

基于改进型 YOLO v3 的蔬菜识别算法

魏宏彬, 张端金, 杜广明, 肖文福

(郑州大学 信息工程学院, 河南 郑州 450001)

摘 要: 针对超市的散装蔬菜区排队称重问题(称重设备能够自动识别蔬菜种类将有效地提高超市的运行效率)提出一种基于改进型 YOLOv3 的蔬菜识别方法。首先,利用高清摄像头以及网络爬虫技术采集蔬菜图片;其次,通过 K-means 聚类分析得到 15 组适应于蔬菜数据集的先验框;再次,采用一种新的边界框回归损失函数 $DIoU$ 来提高检测任务的精度;最后,因蔬菜数据集中的大目标较多,通过增强特征提取网络,获取 5 组不同尺度的特征构成特征金字塔从而实现蔬菜识别任务。改进型 YOLOv3 算法在测试集上的平均精度 mAP 达到 93.2%,识别速度是 $35 \text{ f} \cdot \text{s}^{-1}$ 。该方法在保证实时检测目标的同时提升了识别的平均精度。

关键词: 蔬菜识别; K-means; 卷积神经网络; 特征金字塔; YOLOv3

中图分类号: TP18

文献标志码: A

doi: 10.13705/j.issn.1671-6833.2020.03.002

0 引言

目前,目标检测主要包括传统目标检测技术和深度学习目标检测技术。传统的识别技术主要利用滑动窗口实现图片内目标的搜索,然后通过 HOG(histogram of oriented gradient)或者 SIFT(scale-invariant feature transform)手工设计特征输入到 SVM^[1]内进行识别。多尺度形变部件模型(DPM)^[2]的提出实现了目标检测。但传统的检测方法不仅检测精度较低,而且鲁棒性较差。

Girshick 等^[3]采用候选区域和卷积神经网络结合的目标检测方法取代了传统方法。目前,基于深度卷积神经网络经典的目标检测算法主要有两大类。一类是两阶段目标检测方法,例如 Faster RCNN^[4]、R-FCN^[5]和 FPN^[6]。该类算法第一阶段对前景和背景进行分类以及边界框回归,第二阶段利用第一阶段的输出再进一步地分类和边界框回归。另一类是一阶段目标检测算法,例如 YOLO 系列算法^[7-9]和 SSD 算法^[10]。与两阶段算法相比,该类算法利用深度卷积神经网络提取的特征直接对目标进行识别和定位,满足了实时性的要求,但识别精度较低。在实际应用中,目标检测既要保证精度又要满足实时性的要求,所以一

阶段目标检测算法使用范围较广。其中 YOLOv3 算法^[9]在一阶段目标算法中检测效果最佳,在 COCO 数据集上 51 ms 内 mAP 为 57.9%。

为提高超市的运行效率,使称重设备能够自动检测蔬菜的种类,从而快速地为散装蔬菜称重打码,笔者采用高清摄像机和网络爬虫技术收集蔬菜数据,并对数据进行筛选和标注,制作成用于检测的蔬菜数据集。为保证能实时地检测蔬菜目标,笔者以 YOLOv3 算法为基础模型。由于 YOLOv3 算法是针对 COCO 数据集的优化,为适应新的数据集,笔者提出了改进型 YOLOv3 算法。首先,通过 K-means 算法^[11]分析获得适应于蔬菜数据的 15 组先验框;其次,采用一种新的边界框回归损失函数 $DIoU$ 来提高检测性能;最后,因蔬菜数据集中的大目标较多,因此通过增强特征提取网络,获取 5 组不同尺度的特征构成特征金字塔,从而实现蔬菜检测任务。经实验验证,改进型 YOLOv3 算法对蔬菜数据具有较好的检测性能。

1 YOLOv3 原理

1.1 Darknet-53 网络

YOLOv3 算法性能得到提升的主要原因为特征提取网络的增强。与 YOLOv2^[8]的 DarkNet-19

收稿日期: 2019-06-29; 修订日期: 2019-09-21

基金项目: 国家自然科学基金资助项目(61471323)

通信作者: 张端金(1966—),男,湖北荆州人,郑州大学教授,博士,主要从事智能信号处理、故障检测与滤波, E-mail: djzhang@zzu.edu.cn。

