文章编号:1007-130X(2020)04-0649-09

基于深度卷积神经网络的小目标检测算法*

航^{1,2},朱 李 明1

(1. 中国科学院长春光学精密机械与物理研究所, 吉林 长春 130033; 2. 中国科学院大学, 北京 100049)

要:针对 YOLO 目标检测算法在小目标检测方面存在的不足,以及难以在嵌入式平台上达到实 摘 时性的问题,设计出了一种基于 YOLO 算法改进的 dense_YOLO 目标检测算法。该算法共分为 2 个阶 段:特征提取阶段和目标检测回归阶段。在特征提取阶段,借鉴 DenseNet 结构的思想,设计了新的基于 深度可分离卷积的 slim-densenet 特征提取模块,增强了小目标的特征传递,减少了参数量,加快了网络的 传播速度。在目标检测阶段,提出自适应多尺度融合检测的思想,将提取到的特征进行融合,在不同的特 征尺度上进行目标的分类和回归,提高了对小目标的检测准确率。实验结果表明:在嵌入式平台上,针对 小目标,本文提出的 dense_YOLO 目标检测算法相较原 YOLO 算法 mAP 指标提高了 7%,单幅图像检测 时间缩短了 15 ms,网络模型大小减少了 90 MB,明显优于原算法。

关键词:目标检测;嵌入式平台;小目标;深度卷积神经网络;多尺度预测 文献标志码:A 中图分类号:TP391.4 doi:10.3969/j.issn.1007-130X.2020.04.011

A small object detection algorithm based on deep convolutional neural network

LI Hang^{1,2}, ZHU Ming¹

(1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033; 2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract. In view of the shortcomings of YOLO object detection algorithm in small object detection, and the difficulty of achieving real-time performance on embedded platforms, this paper designs an improved YOLO object detection algorithm, called dense_YOLO. The algorithm contains two phases: feature extraction phase and object detection regression phase. In the feature extraction phase, based on the idea of DenseNet structure, a new slim-densenet feature extraction module based on deep separable convolution is designed, which enhances the transmission of small object features and reduces the parameter quantity to accelerate the network propagation speed. In the object detection stage, the idea of adaptive multi-scale fusion detection is proposed to fuse the extracted features, and the objects are classified and regressed on different feature scales, which improves the detection accuracy of small objects. Experimental results show that, compared with the original YOLO object detection algorithm, the dense_YO-LO object detection algorithm improves mAP by 7%, decreases the single picture detection time by 15 ms, and reduces the model size by 90 MB.

Key words: object detection; embedded platform; small object; convolutional neural network; multiscale prediction

收稿日期:2019-08-29;修回日期:2019-11-26

China

1 引言

随着计算机技术的迅速发展,计算机视觉技术 已经广泛应用于各行各业,在智能视频监控、人机 交互、机器人视觉导航、医学诊断和图像分类等领 域有着十分广阔的应用前景^[1-3]。目标检测是指 从一幅原始的图像中分离出背景和感兴趣的目标, 确定这一目标的类别和位置。目标检测技术是计 算机视觉中十分重要的组成部分,为计算机感知世 界、认知世界提供基础。

传统的目标检测算法主要基于图像梯度直方 图的局部特征(SIFT)和全局特征(HOG)提取方 法^[4,5],利用滑动窗口找出目标的潜在区域,提取 出特征信息,再利用支持向量机(SVM)和自适应 提升(Adaboost)等分类器对目标特征进行分类。 还有一些算法使用全局注意力机制在整个检测场 景中选择突出的候选区域。虽然这些算法检测效 果较好,但鲁棒性较差,无法用于不同的应用场景 中。而且由于滑动窗口策略会导致大量重复计算, 算法的执行速度较慢。

近年来,基于深度卷积神经网络[6-8]的目标检 测技术以其局部感知和权值共享的特点,在目标检 测领域取得了飞速的发展。与传统的目标检测算 法相比,基于深度卷积神经网络的目标检测算法通 过卷积神经网络对训练数据集的学习,可以自动地 从图像中提取特征,获得目标的位置和类别,学习 能力强、精度高、鲁棒性强。当前基于卷积神经网 络的目标检测算法主要分为2类:一类是基于区域 的目标检测算法,如 Faster RCNN^[2]和 Corner-Net^[9]。这类算法将特征提取、区域建议和边界框 分类与回归整合到1个网络中,提高了小目标的检 测率,但是在嵌入式平台上无法满足实时检测的要 求。另一类是基于回归的目标检测算法,如 YO-LO(You Only Look Once)^[10-12] 和 SSD(Single Shot multibox Detector)^[13],只需经过1次卷积运 算,便可直接在原始图像上完成目标检测工作。这 类算法检测速度较快,满足实时性的要求,但是对 小目标的检测效果不尽人意。

