基于 YOLO 深度卷积神经网络的复杂背景下机器人采摘苹果定位

赵德安[1]，吴任迪[1]，刘晓洋[1]，赵宇艳[2]

(1. 江苏大学电气信息工程学院，镇江 212013；2. 常州信息职业技术学院电子与电气工程学院，常州 213164)

摘 要：为提高苹果采摘机器人的工作效率和环境适应性，使其能全天候的在不同光线环境下对遮挡、粘连和套袋等多种情况下的果实进行识别定位，该文提出了基于 YOLOv3(you only look once)深度卷积神经网络的苹果定位方法。该方法通过单个卷积神经网络(one-stage)遍历整个图像，回归目标的类别和位置，实现了直接端到端的目标检测，在保证效率与准确率兼顾的情况下实现了复杂环境下苹果的检测。经过训练的模型在验证集下的 mAP(mean average precision)为 87.71%，准确率为 97%，召回率为 90%，IOU(intersection over union)为 83.61%。通过比较 YOLOv3 与 Faster RCNN 算法在不同数目、不同拍摄时间、不同生长阶段、不同光线下对苹果的实际检测效果，并以 *F*1 为评估值对比分析了 4 种算法的差异，试验结果表明 YOLOv3 在密集苹果的 *F*1 高于 YOLOv2 算法 4.45 个百分点，在其他环境下高于 Faster RCNN 将近 5 个百分点，高于 HOG+SVM(histogram of oriented gradient + support vector machine)将近 10 个百分点。并且在不同硬件环境验证了该算法的可行性，一幅图像在 GPU 下的检测时间为 16.69 ms，在 CPU 下的检测时间为 105.21 ms，实际检测视频的帧率达到了 60 帧/s 和 15 帧/s。该研究可为机器人快速长时间高效率在复杂环境下识别苹果提供理论基础。

关键词：收获机；机器视觉；图像识别；深度学习；采摘机器人；苹果识别；YOLO

doi：10.11975/j.issn.1002-6819.2019.03.021

中图分类号：TP391.4 **文献标志码**：A **文章编号**：1002-6819(2019)-03-0164-10

赵德安，吴任迪，刘晓洋，赵宇艳. 基于 YOLO 深度卷积神经网络的复杂背景下机器人采摘苹果定位[J]. 农业工程学报，2019，35(3)：164－173. doi：10.11975/j.issn.1002-6819.2019.03.021 http://www.tcsae.org

Zhao Dean, Wu Rendi, Liu Xiaoyang, Zhao Yuyan. Apple positioning based on YOLO deep convolutional neural network for picking robot in complex background[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2019, 35(3): 164－173. (in Chinese with English abstract) doi：10.11975/j.issn.1002-6819.2019.03.021 http://www.tcsae.org

0 引 言

果树果实采摘机器人的研究随着科技的发展日臻完善，视觉识别系统作为采摘机器人的一部分，其目标检测的速度、准确率以及对周边环境的适应能力对机器人的工作效率和工作时长有较大影响。稳定的目标识别可以让机器人长时间地工作，压缩劳动成本，提高生产效率。一个能够在不同环境下进行采摘作业的机器人对提升农业生产力有较大的应用价值和现实意义。

目前国内外在水果检测方面的研究已经取得了一定的进展。Kapach 等[1]简述了关于水果采摘机器人视觉部分，全面阐述了当前领域各种方法的优点和局限性，其中大部分工作的研究都涉及到目标的颜色，但是关于套袋苹果或者是未成熟苹果这类颜色不明显的目标检测与识别的研究工作较少且起步较晚。吕继东等[2] 利用基于 *R-G* 颜色特征的 OTSU 动态阈值分割的方法进行图像分割，找出苹果的位置。但是对于光照非常敏感，在逆光环境下会丢失目标。赵德安等[3] 为了解决夜间苹果的识别问题，通过 2 个白炽灯作为照明光源从不同角度照明削弱图像中的阴影，采用二次分割的方法提取出苹果表面的高亮反光区并对此进行修补，得到完整的分割结果。但由于遮挡、光照不均等原因，光照无法保证，因此无法保证苹果的全部识别。Ji 等[4]利用 SVM(support vector machine)分类器进行苹果的分类识别，对套袋苹果的识别率达到 89%。但是识别一幅图像需要 352 ms，识别效率不高，无法达到实时性的要求。Stajnko 等[5]通过热相仪测量苹果与背景之间的温度梯度来识别不同成熟度的苹果并且计算出苹果的直径和数量。Wachs 等[6]利用彩色图像和热图像相结合的方式来识别苹果，但是利用热像仪的方法只能在阳光直射的下午使用，无法满足采摘机器人全天候工作的要求。Rakun 等[7]通过纹理和颜色相结合的方法来识别苹果，但是苹果上的果袋会消去纹理特征，难以识别。同样的还有其他利用颜色、纹理、形状等多种特征组合来实现苹果的目标检测，但是这类方法都是针对某些特定情景下进行的研究，鲁棒性差，无法满足采摘机器人各种复杂情景下的工作需求[8-17]。