网络相比, YOLOv3 借鉴了 ResNet^[12] 的残差结构, 构造出更深的 DarkNet-53 网络。DarkNet-53 特征提取网络由 3×3 和 1×1 卷积层构成, 采用 5 个步长为 2 的 3×3 卷积层替换最大池化层(max pooling)实现下采样。该网络在 ImageNet 数据集测试, 网络性能比 ResNet 网络更加有效, 结果如表 1^[9]。表中 p_{Top-1} 和 p_{Top-5} 分别是模型进行图片识别时得到前 1 个结果和前 5 个结果中有 1 个是正确的概率; 计算量表示浮点运算的次数; 运算速度是每秒多少次浮点运算; 帧速率为每秒刷新图片的帧数。

表 1 特征提取网络

Table 1 Feature extraction network

Backbone	$p_{Top-1} / \%$	$p_{Top-5} / \%$	计算量/次	运算速度/(次·s ⁻¹)	帧速率/(f·s ⁻¹)
Darknet-19	74.1	91.8	7.29×10^9	1.246×10^{12}	171
ResNet-101	77.1	93.7	1.97×10^{10}	1.039×10^{12}	53
ResNet-152	77.6	93.8	2.94×10^{10}	1.090×10^{12}	37
Darknet-53	77.2	93.8	1.87×10^{10}	1.457×10^{12}	78

由表 1 可知, DarkNet-53 网络的速度约是 ResNet-101 网络的 1.5 倍, 并且与 ResNet-152 网络性能几乎一样; 同时, 帧速率提高至 $78 \text{ f} \cdot \text{s}^{-1}$ 。因此 DarkNet-53 在满足检测实时性的同时比 DarkNet-19 具有更高的准确率。

1.2 YOLOv3 设计思想

FPN 算法通过高低层特征融合的方式增强低层特征的语义信息, 从而提高小目标的检测准确率^[6]。YOLOv3 采用了类似的方法, 利用特征金字塔预测不同尺度的目标; 同时, YOLOv3 将 Softmax^[13] 替换为多个独立逻辑分类器, 使用二元交叉熵损失函数进行类别预测。因此, YOLOv3 与其他主流检测网络相比, 无论在速度还是检测准确率上, 均表现出优异的检测性能。

2 改进型 YOLOv3

相比于其他检测网络, YOLOv3 具有速度快、精确度高等优点。对于蔬菜数据集, 因数据之间的差异需重新设计特征提取网络。笔者受 DenseNet^[14] 启发, 将 DenseBlock 结构^[14] 添加到 DarkNet-53 网络中, 提高了检测大目标的性能。此外, 笔者还提出一种新的边界框回归损失函数 $DIoU$, $DIoU$ 与 YOLOv3 的边界框回归损失函数 MSE 相比, 定位精度更高。

2.1 特征提取网络

若在 YOLOv3 算法中输入的图片尺寸为

512×512 , 经过下采样 32 倍后输出特征图的尺寸是 16×16 。根据 FPN 可知, 原图片被下采样了 64 倍, 这对于提高大目标的检测精度起到一定的作用。针对 YOLOv3 算法存在的特征图尺寸偏大、感受野偏小、预测不准确等问题, 笔者在 DarkNet-53 的基础上加入 2 个 DenseBlock 来构造深度卷积神经网络以提高检测精度。单个 DenseBlock 结构如图 1 所示。

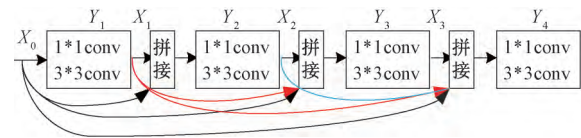


图 1 DenseBlock 结构

Figure 1 DenseBlock structure

图 1 中 X_0 是输入特征, 第 n 层的输入不但与第 $n-1$ 层的输出特征有关, 而且也与之前所有卷积层的输出有关, 其定义的公式如下:

$$X_n = Y_n([X_0, X_1, \dots, X_{n-1}]), \quad (1)$$

式中: $[]$ 表示将第 0 层的输出 X_0 到第 $n-1$ 层输出 X_{n-1} 进行通道拼接后作为第 n 层的输入; 非线性变换 Y_n 表示将第 n 层输入经过批量归一化、ReLU 激活函数和卷积层之后得到输出特征 X_n 。DenseBlock 内每个卷积层由 1×1 和 3×3 卷积构成。 1×1 卷积主要是为了减少特征通道的数量、降低计算量以及融合特征各通道之间的信息。此外, 1×1 卷积核的数量是 $4 \times k$, k 表示 3×3 卷积核的数量, 称为增长率, 在文中设置 $k = 48$ 。

