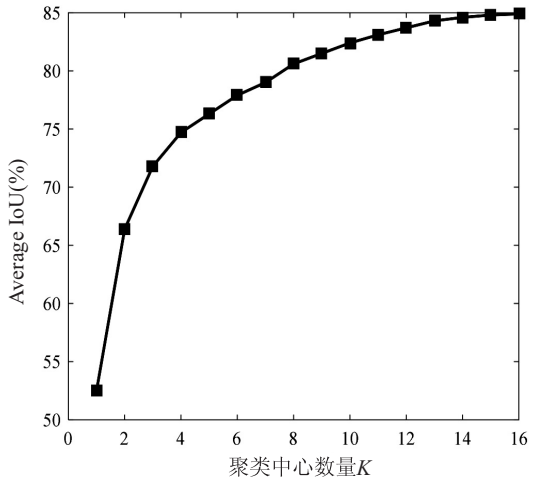

图 4 锚点框聚类分析结果

为了得到适合的锚点框数量,该文对平均 IoU 与锚点框数量的关系进行了分析。从图 4 中可以看出,刚开始随着聚类中心点数量的增加,平均 IoU 快速增长,当聚类中心点数量增加到 9 以后,平均 IoU 增速越来越慢,所以最终选择了 9 个锚点框聚类,并且在 3 个不同尺度上对这 9 组锚点框进行分配。其尺寸分别为(68,74),(142,138),(150,190),(210,190),(235,196),(260,225),(352,314),(355,317),(376,332),平均 IoU 与锚点框个数的关系如图 4 所示。

2.2 深度可分离卷积

标准的卷积在卷积时要同时考虑所有的通道数,所以会产生大量的参数计算。深度可分离卷积[15-16]是分别对每个通道做卷积,然后对每个通道上采集到的特征进行融合。假设有 $H \times W \times C$ 的输入,有 k 个3×3 的卷积,padding = 1,stride = 1,标准卷积的计算量:$H \times W \times C \times k \times 3 \times 3$,深度可分离卷积计算量:$H \times W \times C \times 3 \times 3 + H \times W \times C \times k$,计算量压缩为:

$$\frac{H \times W \times C \times 3 \times 3 + H \times W \times C \times k}{H \times W \times C \times 3 \times 3 \times k} = \frac{1}{k} + \frac{1}{9}$$

可以看出,深度可分离卷积在在输入参数相同的条件下,计算量大大减少,所以对 YOLOv3 的所有 stride=1 的 3×3 的卷积核使用深度可分离卷积模块来代替,加快模型计算速度。

2.3 SENet 嵌入

SENet[17-18]是由 Momenta 胡杰团队于 2017 年提出的网络结构,SENet 以显式地建模特征通道之间的相互依赖关系为切入点,通过学习的方式来自动获取每个特征通道的重要程度,然后按照这个重要程度去提升有用的特征并抑制对当前任务用处不大的特征。

SENet 主要包括两个过程,分别是 Squeeze 过程和 Excitation 过程。其具体过程如下:

(1)Squeeze 过程:相对于传统的卷积只是在一个局部空间上进行特征信息的提取,很难获得足够的信息来表征各个通道之间的关系,而 Squeeze 是在一个通道上对整个空间的特征进行提取,每一个特征通道经过 Squeeze 之后都会产生一个实数,这个实数是通过全局平均池化所产生的,因此在某种程度上这个值具有全局的感受野,计算公式如下:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} u_c(i,j) \tag{3}$$

(2)Excitation 过程:Squeeze 之后得到了图像的全局特征,再用 Excitation 提取各个通道之间的关系。基本思路就是通过 FC-Relu-FC-Sigmoid 操作生成每个通道的权重,然后对每个通道用对应的权重进行加权,实现在各个通道上对原始特征的重标定。计算公式如下:

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2\delta(W_1 z)) \quad (4)$$

$$x_c' = F_{scale}(u_c, s_c) = s_c \cdot u_c \quad (5)$$

SE 模块可以轻松地嵌入到所有主流的网络结构中,并提升原有模型的性能。如图 5 所示,将 SENet 嵌入到 Resent 网络中,可以看到只需要简单的几步即可实现网络优化,提高模型的性能。

