

文章编号: 2095-2163(2019)01-0036-07

中图分类号: TP391.41

文献标志码: A

改进卷积神经网络算法在机械零件实时识别与定位中的应用

王乐, 周庆华, 王磊, 蒋华胜, 林思宇

(长沙理工大学 物理与电子科学学院, 长沙 410114)

摘要: 通用的目标识别与定位卷积神经网络算法难以兼顾精度和速度的要求。本文在 YOLO v2 卷积神经网络的基础上, 采用多尺度训练、网络预训练和 k-means 维度聚类等优化方法, 提出了机械零件实时识别与定位的改进卷积神经网络算法。本文以螺母和垫片 2 种物体为识别与定位的对象, 以工业传送带为场景, 同时考虑到了传送带上干扰物的存在, 对改进算法的准确率和速度进行了实验测试。实验结果证明本文的算法相对其它常用目标检测卷积神经网络算法在识别准确率和速度上达到了很好的平衡, 为零件实时分拣提供了基础。

关键词: 卷积神经网络; 机械零件; 目标检测

Improved convolutional neural network algorithm for real-time recognition and location of mechanical parts

WANG Le, ZHOU Qinghua, WANG Lei, JIANG Huasheng, LIN Siyu

(School of Physical & Electronic Science, Changsha University of Science & Technology, Changsha 410114, China)

[Abstract] Universal target recognition and location of convolutional neural network algorithms are difficult to balance the accuracy and speed requirements. Based on YOLO v2 convolutional neural network, this paper adopts multi-scale training, network pre-training and k-means dimension clustering optimization methods to propose an improved convolutional neural network algorithm for real-time recognition and location of mechanical parts. In this paper, two kinds of objects, nuts and pads, are used to identify and locate the object. The industrial conveyor belt is taken as the scene. At the same time, the existence of interferences on the conveyor belt is considered. The accuracy and speed of the improved algorithm are tested experimentally. Compared with other common target detection convolutional neural network algorithms, the algorithm achieves a good balance in the recognition accuracy and speed, and provides a basis for real-time sorting of parts.

[Key words] convolutional neural network; mechanical parts; object detection

0 引言

工业生产中机械零件种类繁多、产量大,人工分拣不仅影响工作效率,而且带来不可靠的因素,直接影响产品质量与成本。对于一些危险场合,人工分拣更是无法完成的。因此工件自动化分拣是自动化生产系统中的重要环节,可以有效地解决工业生产中传送带上零件集中混型输送和识别^[1],而提高分拣精度、减少分拣耗时是提升生产效率的重要途径。传统自动化生产线中的分拣技术主要应用各种传感器的组合,对工件的材质、颜色、形状等进行检测^[2],这些方法算法简单、对处理器要求不高,但对工件的摆放位置具有严格的要求,灵活性差,检测速度慢,有时对工件进行接触,造成一定损伤。

随着现代信息技术的发展,机器视觉技术因其

高精度、速度快、无接触、无损检的特点逐渐取代传统自动化分类和定位技术。机器视觉技术一般通过摄像头采集图像信息,然后进行二值化、边缘提取、SIFT、SURF 等操作提取特征信息,最后通过支持向量机、k 近邻等分类器对目标物体进行分类。但此类方法很大程度上依赖于特定的检测条件,对于不同的目标或者同一目标的不同形态,需要设计不同的方法去提取特征,算法的最终识别效果也受制于设计者的经验,普适性差,很难满足复杂场景的要求。

卷积神经网络(convolutional neural network, CNN)将人工神经网络和卷积运算相结合,通过逐层提取,获得了丰富的特征信息,并对一定程度的扭曲和形变有良好的鲁棒性^[3]。同时,卷积神经网络采用局部感受野、稀疏连接和权值共享,极大减少了神经网络的训练参数,降低了网络运算的复杂度。因

基金项目: 湖南省教育厅资助科研项目(16K003)。

作者简介: 王乐(1993-),女,硕士研究生,主要研究方向:卷积神经网络;周庆华(1977-),男,博士,副教授,主要研究方向:机器视觉、人工神经网络、电磁场与电磁波。