在 DarkNet-53 网络的后面添加 2 个 DenseBlock 结构达到加深网络的目的。而不使用残差单元的主要原因为: 残差单元相比于 DenseBlock 而言参数量较高。例如, 若输入的特征通道数是 2 048 个, 残差单元 1×1 和 3×3 的卷积核的数量分别是 1 024 个和 2 048 个; 而 DenseBlock 结构中 1×1 和 3×3 的卷积核的数量分别是 192 个和 48 个。此外, 根据 DSOD^[15] 网络结构可知, 在没有预训练的前提下 DenseBlock 结构更易于训练, 这主要源于该结构密集连接的方式。改进型 YOLOv3 特征提取网络详细结构如图 2 所示。

图 2 中每个 DenseBlock 采用 4 组 1×1 和 3×3 卷积构成, 每组的输入均由前面所有组的输出经过通道拼接获得。若输入图片是 512×512 像素, 则由残差单元和 DenseBlock 输出的且用于检测的特征图尺寸分别是 64×64 、 32×32 、 16×16 、 8×8 和 4×4 。为增强每个特征图的上下文语义信息, 提高特征的表达能力, 改进

Type	Filters	Size	Output	
Convolutional	32	3×3	512×512	
Convolutional	64	3×3/2	256×256	
Convolutional	32	1×1		×1
Convolutional	64	3×3	256×256	
Residual			256×256	
Convolutional	128	3×3/2	128×128	
Convolutional	64	1×1		×2
Convolutional	128	3×3	128×128	
Residual			128×128	
Convolutional	256	3×3/2	64×64	
Convolutional	128	1×1		×8
Convolutional	256	3×3	64×64	
Residual			64×64	
Convolutional	512	3×3/2	32×32	
Convolutional	256	1×1		×8
Convolutional	512	3×3	32×32	
Residual			32×32	
Convolutional	1024	3×3/2	16×16	
Convolutional	512	1×1		×4
Convolutional	1024	3×3	16×16	
Residual			16×16	
Convolutional	2048	3×3/2	8×8	
Concatenation			8×8	×4
Convolutional	192	1×1		
Convolutional	48	3×3		
Convolutional	2240	3×3/2	4×4	
Concatenation			4×4	×4
Convolutional	192	1×1		
Convolutional	48	3×3		

图 2 改进的特征提取网络结构

Figure 2 Improved feature extraction network structure

型 YOLOv3 利用以上 5 组特征通过最近邻插值法上采样来构造特征金字塔,从而共享高低层特征信息,进行目标检测。

2.2 DIoU 边界框回归损失函数

目标检测、目标跟踪以及图像分割都需要依靠精确的边界框回归,现有的网络大多采用 L_1 和 L_2 范数作为边界框回归损失函数,却忽略了真实框和预测框之间的交并比 IoU 在边界框回归中的重要性。 IoU 定义如下。

$$IoU = \frac{|P \cap G|}{|P \cup G|}, \quad (2)$$

式中: P 是目标的预测框; G 是目标的真实框; IoU 是真实框和预测框面积之间的交并比。

在目标检测中 IoU 是比较预测框与真实框之间最常用的度量标准,是评价网络性能的重要指标。通过证明可知,采用 L_1 和 L_2 范数作为边界框回归损失函数与改善 IoU 值没有较强的关联,如图 3 所示。图 3 中 L_2 范数均相等,但 IoU 值却不同,可以直观地感受到第 3 个预测框最好,因此采用 L_1 、 L_2 范数作为边界框回归损失函数并不能较好地优化网络。

针对以上问题,可以很自然地想到将 IoU 作为边界框回归损失函数对优化网络的性能十分重要。但将 IoU 直接作为边界框损失函数会存在一些问题。如图 4 所示,黑色框是预测框,红色虚线框是真实框。当预测框与真实框不重合时,则