相比于常规方法，近几年深度卷积神经网络在目标检测领域显现出巨大的优越性，深度神经网络由于其对目标高维特征的高度提取，使得对复杂情况下苹果的识

收稿日期：2018-10-19 修订日期：2019-01-21

基金项目：国家自然科学基金(31571571)；江苏高校优势学科建设项目(PAPD)

作者简介：赵德安，博士，教授，博士生导师，主要从事智能控制、智能机器人研究。Email：dazhao@ujs.edu.cn

别变成可能。其主要分为 2 种方法，一种是基于区域建议的方法，标志性的算法有 RCNN[18]、Fast RCNN[19]和 Faster RCNN[20]，核心思想是先获得建议区域然后在当前区域内进行分类，称为 two-stage 目标检测。熊俊涛等[21]利用 Faster RCNN 方法进行不同光照和不同尺寸的绿色柑橘识别，其模型的精度很高，泛化能力强，但是由于区域建议步骤消耗大量的计算资源，检测时间较长，不满足实时性的要求。另一种方法是无区域建议的方法，标志性的算法有 SSD[22]、YOLO[23]，核心思想是用单一的卷积网络直接基于整幅图像来预测目标的位置及其属性，也称为 one-stage 目标检测。薛月菊等[24]采用 YOLOv2 网络进行未成熟芒果的识别，在保持着精度和泛化能力的同时，提高了检测速率。但是其对于密集苹果的检测召回率明显下降，容易将一簇苹果只识别为一个苹果，而且其试验平台为 12 GB 的 GTX Titan X 显卡，其功率和可靠性都无法安置在采摘机器人之中，没有验证在只有 CPU 的工控机中的运行效率。

本文提出用 YOLOv3 神经网络来实现复杂环境下苹果的识别，在不影响精度的情况下压缩网络规模来加快检测速度，在不同拍摄时间、不同生长阶段、不同光线等干扰场景下分别进行网络的识别试验，以期检测网络效果，并且在 CPU 和 GPU 配置下进行试验以验证能否达到实时的要求。

1　目标识别算法

1.1　识别对象的分析

苹果图像的采集地点在江苏省徐州市丰县大沙河镇苹果示范基地和山东省烟台市蓬莱县苹果示范基地，采集时间分为白天和夜晚（图 1），白天是在自然光下采集，夜晚是在 LED 下采集，苹果的品种是红富士，有裸露和套袋 2 种（给苹果套上果袋之后能够提高苹果的品质）。

图 1　复杂的实际苹果图像

Fig.1　Complex actual apple images

图 1 中分别列举苹果分别在白天、夜晚、重叠、遮挡、带套、逆光、密集等各种环境下实际画面，苹果在裸露条件下能够保持基本颜色特征不变，带上果袋之后因为薄膜对光的反射使得颜色通道被割裂和腐蚀，并且由于果袋的不完全覆盖，使得苹果形状特征变得不规则。同时苹果如果在重叠和遮挡环境下会使得形状特征不完整，在夜晚会让苹果颜色强度发生改变。

如果使用 *R-G* 色差分割的方法难以将完整的苹果分离出来，用椭圆拟合的方法难以对不规则的苹果进行拟合。如果使用 SVM 支持向量机对多个特征进行精确整合创建词袋（Word of Bag）并进行分类来识别苹果，可能会因为正负样本的局限性无法适应所有环境下苹果的识别。