通讯作者: 周庆华 Email: zhouqinghua@csust.edu.cn

收稿日期: 2018-11-08

此, 本文将卷积神经网络算法用于机械零件的定位与分类中。2012 年 Hinton 等人^[4]提出了 AlexNet 卷积神经网络, 在著名的 ImageNet 图像分类问题上取得了最好成果, 将 top-5 的错误率下降了 15%。2014 年 Christian Szegedy^[5]等人提出的 Inception 结构, 进一步将 top-5 的错误率降到了 6.67%, 获得了大型机器视觉挑战赛 (ILSVR) 的冠军。2014 年 Ross Girshick^[6]等人提出了 R-CNN 算法, 其主要分为产生候选区域、提取特征信息和进行图像分类 3 个阶段。但由于这 3 个阶段彼此分离, 导致检测过程复杂, 花费时间较长, 且难以优化。2015 年 Ross Girshick 和 Shaoqing Ren 又相继提出了 Fast R-CNN^[7]和其改进版 Faster R-CNN^[8]。Faster R-CNN 在一定程度上降低了检测的复杂度, 但检测速度仍然较慢, 难以满足实际工业中实时性的要求。

2016 年 Joseph Redmon^[9]等人提出了 YOLO 目标检测算法, 这是一种全新的端到端 (End-to-End) 的检测算法, 使用单一的卷积神经网络在检测图像上同时完成了边界框和类别概率预测。同年在 YOLO 的基础上, Joseph Redmon^[10]又提出了 YOLO v2。虽然 YOLO v2 相比 R-CNN 系列检测算法在检测速度上有了很大的提升, 但检测准确率并不是很高, 尤其是对于复杂场景和小目标物体的检测。

本文以螺母和垫片 2 种目标物体为检测对象, 以工业传送带为研究场景。同时, 考虑到传送带上干扰物的存在, 在 YOLO v2 检测算法的基础上, 采用多尺度训练、网络预训练和 k-means 维度聚类等优化方法, 提出了用于机械零件实时识别与定位的改进卷积神经网络算法。改进算法能够实现运动传送带上目标物体的识别和定位, 检测准确率达到 85% 以上, 检测速度达到了实时性的要求, 为零件实时分拣提供了基础。

1 基本理论

1.1 卷积神经网络的特点

卷积神经网络是一种前向传播和反向传播相结合的人工神经网络, 是一种典型的深度学习算法, 在传统的神经网络的基础上增加了卷积和池化操作。卷积神经网络采用局部感知、参数共享、下采样等技术减少了网络参数的数量。在保证提取足够特征信息的同时, 降低了计算的复杂度, 提高了网络识别的效率^[11]。

1.2 CNN 模型

LeNet-5^[12]是卷积神经网络的经典网络, 最早

被用于手写数字的识别, 取得了很好的效果。其总共有 7 层 (不包含输入层), 由 2 个卷积层、2 个池化层、2 个完全连接层和 1 个输出层组成。局部感知和参数共享理论被应用在卷积层, 下采样操作作用在池化层。这样就构成了完整的卷积神经网络系统。

图 1 为 LeNet-5 网络结构图, 输入图片大小为 32×32 , 采用 5×5 的卷积核进行卷积操作, 步长为 1, 即每次滑动窗口的大小为 1 个像素, 得到 28×28 的特征图。下采样层以 2×2 的卷积核, 每次滑动 2 个像素大小进行平均池化操作, 得到 14×14 特征图。完全连接层将所有结点汇总起来, 传给输出层。输出层具有 10 个结点, 代表 0 到 9 数字的输出概率。