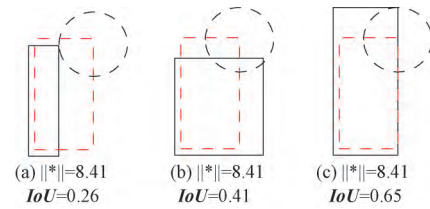
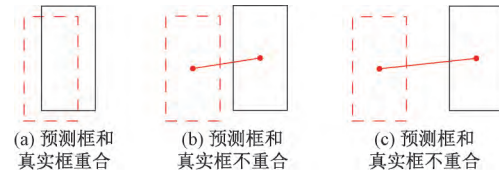
图 3 L_2 范数相同 IoU 不同Figure 3 L_2 norm is the same and IoU is different

图 4 真实框与预测框之间关系

Figure 4 The relation between ground truth box and predicted box

IoU 的值为 0, 导致无法优化网络,使得网络性能较差。

为了在边界框回归损失函数中利用 IoU , 如公式 (3) 所示, 将 F_{IoU} 作为边界框回归损失函数, 它表示预测框和真实框之间的偏差。

$$F_{IoU} = 1 - IoU, \quad (3)$$

式中: F_{IoU} 的取值范围 $(0, 1]$ 。当 $IoU = 0$ 时, F_{IoU} 一直为 1, 则无法反映预测框与真实框之间的距离关系, 如图 4 中的 (b) 和 (c) 两种情况。

为了体现不相交两个框之间距离关系, 受 YOLOv1^[7] 损失函数启发, 笔者利用真实框和预测框之间中心点距离 D 解决此问题, 如式 (4) 所示:

$$D = (x - x')^2 + (y - y')^2, \quad (4)$$

式中: (x, y) 是真实框的中心坐标; (x', y') 是预测框的中心坐标; 其值均是相对整张图片的坐标位置, 取值范围 $(0, 1)$ 。笔者提出最终的边界框回归损失函数 $DIoU$, 如式 (5) 所示:

$$DIoU = \lambda D + F_{IoU}, \quad (5)$$

$DIoU$ 是由式 (3) 和式 (4) 的结合, 式中 λ 是为了平衡两个损失函数值之间的差距, 取 $\lambda = 10$ 。

3 实验结果与分析

3.1 蔬菜数据集

由于目前没有公开的蔬菜数据集, 因此需要自己采集数据。蔬菜数据的来源主要是采用高清摄像机拍摄和爬虫技术获取。拍摄的数据占主要部分, 爬虫技术获取的数据为次要部分。蔬菜数据集目前有 20 类, 均为超市常见的蔬菜种类, 总共 7 632 张图片, 图片内的蔬菜类别为随机拍摄, 如图 5 所示。

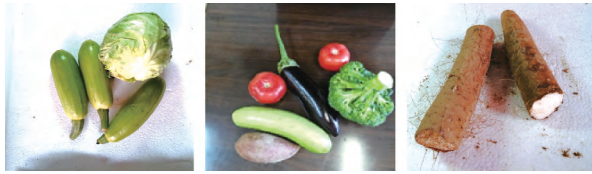


图5 蔬菜数据集

Figure 5 Vegetable dataset

数据集的类别分别是上海青、白菜、萝卜、黄瓜、西红柿、茄子、土豆、香菇、杏鲍菇、西葫芦、山药、洋葱、辣椒、西兰花、胡萝卜、莴笋、红薯、包菜、苦瓜和豆角等。笔者对蔬菜数据集内的目标进行统计分析得到总目标数是 22 493 个,表 2 是 VOC2007 数据集和蔬菜数据集分别在某个像素范围内的目标占总目标的比重。

表2 目标像素大小分布

Table 2 Object pixel size distribution

目标尺寸	占 VOC2007 总目标数的 比重/%	占蔬菜数据集 总目标数的 比重/%
(0 ,64×64]	30.8	0.6
(64×64 ,128×128]	25.2	13.9
(128×128 ,256×256]	27.7	22.2
(256×256 ,512×512]	16.3	49.8
大于 512×512	0.0	13.5

通过表 2 可知,蔬菜数据集的小目标相对较少,而大目标占据了较大的比重。因 YOLOv3 是针对 VOC 数据集和 COCO 数据集进行优化,对于蔬菜数据集无法获得较好的性能,因此笔者提出了改进型 YOLOv3 目标检测算法,使其适应于蔬菜数据集。