为了在嵌入式平台上实现小目标检测精度和 检测速度方面的良好平衡,本文借鉴 YOLO 算法, 设计了一种新的 dense_YOLO 目标检测算法。在 特征提取阶段,通过将 DenseNet^[14] 网络的思想和 深度可分离卷积^[15,16] 思想相结合,设计出基于深 度可分离卷积的 slim-densenet 特征提取网络,增 强了小目标的特征传递,也加快了算法的检测速 度。在检测阶段,采用自适应多尺度融合检测的思想,在不同的特征尺度上进行目标的分类和回归, 提高了小目标的检测精度。dense_YOLO目标检 测算法可以为深度学习^[17]目标检测技术应用于嵌 入式平台提供可能。

2 dense_YOLO 目标检测算法

2.1 原始 YOLO 算法

YOLO 算法将物体检测问题处理成回归问题,用1个神经网络结构就可以从图像中直接预测目标的位置和类别。在网络结构上,首先采用 Re-LU(Rectified Linear Unit)^[6]激活函数和卷积操作来提取特征,之后采用全连接层进行目标位置和类别的预测,从而得出最终的检测结果。

YOLO算法采用分块思想,将输入图像分成 S×S个小块(grid cell),若目标的中心点落入到某 一个小格中,那么这个小格便负责这个目标的预测 任务,通过网络的训练学习数据分布,预测出这个 目标的中心坐标、长宽和类别;再经过非极大值抑 制 NMS(Non-Maximum Suppression)算法^[18],将 置信度高的候选框作为最终的目标位置输出,如图 1 所示。虽然 YOLO 算法检测速度较快,但全连 接层丢失了较多的空间信息,使得算法对小目标的 检测效果不好。



Figure 1 YOLO detection principle 图 1 YOLO 检测原理

2.2 dense_YOLO 目标检测原理

本文针对 YOLO 算法在检测小目标时精度低的问题,设计了新的 dense_YOLO 目标检测算法, 算法由 2 部分组成:特征提取模块和多尺度目标检 测模块。dense_YOLO 目标检测算法舍弃了原始 YOLO 算法中的全连接层,改为采用全卷积网络 (Fully Convolutional Networks)^[19]的结构,使得 网络可以适应不同大小的输入图像。然后使用新的基于深度可分离卷积的 slim-densenet 特征提取 网络来提取特征,新的网络结构参数量更少,特征 在网络中的传递速度更快。最后采用自适应多尺 度融合算法进行目标检测,在不同的特征尺度上进 行目标的分类和回归,得到目标的具体位置和类别 信息,提高了算法对小目标的检测准确率。

2.2.1 slim-densenet 特征提取模块

本文借鉴 DenseNet 网络结构,设计了 slimdensenet 特征提取网络结构。原始的 YOLO 网络 结构中,特征的传递方式为逐层传递,每层的输入 只来 自和 它相连的前一层。而本文的 slimdensenet 特征提取网络结构改变了特征在网络层 之间的传递方式,使得特征可以跳过部分网络层直 接传递到后面的网络层中,不仅减轻了梯度消失的 问题,而且还加快了特征在网络中的传递。本文的 slim-densenet 特征提取网络结构如图 2 所示。其 中 Dense block 模块是 slim-densenet 网络中加速 特征传递的部分。从图 3 可以看出,特征经过 Dense block 可以跳过其中的部分网络直接传递到 后端网络,因而加快了特征在网络中的传递。

为了进一步提高特征在网络中的传递速度,本 文将特征提取模块中的7×7,5×5和3×3卷积替 换为深度可分离卷积。深度可分离卷积区别于传 统的卷积:传统的卷积利用卷积核对输入的多通道 图像进行卷积操作,随着卷积核的增大,计算量成 指数增长。而深度卷积神经网络将卷积操作分为 Depthwise Convolution 和 Pointwise Convolution 2部分,如图4所示,Depthwise Convolution 对输 入的多通道向量进行分层卷积,即对每一层通道都 进行卷积运算;之后 Pointwise Convolution 使用 大小为1×1的卷积核对 Depthwise Convolution 得到的中间结果在通道维度进行扩充,最终得到和 传统卷积操作相同的结果。与传统的卷积操作相 比,深度可分离卷积在减少参数量的同时加快了特 征的传递。