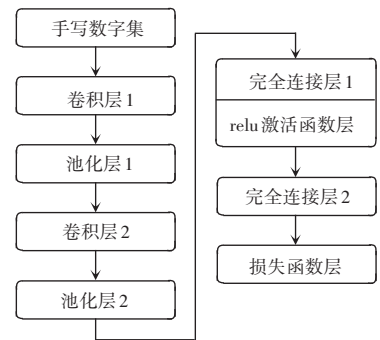


图 1 LeNet-5 模型的结构

Fig. 1 The structure of the LeNet-5 model

1.3 YOLO 算法

YOLO 算法属于 CNN, 拥有 24 个卷积层和 2 个完全连接层, 其中大量运用了卷积的级联结构, 卷积核主要包括 3×3 和 1×1 两种大小的卷积核。YOLO 将整张图片用于训练, 不需要任何的裁剪和预处理操作。其将整个目标的检测作为回归问题来处理, 训练出的网络模型能够接受原始的像素信息, 在输出端直接输出目标物体的边界框和类别信息, 是一个端到端的检测模型。在检测过程中, 将整个图片分成 $S \times S$ 的网格, 如果有目标物体的中心落在某个单元格里, 这个单元格负责检测该物体。每个单元格需要预测 B 个边界框, 对于每个边界框需要预测其置信度。计算公式如下:

$$p_r(\text{object}) \times IOU_{pred}^{truth} \quad (1)$$

其中, $p_r(\text{object})$ 表示边界框中包含目标物体的概率, IOU_{pred}^{truth} 则为预测边界框和实际边界框的交集除以预测边界框和实际边界框的并集。每个检测边界框包含 5 个参数: x, y, w, h 和置信度, (x, y) 代表检测边界框的中心相对其母网格的位置。 (w, h) 为检测边界框相对于整张图片的占比, 同时每个网

格还预测 C 个类别概率,即 $p_i(class_i/object)$, 该概率表示第 i 类物体中心落入该网格的概率, C 为类别数量。YOLO 采用误差平方和作为损失函数,其具体形式如下:

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\ & \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] + \\ & \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{noobj} (C_i - \hat{C}_i)^2 + \\ & \sum_{i=0}^{s^2} I_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \quad (2)$$

可以看到,损失函数主要包括 5 部分。其中 I_i 表示是否有物体中心落入网格 i 中,若有 I_i 值为 1,反之为 0。 I_{ij} 表示判断网格 i 中第 j 个检测边界是否负责该物体,若是则为 1,反之为 0。

YOLO 虽然速度快,但有以下 3 个缺点:

(1) 与目前流行的 R-CNN 系列目标识别算法相比,精度并不是很高。

(2) 与基于候选区域的算法相比,虽然对于背景的误判较小,但是对于目标的定位并不是很准确。

(3) 由于 YOLO 算法规定一个网格中只能预测一类物体,因此 YOLO 对于粘连物体和小目标物体的识别精度不高。

针对以上缺点,YOLO v2 在 YOLO 的基础上进行改进。去掉了 YOLO 网络的完全连接层和最后一个最大池化层,使用每个 anchor box 来预测物体的种类,并且将网络的输入调整到了 $416 * 416$,使得网络在经过多次卷积和下采样操作后,得到的特征图为 $13 * 13$,有利于中心点的计算。YOLO v2 在每个卷积层的后面都加入了批量归一化操作^[13],这样在去掉 dropout 层后,网络也不会过拟合,同时也提高了网络预测的精度。针对 YOLO 对于小物体预测精度低的问题,YOLO v2 借鉴残差网络,引入了传递层,即将上一层 $26 * 26$ 的特征图直接与 $13 * 13$ 的特征图相连,使得网络能够获取更加丰富的特征信息。

在 YOLO v2 之前,大部分的网络结构都是基于 VGG-16^[14],但 VGG-16 网络结构复杂、参数较多、计算量大。Darknet-19 作为 YOLO v2 的基本网络结构,参考了 YOLO 和 SSD^[15] 的网络结构,兼顾了精度和复杂度,使得网络在具有较高精度的同时,具有较少的参数。Darknet-19 具有 19 个卷积层和 5 个最大池化层,在 ImageNet 数据集上可以达到 72.9% 的 top-1 精度和 91.2% 的 top-5 精度。网络

中以 $3 * 3$ 的卷积核为主,大量使用了卷积的级联,并借鉴了 Network in network^[16] 的思想,在 $3 * 3$ 的卷积之间均采用了 $1 * 1$ 的卷积核,在减少网络训练参数,降级计算成本的同时增加了网络的深度。