3.2 实验环境及参数

实验的蔬菜数据集共有 7 632 张图片,其中 1 232 张是测试集,6 400 张是训练集,在显卡 NVIDIA TITAN V 上运行。改进型 YOLOv3 算法输入图片像素尺寸设置为 512×512,对 5 组不同尺度的特征上采样构成特征金字塔,预测蔬菜目标的位置和种类。YOLOv3 的 9 组先验框是基于 COCO 数据集产生的,蔬菜数据集的目标尺度普遍偏大,因此原始的 9 组先验框已不适应蔬菜数据集。

为获得合适的先验框,笔者采用了 K-means 算法对蔬菜数据进行聚类分析,获取的 15 组先验框依次是(88 ,88)、(115 ,121)、(150 ,142)、(128 ,200)、(310 ,100)、(107 ,297)、(194 ,170)、(99 ,432)、(176 ,262)、(235 ,220)、(380 ,152)、(152 ,390)、(314 ,260)、(235 ,354)和(396 ,394)。再分

别将 15 组先验框均分到 5 组不同尺度的特征金字塔上。小尺寸的先验框在高分辨率特征图上用于检测小目标;大尺寸的先验框在低分辨率特征图上用于检测大目标。

训练时,改进型 YOLOv3 算法进行 50 000 次迭代,其中 momentum 与 weight decay 分别配置为 0.9 与 0.000 5,批 (batch size) 设置为 8,初始学习率为 0.000 1,迭代次数为 30 000 和 40 000 时,学习率分别降至 0.000 1、0.000 001。利用改进型 YOLOv3 算法对蔬菜数据进行目标检测的结果如图 6 所示。

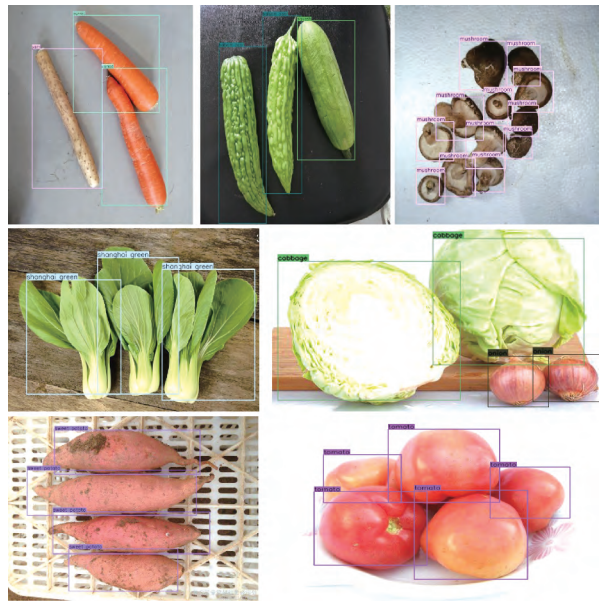


图6 检测结果

Figure 6 Detection results

3.3 DIoU 性能分析

在目标检测算法中,边界框回归损失函数作为衡量预测框与真实框之间的误差,其对目标定位效果起到重要作用。为验证 F_{IoU} 和 $DIoU$ 对 YOLOv3 算法的影响,笔者将 YOLOv3 的边界框回归损失函数 MSE 分别替换为式(3) 和式(5) 的 F_{IoU} 和 $DIoU$ 。3 种损失函数对 YOLOv3 算法影响如表 3 所示。

表3 边界框回归损失函数

Table 3 Bounding box regression loss function

损失函数	mAP/%
YOLOv3+MSE	89.8
YOLOv3+ F_{IoU}	90.9
YOLOv3+ $DIoU$	91.3

由表 3 可知,采用边界框回归损失函数 F_{IoU} 训练 YOLOv3 与 MSE 相比 mAP 提升 1.1%,从而可知将 IoU 考虑进边界框回归损失函数对网络性

能的提升具有重要作用。此外 ,YOLOv3 采用 $DIoU$ 边界框回归损失函数与 F_{IoU} 相比 mAP 提升 0.4% ,因此在考虑 IoU 的同时 ,引入预测框与真实框之间的中心点距离可以更好地优化 YOLOv3 算法。