Figure 4 Depthwise separable convolution 图 4 深度可分离卷积

2.2.2 自适应多尺度融合检测模块

针对小目标检测这一难点,本文提出了多尺度 融合检测的思想。基于深度学习的 YOLO 目标检 测算法需要经过多个下采样层来提取特征,在最后 一层特征图上进行目标的分类和回归,而每经过1 个下采样层,网络的分辨率就会被压缩1次,从而 损失不同分辨率的特征图的信息,这样使得网络在 最后很难提取到针对小目标的特征,因此 YOLO



Figure 2 Network structure of slim-densenet 图 2 slim-densenet 特征提取网络结构

算法对小目标的检测效果不尽人意。针对这一情况,本文在3个特征尺度上进行目标检测,每1个特征尺度都融合了其他特征尺度的信息,网络浅层提取目标边缘和纹理等细节特征,网络深层提取物体轮廓的特征,将它们相互融合之后进行卷积运算,提取出目标的位置类别信息。本文的自适应多尺度融合检测流程如图5所示。



Figure 5 Adaptive multi-scale fusion detection 图 5 自适应多尺度融合检测

首先通过最后 3 个 Dense block 模块提取出 3 个尺度不同的 feature map。接着进行特征融合, 将小尺度的 feature map 包含的特征传入前面一层 大尺度的 feature map 中,这样每一层预测所用的 feature map 都融合了不同分辨率、不同语义强度 的特征,大尺度 feature map 由于其分辨率较高,保 留了较多的细节特征,更容易预测小的预测框。小 尺度 feature map 由于其分辨率较低,保留的是物 体的轮廓等信息,更容易预测大的预测框。然后在 特征融合之后的 3 个 feature map 上进行卷积运 算,提取出目标的坐标信息和类别信息。最后将 3 个特征尺度上提取到的目标信息相结合,采用非极 大值抑制 NMS 算法得到最终的检测结果。

2.3 dense_YOLO 目标检测流程

dense_YOLO 目标检测算法分为 3 部分。算 法首先通过基于深度可分离卷积的 slim-densenet 模块进行特征的提取,之后将提取到的特征在不同 尺度上进行融合和检测,最后采用非极大值抑制 NMS算法将不同尺度上提取到的目标位置和类别 信息进行结合,得到最终的检测结果。算法的基本 流程如下:

(1)图像初始化。将输入图像进行初始化,转 换成大小为 608×608 的彩色图像送入网络。

(2)特征提取。使用 slim-densenet 特征提取 模块对送入的图像进行特征提取,将结果送入后续 的目标检测模块。 (3)自适应多尺度融合检测。使用多尺度融合检测算法对特征进行目标的位置和类别预测。

(4)检测结果输出。通过 NMS 算法计算出不同尺度特征中目标的具体位置和类别,并在原始图像上进行标注,最终输出结果。

算法的流程图如图 6 所示。



Figure 6 Flow chart of dense_YOLO algorithm 图 6 dense_YOLO 算法流程

3 数据集与实验设计

3.1 数据集和评价指标

本文在 MS COCO(Microsoft COCO: Common Objects in Context)^[20]通用数据集上进行了 测试,也在本文制作的小目标数据集上和原算法做 了对比实验。

目标检测领域主要有 2 大通用数据集,一个是 PASCAL VOC 数据集,另一个是 MS COCO 数据 集。MS COCO 数据集是微软构建的一个数据集, 其包含 detection、segmentation 和 keypoints 等任 务。在检测方面,MS COCO 包含 80 种类别,相对 于 PASCAL VOC 数据集的 20 种类别,MS COCO 数据集类别丰富,包含了自然图像和生活中常见的 目标图像,背景比较复杂,目标数量比较多,目标尺 寸更小,因此在 MS COCO 数据集上的实验更难。

本文制作数据集的方法是使用无人机平台,对 三亚海滩附近场景进行视频拍摄,着重选取一些陆 地上和海滩上的行人进行拍摄,拍摄时光线充足, 无雨雾雪,温度 27℃。对得到的视频每隔1s提取 1 帧,最终得到 270 幅尺寸为 1920×1080 的彩色 图像。采用开源的 Labelimage 软件对采集到的 270 幅图像进行标记,对应得到 270 个.xml 文件。 对其中的 210 幅图像进行数据增强,通过各种变换 增加训练的数据量,增加噪声数据,防止网络模型 过拟合情况的出现,以得到能力更强的网络,更好 地适应应用场景。本文主要使用的数据增强方法 有:旋转、缩放、裁剪,得到近 2 100 幅图像,大小为 1000×600。这 2 100 幅图像转换成 voc 数据集的 格式作为训练样本。剩余的 60 幅图像作为验证 集。