2 用于机械零件识别和定位的改进算法

考虑到在实际分拣中可能存在各种干扰物,采用通用的目标识别与定位卷积神经网络算法难以兼顾精度和速度的要求。因此,本文以 Darknet-19 作为网络训练框架,并从网络预训练、多尺度训练、k-means 维度聚类 3 个方面进行优化,以达到复杂场景中,传送带上目标物体实时识别与定位的目的。目标检测算法的流程如图 2 所示。由于标签数据集稀少,首先在 ImageNet 数据集上进行网络预训练,然后在标签数据集上进行再训练。通过前向传播获取网络输出,依靠反向传播更新参数。在目标检测的过程中,加载训练好的网络模型获取特征图,使用 k-means 维度聚类优化候选框,得到 2 组候选框的尺寸,然后在特征图中产生边界框。对于一个目标物体可能产生多个边界框,需要通过非极大值抑制来留下置信分数最高的边界框,实现对物体的精确定位,对于每个边界框预测出的多个物体的类别概率,也通过非极大值抑制留下类别分数最高的预测,实现对物体的分类。

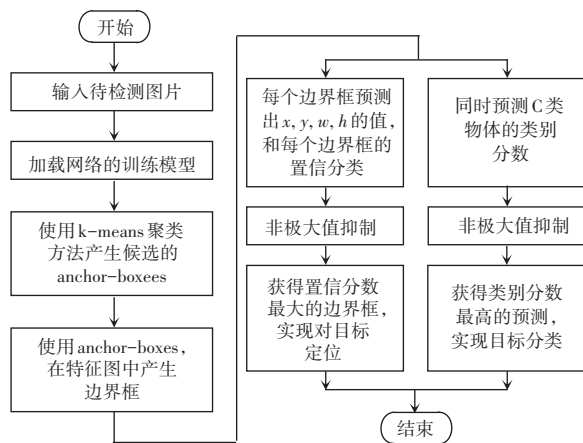


图 2 目标检测算法的流程

Fig. 2 The flow chart of target detection of this paper

2.1 网络预训练

由于目标物体的标签数据较少,且往往分辨率较低,直接采用制作的标签数据集进行网络的训练,往往精度不高,定位较差。因此,在实际的训练中采用了预训练的方法,提高了物体识别的精度,同时降低网络训练的时间。本文网络的训练主要分为以下 2 步:

(1) ImageNet 数据集有 1 400 多万幅图片, 有着 1 000 个目标类别, 图像清晰, 分辨率较高, 且多数的图片具有明确的类别标注信息。因此, 可以在 ImageNet 1 000 类的数据集上进行预训练。由于手工收集的标签数据集只有 2 个类别, 因此, 将预训练后的网络最后一层由 1 000 个类别输出调整为 3 个类别输出 (包括 2 个类别和背景), 使网络获得对目标物体的普遍认识, 达到对网络权重参数的初始化操作。

(2) 经过预训练的模型已经收敛到局部最优解, 然后采用收集到的 1 000 个手工标签的螺母和垫片数据集对网络进行再训练。在此以 Darknet-19 作为网络训练框架, 为了适应本文的检测任务去掉了最后一个卷积层, 增加了 3 个 3×3 的卷积层, 卷积核的个数为 1 024, 同时, 添加了一个 1×1 的卷积层, 使卷积层的输出数量与检测任务的输出一致。在每轮模型训练开始之前将训练数据模型集随机打乱, 确保模型不同轮数相同批次状态下接触到不同数据, 这样的操作可以加快网络的收敛速度。在模型训练时, 将初始的学习率设为 0.000 1, 在训练 10 轮后将学习率提高到 0.001, 适当的加快模型的训练次数, 当训练次数达到 20 000 时将学习率降低为 0.000 1, 在 5 000 时将学习率降为 0.000 01, 避免因为较大的学习率使模型跳过最优解。在迭代训练的过程中将批处理样本设置为 32, 通过前向传播预测输出, 通过反向传播和随机梯度下降 (SGD) 更新参数。