图 7 是 YOLOv3 采用不同边界框回归损失函数训练时每轮的网络对测试集的检测结果。由此可知 ,与另外两种损失函数相比 , $DIoU$ 可以更快地提升网络的性能。不同边界框回归损失函数对蔬菜的检测效果如图 8 所示 ,其中红色框是目标真实框。由图 8 可知 ,以 IoU 为基础的 F_{IoU} 和 $DIoU$ 方法对目标的定位精度基本相同 ,而 MSE 方法的定位精度稍差。

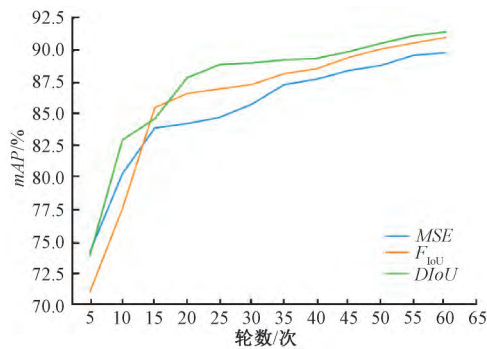


图 7 不同边界框回归损失函数对网络的影响
Figure 7 Effect of regression loss functions of different bounding box on network

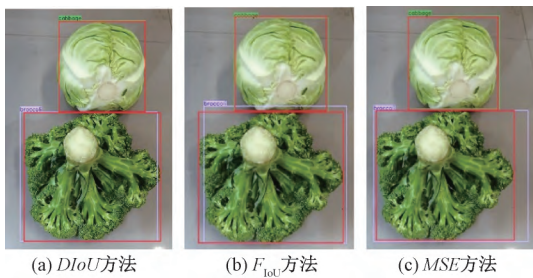


图 8 不同损失函数检测结果
Figure 8 Detection results of different loss functions

3.4 多尺度预测性能分析

原 YOLOv3 算法对蔬菜数据集的检测结果如图 9 所示。由图 9 可知 ,YOLOv3 在检测蔬菜时存在一些问题 ,如 图 9(a) 和 (c) 中山药和黄瓜出现漏检 ,图 9(b) 中将香菇错误检测为土豆 ,以及图 9(d) 中同一个目标被多个框标记。

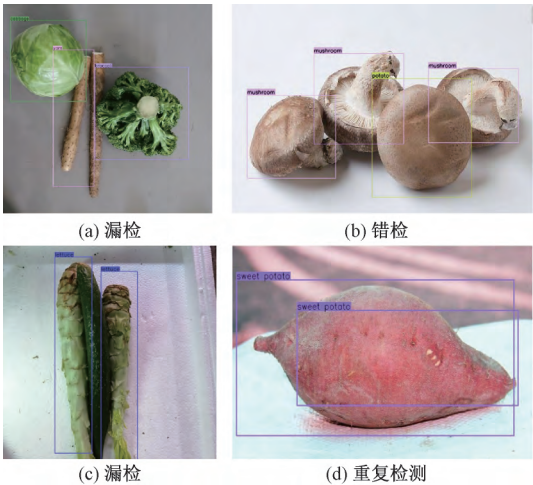


图 9 YOLOv3 检测结果
Figure 9 YOLOv3 detection results

改进型 YOLOv3 算法输入图片的像素尺寸设置为 512×512 ,从而获得 5 组特征图 ,其像素尺寸分别是 64×64 、 32×32 、 16×16 、 8×8 和 4×4 ,再将 5 组特征图上采样构成特征金字塔进行目标检测。改进型 YOLOv3 算法采用多尺度特征金字塔之后在蔬菜数据集上的检测性能如表 4 所示。由 4 组、5 组特征构成特征金字塔的 YOLOv3-1、YOLOv3-2 的 mAP 依次是 92.6%、93.2% ,与 YOLOv3 算法相比分别提升 1.3% 和 1.9%。而 YOLOv3-2 算法提升较大的原因是新的先验框、图片尺寸的增大以及更深的网络。

虽然 YOLOv3-2 检测速度与 YOLOv3 相比有一定的下降 ,但是 YOLOv3-2 算法利用 5 组不同尺度特征构成的特征金字塔进行目标检测时仍能满足实时性要求 ,并且检测精度更高。改