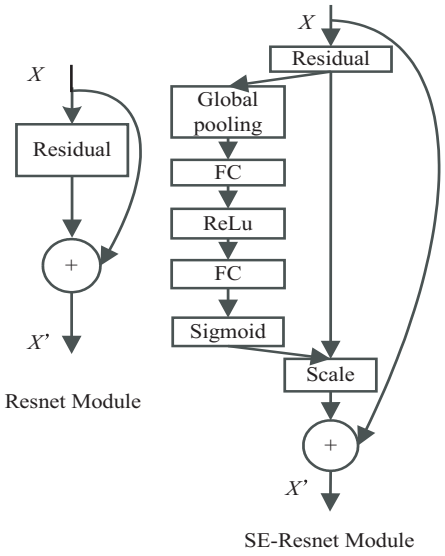

图 5　SENet 嵌入 Resnet 网络结构

提出将 SENet 与 Resnet[19] 相结合并引入到 YOLOv3 的多尺度预测网络中,实现对网络结构的优化,文中算法的网络结构如图 6 所示。

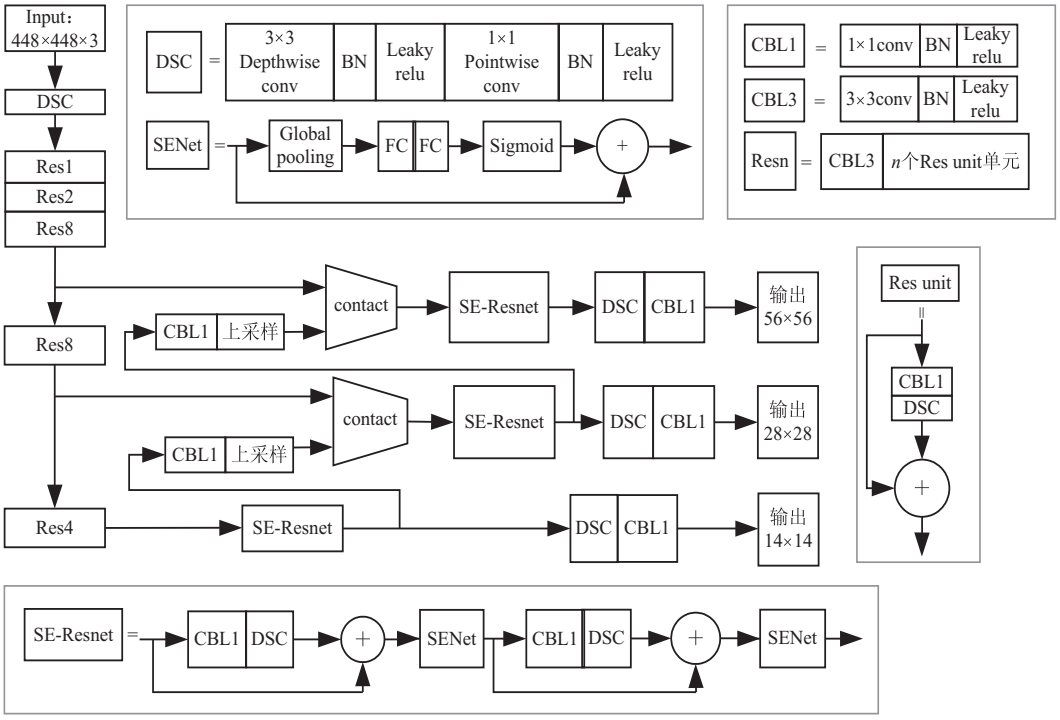

图 6　改进算法网络结构

2.4　损失函数改进

在 YOLOv3 中,损失函数由中心坐标误差、宽高坐标误差、置信度误差和分类误差四部分构成。其中,中心坐标误差、宽高坐标误差采用均方误差计算,置信度误差和分类误差采用交叉熵损失函数计算。在置信度误差计算中,IoU 的大小直接决定了预测框与真实框的相似度情况,对损失函数的计算有着重大影响,但是使用 IoU 时对于预测框和真实框不相交、两个边框的真实重叠情况等问题,无法有效做出判断。为了解决上述问题,该文使用 DIoU[20] 来替代 IoU,其中 DIoU Loss 的计算公式如下:

$$L_{DIou} = 1 - IoU + \frac{\rho^2(a, b)}{c^2} \quad (6)$$

$$d = \rho(a, b) \quad (7)$$

其中,IoU 在前文中已经定义过, a 为预测框的中心点, b 为真实框的中心点,是两框中心点的欧氏距离, c 是覆盖两框的最小封闭框的对角线长度, d 是预测框中心点和真实框中心点的距离,DIoU Loss 的示意图如图 7 所示。