2.2 多尺度训练

为了使网络对不同输入尺寸的图片具有一定的鲁棒性, 在训练的过程中采用 5 种单一尺度和多尺度训练的方法。对于单一尺度的训练, 进行了 5 组实验, 5 组实验的输入图片尺寸分别为: 288×288 , 352×352 , 416×416 , 480×480 , 544×544 。对于多尺度的训练方法, 在训练的过程中每隔 10 轮, 则随机的改变输入图片的尺寸。因为模型的下采样因子为 32, 则多尺度输入图片必须是 32 的倍数, 其计算公式为:

$$s = 32(7 + a) \quad (3)$$

其中, s 为输入图片的尺寸, a 为在训练过程中随机产生的 0 到 12 之间的整数。

2.3 用 k-means 维度聚类法优化候选框

为了提高网络的召回率, YOLO v2 的作者 在文中使用了 5 种不同尺寸的候选框, 但是对于特定的检测任务效果并不理想^[18]。本文针对螺母和垫片的检测任务, 通过 k-means 对数据集中手工标记的

目标框做聚类分析, 找到目标框的统计规律。首先, 通过肘部法则来估计聚类的个数。随着 k 值的增大, 平均畸变程度会降低, 每个类中包含的样本数会减少, 样本离其聚类中心点会更近, 但随着 k 值继续增大, 平均畸变的改善效果会逐渐降低。因此, 取平均畸变改善效果最好的位置所对应的 k 值为聚类的个数。然后, 以聚类个数 k 为候选框的个数, 对标签数据集中目标物体归一化后的宽和高进行聚类分析。图 3 显示了 k 值对平均畸变的影响。可以看出, 当 $k = 2$ 时, 平均畸变的改善效果下降幅度最大, 平均畸变的改善效果最好, 此时标签数据集的聚类效果达到最好, 其空间聚类效果如图 4 所示, 其中绿色的正方形代表各聚类中心点, 候选框的尺寸为 2 个颜色区域的聚类中心点所对应的目标边界框的宽和高。由此得到 2 组候选框的尺寸为 $[4.54, 4.43]$, $[5.27, 5.41]$ 。

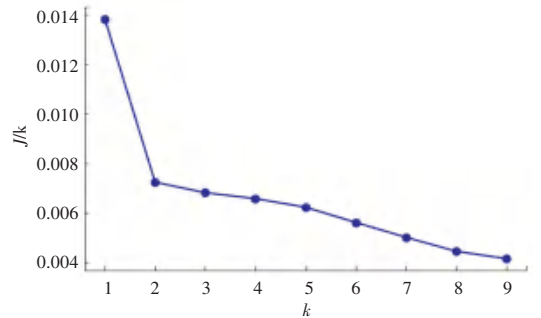


图 3 k 值对平均畸变的影响

Fig. 3 The effect of k value on the average distortion

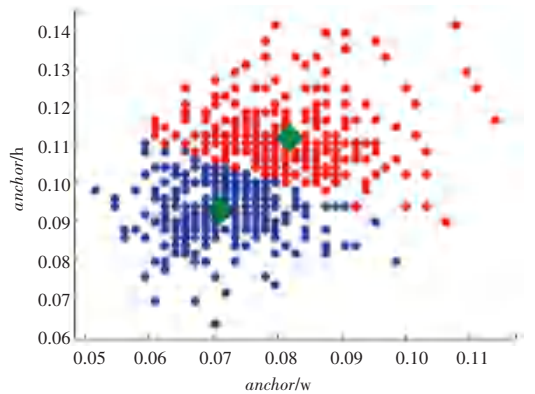


图 4 $k=2$ 时的聚类效果

Fig. 4 The clustering result when $k=2$

3 实验与结果分析

3.1 实验配置与训练结果

本文以螺母和垫片 2 种物体为识别与定位的对象, 以工业传送带为场景, 同时考虑到了传送带上干扰物的存在。通过传送带上的摄像头, 实时截取

1 000张分辨率为 $640 * 480$ 的图片,其中 100 张图片作为验证数据集,900 张图片作为训练数据集。通过 Labellmg 软件,手动标记出图像中螺母和垫片的位置,并通过旋转、缩放、饱和度和曝光度等方式对数据进行扩充。

本文在配置有 2.60 Ghz 双核 Intel Core CPU 和 Ubuntu16.04 系统下进行网络的训练和测试,显卡型号是 NVIDIA - GTX 1050 Ti 独立显卡,开发包 CUDA 8.0。开发环境为 vs2013 和 opencv3.1.0。开发框架为 Darknet,训练过程中的部分网络参数的设置见表 1。