本文采用的主要评价指标是 AP,因为目标检 测领域中 Precision 和 Recall 是一对对立关系的指 标,一个指标升高伴随着另一个指标下降,故人们 采用 PR 曲线,即 Precision 和 Recall 曲线的线下 面积来作为评价一个算法性能的指标 AP,对于多 个类别,AP 的平均值称为 mAP。AP@0.5 的含 义为:将交并比 IoU(Intersection over Union)阈值 设置为 0.5,与真值框的 IoU大于 0.5 的检测框判 定为检测正确,通过在召回率坐标轴上每隔 0.1 计 算出对应的准确率值,绘制出 PR 曲线,最终得到 检测指标 AP。其中 IoU 指预测结果框和真值框 重合区域的面积与总面积的比值。同时还有针对 小目标、中目标和大目标的 mAP 以及在 IoU 改变 时各个 Recall 的值。另一个评价指标为 TOP-1, 即置信度最高的检测结果的准确率。

3.2 训练参数的设置

本文使用的训练参数及其值如下所示:

batch:每次迭代训练的图像数目,本文设置为 32。

subdivisions:将 *batch* 进行分组后送入网络, 本文设置为 16。

网络输入尺寸:本文设置为 608×608。

angle:图像角度变化,单位为度,以增加训练 样本数,本文设置为7°。

weight decay:权值衰减正则项系数,防止过 拟合,本文设置为 0.000 5。

saturation:饱和度,以增加训练样本数,本文 设置为 0.75。

exposure:曝光度,以增加训练样本数,本文设置为 0.75。

hue:色调变化范围,以增加训练样本,本文设置为 0.1。

learning rate:学习率,本文设置为 0.001。

max_batches:最大迭代次数,本文设置为 500 000。

policy:学习率调整策略,本文选择 *poly*策略。*poly*的表达式为 *new*<u>Ir</u> = *base*<u>Ir</u> *(1 – *iter/maxiter*)^{*power*},可以看出学习率主要由 *power*参数控制,*power*<1,学习率为凸状,*power*>1,学习率为凹状。

anchors:预测框大小的预设值,具体如下:

 $(10 \times 13), (16 \times 30), (33 \times 23), (30 \times 61),$ $(62 \times 45), (59 \times 119), (116 \times 90), (156 \times 198),$ $(373 \times 326)_{\circ}$

k:anchor的数目,本文设置为9。

本文采用 kmeans_YOLO 聚类来处理数据 集,不同 k 值对应的损失如图 7 所示。



图 7 数据集聚类结果

3.3 实验环境和训练方法

实验环境硬件配置为:训练时采用计算机服务器,配置 core i7 处理器,2 块 Titan xp 显卡。检测时采用 NVIDIA JETSON XAVIER^[21]嵌入式平台,为 NVIDIA 公司为深度学习专门设计的嵌入式系统,内置 CPU 和 GPU 模块,默认安装 ubuntu 系统。

网络在每个卷积层前都加入了 BN(Batch Normalization)层,将每一层的神经网络中的任一 个神经元的输入进行归一化,分布为均值为 0、方 差为1的标准正态分布,解决了训练过程中,随着 网络层数的加深,输入数据的分布逐渐发生偏移, 从而导致的训练收敛慢和梯度消失问题。

本文使用交叉熵损失函数替换 YOLO 算法使 用的 Softmax 函数。YOLO 算法中的损失函数将 1 个实数的 k 维向量压缩成另 1 个 k 维向量,使得 所有分量的范围在(0,1),并且分量的和为 1。但 是,在实际的分类任务中,检测到的目标可能会属 于多个类别,比如在 Open Images Dataset 中,1 个 目标有多个标签(例如人和女人),故本文使用交叉 熵损失函数来替代 Softmax 函数,使得网络预测的 目标可以有多个不同的类别与之对应。 训练时采用的是深度学习经常使用的随机梯 度下降算法,从所有训练数据里随机选择1个 batch 作为样本,送入网络学习,接着再随机选择样 本来进行学习,最终达到最优点。