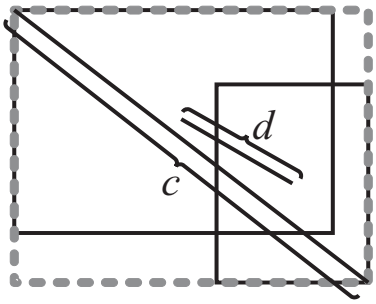

图 7 DIou Loss 示意图

2.5 非极大值抑制

在传统 NMS 中,IoU 指标常用于抑制冗余边界框,因为只考虑重叠区域的影响,所以对存在遮挡的情况经常产生错误的抑制。该文采用 DIoU-NMS[20] 算法,利用 DIoU 代替原来的 IoU,同时考虑了重叠区域和两个边界框的中心距离的影响,减少错误的抑制。计算公式如下:

$$s_i = \begin{cases} s_i & \text{IoU} - R_{\text{DIoU}}(M, B_i) < \varepsilon \\ 0 & \text{IoU} - R_{\text{DIoU}}(M, B_i) \geqslant \varepsilon \end{cases} \tag{8}$$

$$R_{\text{DIoU}} = \frac{d^2}{c^2} \tag{9}$$

其中,阈值可以取 0.43 到 0.48 的值, c 和 d 分别表示覆盖两框的最小封闭框的对角线长度和预测框中心点和真实框中心点的距离,在前面已经给出。

3 实验结果与分析

文中算法是在 pytorch 深度学习框架上实现,实验使用的配置为 NVIDIA GeForce GTX1070 GPU、16G 内存、操作系统为 Ubuntu16.04 的硬件平台上训练与检测。模型训练策略如表 1 所示。文中通过计算一张图片耗时、帧率、准确度与平均精度进行模型性能评估。

表 1 检测模型训练策略

参数	参数值
学习率	0.001
动量系数	0.9
权重衰减	0.000 5
训练轮数	120
批尺寸	12
数据增强	缩放、裁剪、旋转等

3.1 数据集

文中数据集是从 COCO 数据集获得,目标包括人、小汽车、卡车、自行车、客车 5 类道路常见的目标。训练集共计 12 750 张样本,其样本分布如图 8 所示。

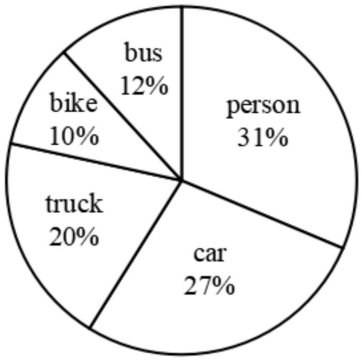

图 8 数据集样本分布

3.2 实验结果与分析

为了验证文中提出模型的性能,将改进算法与 YOLOv3 进行性能测试对比,结果见表 2 和表 3。通过对表格数据的分析可以发现改进之后的算法在均值平均精度(mAP)、5 类不同目标检测准确率和检测耗时上都较 YOLOv3 有了不同程度的提升,这反映出了文中算法改进的有效性。

表 2 改进算法与 YOLOv3 模型检测准确率对比

	YOLOv3	改进算法
mAP/%	70.2	74
Person	0.62	0.65
Car	0.64	0.68
Truck	0.75	0.79
Bus	0.86	0.88
Bike	0.64	0.70