表 4 改进型 YOLOv3 对蔬菜数据集的检测结果
Table 4 Detection results of improved YOLOv3 on vegetable dataset

方法	输入图片尺寸	特征金字塔	帧速率/(f·s ⁻¹)	mAP/ %
YOLOv3	416×416	52×52 ,26×26 ,13×13	41	91.3
YOLOv3-1	512×512	64×64 ,32×32 ,16×16 ,8×8	37	92.6
YOLOv3-2	512×512	64×64 ,32×32 ,16×16 ,8×8 ,4×4	35	93.2

进型 YOLOv3 算法在蔬菜数据集上的检测结果如图 10 所示。从图 10 可以直观地看到 ,增强特征提取网络可以较好地解决漏检、错检和重复检测等问题。

笔者使用蔬菜测试集分别在 Faster RCNN、SSD、YOLOv3 和改进型 YOLOv3 进行测试。由表 5 可知 ,改进型 YOLOv3 比原 YOLOv3 的 mAP 提高了 3.4% ,虽然检测速度有所降低 ,但仍能满足实时性

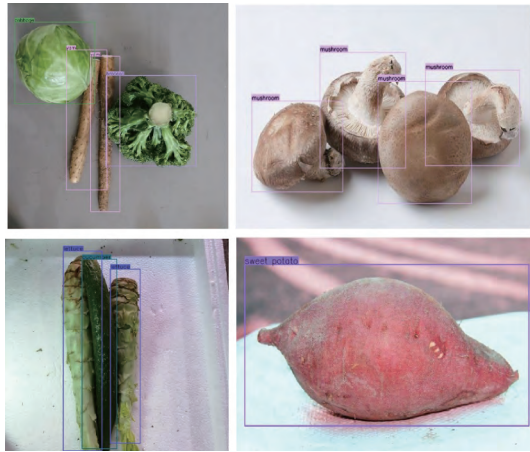


图 10 改进型 YOLOv3 检测结果

Figure 10 Improved YOLOv3 detection results

要求。同时,改进型 YOLOv3 算法的 mAP 比 SSD 和 Faster RCNN 分别高出 6.8% 和 5.4%,说明改进型 YOLOv3 算法对蔬菜数据更有效。

表 5 不同方法在蔬菜数据集上的性能对比

Table 5 Performance comparison of different methods on vegetable dataset

网络框架	主干网络	$mAP/\%$	帧速率/($f \cdot s^{-1}$)
Faster RCNN ^[4]	ResNet-50	87.8	9
SSD-300 ^[10]	VGG16	86.4	45
YOLOv3 ^[9]	Darknet-53	89.8	41
改进型 YOLOv3	Darknet-53	93.2	35

4 结论

笔者针对蔬菜数据集设计了改进型 YOLOv3 的特征提取网络,并且采用一种新的边界框回归损失函数 $DIoU$ 来提高精度。首先,改进型 YOLOv3 算法将 DenseBlock 和 DarkNet-53 相结合,获取具有更高语义信息的 5 组特征,进而构成特征金字塔,这在一定程度上解决了漏检、错检以及重复检测等问题。其次,通过 K-means 聚类分析得到适应于蔬菜数据集的先验框;同时, $DIoU$ 边界框回归损失函数增强了目标定位精度。改进型 YOLOv3 算法在蔬菜数据集上的 mAP 达到 93.2%,并获得 $35 f \cdot s^{-1}$ 的检测速度。在后续的工作中,将收集更多种类的蔬菜数据,并且利用改进型 YOLOv3 算法和硬件结合实现电子秤智能化,解决散装蔬菜称重问题,提高超市的运行效率。

参考文献:

[1] BURGESS C J C. A tutorial on support vector machines for pattern recognition[J]. Data mining and knowledge discovery, 1998, 2(2): 121-167.

[2] FELZENSZELB P F, GIRSHICK R B, MCALLESTER D, et al. Object detection with discriminatively trained part-based models[J]. IEEE transactions on pattern analysis and machine intelligence, 2010, 32(9): 1627-1645.

[3] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//2014 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Columbus, OH, USA: IEEE, 2014: 580-587.

[4] REN S Q, HE K M, 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.