和大部分的目标检测算法一样,本文首先使用 slim-densenet 分类网络在 ImagNet 竞赛数据集上 进行预训练,得到预训练权重。使得初始的网络学 习到类别的大致特征,为之后的检测提供基础。接 着在 backbone 网络上添加自己的检测网络,在通 用的 MS COCO 数据集上进行训练,经过 10 天 500 000 次迭代之后,损失稳定在 3.5 左右不再变 化,结束训练。损失值与迭代次数的关系如图 8 所 示。



Figure 8 Loss function visualization 图 8 损失函数可视化

4 实验结果对比与分析

本文基于 slim-densenet 的网络模型在 ImageNet 分类数据集上的表现如表 1 所示。

Table 1	Netw	ork	classification	result
	表 1	网络	各分类结果	

Backbone model	TOP-1	CPU Time/s
VGG-16	70.5	4.36
Darknet19	72.9	0.87
ResNet 101 ^[22]	77.1	2.23
slim-densenet(本文)	77.0	1.38

从表1中可以看出,在TOP-1准确率方面,本 文基于 slim-densenet 的网络模型比 VGG-16 和 Darknet19 提高了7%,和经典的 ResNet 101 相差 无几,但是在速度方面有着明显的优势。

在通用的目标检测数据集 MS COCO 上,硬件 平台为 NVIDIA JETSON XAVIER,将 dense __ YOLO-416 算法和 YOLO-608、SSD512 等经典算 法进行了比较,其中 dense __YOLO-416、YOLO-608、SSD512 中的 416,608,512 分别代表网络的输 入尺寸大小。具体的检测性能和检测结果如表 2 和图 9 所示,其中 APs 表示对小目标(像素小于 32×32)检测的 AP 指标;AP₅₀即 AP@0.5 是 3.1 节中的 AP@0.5 的简写形式;AP 为将 IoU 阈值 从 0.5~0.95 中每隔 0.5 进行取值,计算出的 10 个 AP 值求取平均所得;Weights 为网络模型的权 重大小;GPU Time 表示算法检测单幅图像所用时 间。

Table 2 Detection results on MS COCO 表 2 MS COCO 上的检测结果

算法	$AP_{\rm S}$	AP	AP_{50}	Weights /MB	GPU Time/s
YOLO-608	7.2	25.4	48.1	259	0.060 5
SSD512	9.0	26.8	46.5	230	0.0989
Faster RCNN	7.7	24.2	45.3	306	0.165 5
dense_YOLO-416(本文)14.9	36.4	59.4	160	0.045 4

在本文创建的目标检测数据集上,硬件平台为 NVIDIA JETSON XAVIER,本文设计的 dense_ YOLO 算法检测性能如表 3 所示,检测结果如图 10 所示。

在 NVIDIA JETSON XAVIER 嵌入式平台 上,本文的 dense_YOLO 目标检测算法相较于原 YOLO 算法,检测时间缩短了 15 ms 左右,可以达 到 20 fps 以上的实时性,也优于其他的算法。在 检测准确率方面,针对 MS COCO 数据集中的小目 标,本文设计的 dense_YOLO 算法相较于 YOLO 算法在 AP_s 指标上提高了 7%,而在 $AP_{0.5}$ 指标 上, dense_YOLO 算法超过了 YOLO、SSD 和 Faster RCNN 算法,都至少提高了 10%,并且网络模 型大小为原 YOLO 算法的 66%。针对本文创建 的数据集,dense_YOLO 算法也比原 YOLO 算法 的 $AP_{0.5}$ 提高了 8%。

5 结束语

本文提出了一种新的基于深度卷积神经网络的 dense_YOLO 目标检测算法,在嵌入式平台上可以实现高准确率的实时目标检测。dense_YO-LO 算法首先利用新提出的 slim-densenet 特征提取模块对图像提取特征,减少了参数量,加快了网络的训练和预测。然后采用自适应多尺度融合检测方法丰富了小尺度特征图的信息,提高了小目标的检出率。最后通过非极大值抑制算法得到最终检测结果。实验表明,在通用的 MS COCO 数据集上, dense_YOLO算法在小目标检测方面,较原

1-1 11-11 d test 4 c test 3 THE WARN IS I THE PHONE e test 5 the direct has dire hik f test 6

 Figure 9
 YOLO algorithm and dense_YOLO algorithm detection results on MS COCO dataset

 图 9
 原算法(左)和改进算法(右) 在 MS COCO 数据集上检测结果

Table 3	Dete	ction	results	on	dataset	created	in thi	s paper
	表 3	在本	文创建	ま数	据集上	的检测	结果	