表 3 改进算法与 YOLOv3 模型检测耗时对比

	FPS	Time/ms
YOLOv3	24	41.67
改进算法	26	38.46

4 结束语

基于 YOLOv3 算法提出了一种针对道路环境下常见目标检测的改进方法。首先,利用 K-Means++算法对数据集的锚点框重新聚类,得到新的聚类中心点。然后,对 YOLOv3 的 3×3 的卷积核进行深度可分离卷积,提高模型计算效率。其次,将 SENet 与 Resnet 相结合并引入到 YOLOv3 的多尺度预测网络,提高模型的性能。最后,改用 DIoU 代替 IoU 去判断真实框和预测框的相交情况。实验结果表明,该方法具有较好的鲁棒性,相对于 YOLOv3 算法,模型的均值平均精度(mAP)、5 类不同目标检测准确率均有所提升,检测耗时更短,提高了模型的检测性能。

参考文献:

[1] 赵建国,曹朝辉,梁　杰. 卷积神经网络 SSD 的道路目标检测[J]. 机械设计与制造,2020(6):181-184.

[2] 徐　融,邱晓晖. 一种改进的 YOLO V3 目标检测方法[J]. 计算机技术与发展,2020,30(7):30-33.

[3] 曹诗雨,刘跃虎,李辛昭. 基于 Fast R-CNN 的车辆目标检测[J]. 中国图象图形学报,2017,22(5):671-677.

[4] 李　珣,时斌斌,刘　洋,等. 基于改进 YOLOv2 模型的多目标识别方法[J]. 激光与光电子学进展,2020,57(10):101010.

[5] ROSS G,JEFF D,TREVOR D. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. Boston, MA, USA: IEEE, 2015: 705-713.

[6] ROSS G. Fast R-CNN[C]//Proceedings of the IEEEconference on computer vision and pattern recognition. Honolulu, HI,USA:IEEE,2017:6517-6525.

[7] 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 and Machine Intelligence,2017,39(6):1137-1149.

[8] REDMON J,DIVVALA S,GIRSHICK R,et al. You only look once:unified,real-time object detection[C]//Proceedings of 2016 IEEE conference on computer vision and pattern recognition. Las Vegas,USA:IEEE,2016:779-788.

[9] REDMON J,FARHADI A. YOLO9000:better,faster,stronger[C]//2017 IEEE conference on computer vision and pattern recognition (CVPR). Honolulu, HI, USA: IEEE, 2017: 6517-6525.

[10] REDMON J,FARHADI A. YOLOV3:an incremental improvement[C]//Proceedings of 2018 IEEE conference on computer vision and pattern recognition. Washington D. C, USA:IEEE,2018:1-6.

[11] LIU W,ANGUELOV D,ERHAN D,et al. Ssd:single shot multibox detector[C]//Computer vision - ECCV 2016. Amsterdam,The Netherlands:Springer,2016:21-37.

[12] 陈莉君,李　卓. 基于深度神经压缩的 YOLO 优化[J]. 计算机技术与发展,2019,29(12):72-75.

[13] LIN T Y,GOYAL P,GIRSHICK R. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,2017(99):2999-3007.

[14] 黄亚峰,何　威,吴光琴,等. 基于 K-means++和 LSTM 网络的光伏功率预测研究[J]. 电气自动化,2020,42(5):25-27.

[15] CHOLLET F. Xception:deep learning with depthwise separable convolutions[C]//2017 IEEE conference on computer vision and pattern recognition(CVPR). Honolulu,HI:IEEE,2017:1251-1258.

[16] 石陆魁,马红祺,张朝宗,等. 基于改进残差结构的肺结节检测方法[J]. 计算机应用,2020,40(7):2110-2116.

[17] HU J,SHEN L,SUN G. Squeeze-and-excitation Networks[C]//Proceedings of the 2018 IEEEcomputer society conference on computer vision and pattern recognition. Washington:IEEE,2018:7132-7141.

[18] 刘学平,李玮乾,刘　励,等. 嵌入 SENet 结构的改进 YOLOV3 目标识别算法[J]. 计算机工程,2019,45(11):243-248.

[19] HE K,ZHANG X,REN S,et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. Amsterdam, Netherlands:IEEE,2016:770-778.

[20] ZHENG Z H,WANG P,LIU W,et al. Distance-IoU loss:faster and better learning for bounding box regression[C]//Proceedings of the 34th AAAI conference on artificial intelligence. New York:AAAI,2020:12993-13000.