表 1 网络训练参数

Tab. 1 Network training parameters

参数	数值
learning_rate	0.000 1
policy	steps
steps	100, 20 000, 35 000
scales	10,0.1,0.1
batch	32
subdivisions	8
max_batches	50 000
momentum	0.9
decay	0.000 5

图 5 显示了一个检测结果的例子。其中,nut 代表目标物体螺母,pad 代表目标物体垫片,可以看到运用训练出来的模型能够对复杂场景中的目标物体进行正确的分类和定位。验证集中平均准确率达到 0.858,检测速度可以达到每秒 23 帧。

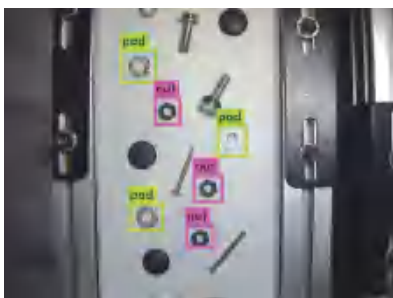


图 5 多目标、复杂场景检测结果

Fig. 5 The results of multi-target, complex scene detection

3.2 改进算法的效果

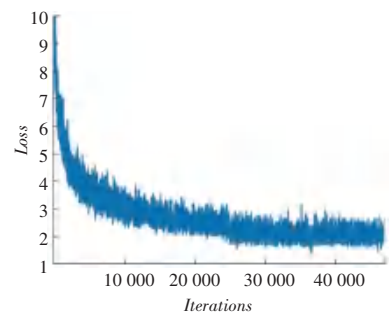
为了测试本文算法的有效性,首先分别对网络预训练、多尺度训练、k-means 维度聚类优化候选框等优化方法的效果进行测试,然后将本文算法与 Faster R-CNN 及 YOLO v2 算法的准确率和检测速度进行了对比。表 2 给出了网络预训练与直接采用

标签数据集训练的结果对比,参数 AP (Average Precision) 表示平均准确率。可以看到本文的检测精测与直接使用标签数据集的检测精度相比有了一定的提高。图 6 是采用网络预训练和直接训练时,迭代次数与损失函数的关系。(a) 为直接训练时,损失函数与迭代次数的关系,(b) 为采用网络预训练时,迭代次数与损失函数的关系。可以看到当迭代次数达到 40 000 次时,采用直接训练的方法,损失函数收敛到 2.0,而采用网络预训练的方法,损失函数收敛到 1.5 左右。因此,采用预训练的算法能够使模型更快更好地收敛。训练时使用的输入图片大小为 $416 * 416$,批处理尺寸为 16。

表 2 网络预训练与直接训练结果对比

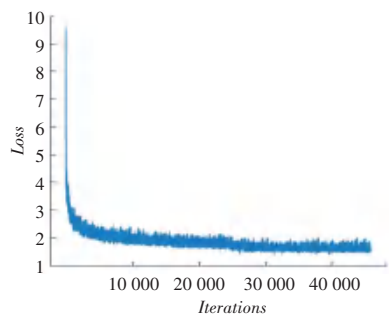
Tab. 2 The comparison between network pre-training and direct training results

训练方法	验证集 AP	
	螺母	垫片
直接训练的方法	0.796	0.790
网络预训练的方法	0.825	0.829



(a) 直接训练

(a) Direct training



(b) 网络预训练

(b) Network training

图 6 采用网络预训练和直接训练时,迭代次数与损失函数的关系

Fig. 6 The relationship between the iterations and the loss function during pre-training and direct training

在对多尺度训练效果测试实验中,采用了 $288 * 288, 352 * 352, 416 * 416, 480 * 480, 544 * 544$ 这 5 种单一尺寸和多尺度的训练方式。在多尺度的

训练过程中,每隔 10 轮就改变输入尺寸的大小。使用验证集对训练好的模型进行评估,表 3 给出了验证集中每类物体的检测时间和精度,参数 AP (Average Precision) 表示平均准确率。可以看到,随着输入图片尺寸的增加,检测时间增加,同时精度提高。对于使用多尺度训练的网络,采用 416 * 416 的输入图片进行测试,可以看到相比其它输入图片的尺寸,多尺度训练的图片在检测时间和检测精度上达到了较好的平衡。因此,采用多尺度训练的网络对不同输入尺寸的图片具有较强的鲁棒性。