算法	AP_{50}	GPU Time/s
YOLO-608	67.3	0.060 5
SSD512	55.5	0.098 9
Faster RCNN	63.3	0.165 5
dense_YOLO-416(本文)	75.4	0.045 4

YOLO算法 mAP 指标提高了 7%,检测时间缩短 了 15 ms,网络模型大小为原算法的 66%。此外, 在本文采集并手工标注的数据集上,该算法检测准 确率和速度也明显优于原算法,满足嵌入式平台对 算法速度和精度的要求,可为深度学习在嵌入式平 台上的应用提供借鉴。本文提出的算法还有很多 地方需要改进,比如 slim-densenet 还可以通过剪 枝操作,减少通道的冗余,加快推断速度,在本文的 模型上进行压缩仍然需要更进一步的研究。

参考文献:

- Xiong Chang-zhen, Shan Yan-mei, Guo Fen-hong. Image retrieval method based on image principal part detection[J].
 Optics and Precision Engineering, 2017, 25(3), 792-798. (in Chinese)
- [2] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards realtime object detection with region proposal networks [C] // Proc of International Conference on Neural Information Processing Systems, 2015:91-99.
- [3] Li Y,Qi H,Dai J,et al. Fully convolutional instane-aware semantic segmentation[C]//Proc of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 4438-4446.
- [4] Cutter G, Stierhoff K, Zeng J. Automated detection of rockfish in unconstrained underwater videos using Haar cascades





and a new image dataset: Labeled fishes in the wild[C] // Proc of 2015 IEEE Winter Applications and Computer Vision Workshops,2015:57-62.

- [5] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]// Proc of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR),2005:886-893.
- [6] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C] // Proc of the 25th International Conference on Neural Information Processing Systems, 2012, 1097-1105.
- [7] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] // Proc of 2014 IEEE Conference on Computer Vision

and Pattern Recognition(CVPR), 2014:580-587.

- [8] Li Lin-hui, Lun Zhi-mei, Lian Jing, et al. Convolution neural network-based vehicle detection method[J]. Journal of Jilin University (Engineering and Technology Edition), 2017, 47 (2);384-391. (in Chinese)
- [9] Law H.Deng J. CornerNet: Detecting objects as paired keypoints[C] // Proc of the European Conference on Computer Vision (ECCV), 2018; 734-750.
- [10] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection [C] // Proc of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2016.779-788.
- [11] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger
 [C] // Proc of IEEE Conference on Computer Vision and

- [12] Redmon J, Farhadi A. YOLOv3: An incremental improvement[J]. arXiv:1804.02767,2018.
- [13] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector[C]// Proc of European Conference on Computer Vision, 2016: 21-37.
- [14] Huang G.Liu Z, van de Maaten L, et al. Densely connected convolutional networks [C] // Proc of IEEE Conference on Computer Vision and Pattern Recongnition, 2017: 4700-4708.
- [15] Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications [J]. arXiv:1704.04861,2017.
- [16] Sandler M. Howard A. Zhu M. et al. MobileNetv2: Inverted residuals and linear bottlenecks[C] // Proc of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).2018:4510-4520.
- [17] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015,521(7553):436-444.
- [18] Neubeck A, van Gool L. Efficient non-maximum suppression[C]//Proc of the 18th International Conference on Pattern Recognition (ICPR'06),2006;850-855.
- [19] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation [C] // Proc of 2015 IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3431-3440.
- [20] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common objects in context[C] // Proc of European Conference on Computer Vision, 2014;740-755.

- [21] Deep learning AI developer [EB/OL]. [2019-07-13]. https://www. nvidia. com/en-us/deep-learning-ai/developer/.
- [22] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition(CVPR),2016;770-778.

附中文参考文献:

- [1] 熊昌镇,单艳梅,郭芬红.结合主体检测的图像检索方法[J].
 光学精密工程,2017,25(3):792-798.
- [8] 李琳辉,伦智梅,连静,等.基于卷积神经网络的道路车辆检 测方法[J].吉林大学学报(工学版),2017,47(2):384-391.

作者简介:



李航(1994-),男,吉林长春人,硕士 生,研究方向为计算机视觉。E-mail: 806184365@qq.com

LI Hang, born in 1994, MS candidate, his research interest includes computer vi-



朱明(1964-),男,江西南昌人,硕士, 研究员,研究方向为图像处理。E-mail: zhuming1987@qq.com

ZHU Ming, born in 1964, MS, research fellow, his research interest includes image

processing.