[5] DAI J F, LI Y, HE K M, et al. R-FCN: Object detection via region-based fully convolutional networks[C]//2016 30th Annual Conference on Neural Information Processing Systems(NIPS). Barcelona, Spain: NIPS, 2016: 379-387.

[6] 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). Honolulu, HI: IEEE, 2017: 936-944.

[7] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Las Vegas, NV, USA: IEEE, 2016: 779-788.

[8] REDMON J, FARHADI A. YOLO9000: better, faster, stronger[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Honolulu, HI: IEEE, 2017: 6517-6525.

[9] REDMON J, FARHADI A. YOLOv3: an incremental improvement[EB/OL]. (2018-4-8) [2019-6-29]. <https://arxiv.org/pdf/1804.02767.pdf>.

[10] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]//2016 European Conference on Computer Vision(ECCV). Amsterdam, Netherlands: Springer, 2016: 21-37.

[11] 张素洁, 赵怀慈. 最优聚类个数和初始聚类中心点选取算法研究[J]. 计算机应用研究, 2017, 34(6): 1617-1620.

[12] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Las Vegas, NV, USA: IEEE, 2016: 770-778.

(下转第 31 页)

Sensorless Control of BLDCM Based on Improved Sliding Mode Observer

BAI Guochang , YAO Jiliang

(School of Mechanical and Power Engineering , Zhengzhou University , Zhengzhou 450001 , China)

Abstract: When the traditional sliding mode observer was used to observe the line back electromotive force (line back-EMF) of brushless dc motor (BLDCM) , a low-pass filter was required because of the excessive chattering of the system. However , the phase delay caused by the filter could not be accurately compensated , resulting in inaccurate commutation. This paper proposed to apply the sigmoid function to the sliding mode observer , and used its smooth and continuous characteristics to reduce chattering. At the same time , a variable sliding mode gain was deduced to further weaken the chattering of the system based on the Lyapunov theorem. Through these measures , the line back-EMF observations observed could directly determine the commutation signal. The simulation and experimental results showed that the improved sliding mode observer reduced the peak value of line back-EMF observation errors at 400 r/min and 3 000 r/min by 70% and 54. 8% respectively , obtained more accurate commutation signals , and improved the brushless dc motor positionless control system performance.

Key words: brushless dc motor; sensorless control; improved sliding mode observer; the system chattering; the sigmoid function; variable sliding mode gain

(上接第 12 页)

[13] LIU Y T , LIU Q S. Convolutional neural networks with large-margin softmax loss function for cognitive load recognition [C]//2017 36th Chinese Control Conference (CCC) . Dalian , China: IEEE , 2017: 4045 -4049.

[14] HUANG G , LIU Z , VAN DER MAATEN L , et al. Densely connected convolutional networks [C]//2017

IEEE Conference on Computer Vision and Pattern Recognition(CVPR) . Honolulu , HI: IEEE , 2017: 2261 -2269.

[15] SHEN Z Q , LIU Z , LI J G , et al. DSOD: Learning deeply supervised object detectors from scratch [C]//2017 16th IEEE International Conference on Computer Vision (ICCV) . Venice , Italy: IEEE , 2017: 1937 -1945.

Vegetable Recognition Algorithm Based on Improved YOLOv3

WEI Hongbin , ZHANG Duanjin , DU Guangming , XIAO Wenfu

(School of Information Engineering Zhengzhou University , Zhengzhou 450001 , China)

Abstract: The queuing and weighing problem was common in bulk vegetable area of supermarket. If weighing equipment could automatically recognize vegetable , it would effectively improve the operational efficiency of supermarket. Therefore , a vegetable recognition method based on improved YOLOv3 was proposed. Firstly , vegetable pictures were collected by using high-definition camera and web crawler technology. Secondly , 15 groups of anchors suitable for vegetable datasets were obtained by K-means clustering analysis. Thirdly , a new bounding box regression loss function DIoU was proposed to improve the precision of detection task. Finally , as there were many large objects in vegetable datasets , 5 groups of feature pyramids with different scales were obtained by enhancing feature extraction network to realize vegetable detection task. The mAP of the improved YOLOv3 algorithm on the test dataset was 93. 2% , and the recognition rate was 35 fps. This method improved the recognition of mAP while guaranteeing real-time object detection.

Key words: vegetable recognition; K-means; convolutional neural networks; feature pyramid; YOLOv3