表 3 多尺度与单一尺度训练结果对比

Tab. 3 The comparison of multi-scale and single-scale training results

输入尺寸	检测时间/s	验证集 AP	
		螺母	垫片
288×88	0.022	0.795	0.798
352×352	0.026	0.812	0.816
416×416	0.037	0.825	0.829
480×480	0.042	0.836	0.836
544×544	0.052	0.838	0.839
多尺度	0.035	0.832	0.837

在对 k-means 维度聚类优化候选框效果的测试实验中,使用自采集的标签数据集,对目标边界框的宽和高进行聚类信息,得到 2 组候选框的尺寸为 [4.54, 4.43], [5.27, 5.41]。表 4 给出了 YOLO v2 和维度聚类方法的结果对比,本文的维度聚类方法在候选框数量较少的情况下保证了较高的平均重叠率 (IOU) 和检测精度,同时也节省了计算资源。训练时使用的输入图片大小为 416 * 416,批处理尺寸为 16。

表 4 k-means 维度聚类与 YOLO v2 结果对比

Tab. 4 The comparison of k-means dimension clustering and YOLO v2 results

候选框产生的方法	候选框的数量	平均重叠 (IOU)		验证集 AP	
		螺母	垫片	螺母	垫片
YOLO v2	5	0.657	0.628	0.825	0.829
维度聚类	2	0.661	0.643	0.827	0.832

最后,将本文提出的改进卷积神经网络算法与 Faster R-CNN 和 YOLO v2 算法进行对比,结果见表 5。Faster R-CNN 中采用了 ZP 网络框架训练的模型检测准确率高于本文的检测算法,但在本文硬件配置下只达到了每秒 7 帧的检测速度,很难满足复杂场景中实时性要求。同时,将本文的算法与 YOLO v2 算法进行对比,本文的算法在检测准确率

和速度上都有了提升,基本能够满足复杂场景中实际分拣的需求。

表 5 本文算法与通用目标检测算法的对比

Tab. 5 The comparison between the proposed algorithm and universal target detection algorithms

检测算法	验证集 AP		检测速度 (FPS)
	螺母	垫片	
Faster R-CNN ZF	0.888	0.974	7
YOLO v2	0.825	0.829	20
本文算法	0.858	0.858	23

4 结束语

本文以 YOLO v2 卷积神经网络算法为基础,通过多尺度训练、网络预训练、k-means 维度聚类以及批处理尺寸设置对算法进行改进。通过多尺度训练,增强了网络对输入图片尺寸的鲁棒性;采用网络预训练在标签数据较少的、图像分辨率不高的情况下提高了物体识别的准确率;k-means 维度聚类算法对自己的标签数据集中目标边界框的宽和高进行聚类分析,得到候选边界框信息,提高了物体定位的准确度;通过设置适当的批处理大小,在保证实时性分拣的同时,提高物体识别的准确度,减少网络训练的时间和 GPU 计算的负担。本文算法在 NVIDIA - GTX 1050 Ti 的配置下,达到 0.858 的检测准确率和每秒 23 帧的检测速度。在识别准确率和速度上达到了很好的平衡,为零件实时分拣提供了基础。

参考文献

- [1] 张丛. 基于机器视觉的机械零件的识别研究[D]. 天津:天津职业技术师范大学,2015.
- [2] 管经纬. 基于机器视觉的工件自动分拣系统研究[D]. 上海:东华大学,2015.
- [3] 黄睿. 基于卷积神经网络的目标检测模型的研究[D]. 上海:上海交通大学,2014.
- [4] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks [C]// Advances in neural information processing systems. Lake Tahoe, Nevada, USA: dblp,2012: 1097-1105.
- [5] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA: IEEE,2015: 1-9.
- [6] GIRSHICK R, DONAHUE J, DARELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]//Proceedings of the IEEE conference on computer vision and pattern recognition.Columbus, Ohio:IEEE, 2014: 580-587.
- [7] GIRSHICK R. Fast r-cnn [C]//Proceedings of the IEEE International Conference on Computer Vision. Santiago, Chile: IEEE, 2015:1440-1448.