因此本文考虑用深度卷积神经网络来提取苹果特征，采用端到端的整体训练让神经网络自适应地学习不同环境下的苹果所需要的特征，实现复杂环境下苹果的定位。

1.2　YOLO 算法

YOLO 是目前为止最先进的目标检测方案之一[23]，它能够在一幅图像中同时检测和分类对象。

1.2.1　基于 YOLO 的苹果识别

本文提出的基于深度学习 YOLO 的苹果识别方法能够定位视频中的苹果并返回其坐标。不同于基于区域预测的目标检测算法（Faster RCNN）[20]，YOLO 通过单个卷积神经网络检测整个图像回归目标的类别和位置，为了让卷积遍历整幅图像之后能够得到固定格式的预测大小，首先将图像调整为固定大小的尺寸 416×416，再分成 13×13 的非重叠的网格单元，然后每个单元负责检测出该单元可能拥有的 *B* 个边界框及其置信度，具体包含 5 个预测参数：*x*,*y*,*w*,*h* 和置信度。(*x*,*y*)代表目标的坐标，(*w*,*h*)代表目标外包矩形的宽度和高度，置信度用来通过阈值对预测结果进行取舍。

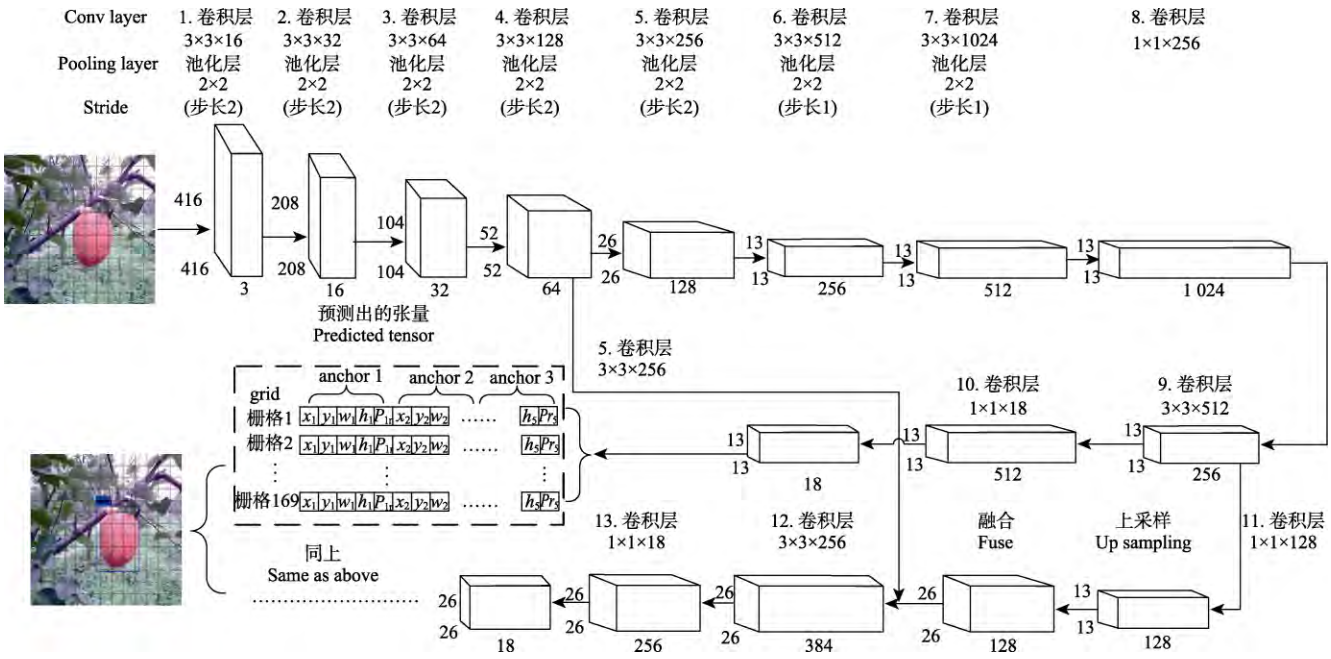
在 YOLOv2[25]中引用锚点(anchor)来提升模型的表现，在 YOLOv3 中使用多尺寸锚点和残差网络来进一步提高模型的表现。

1.2.2　神经网络改进设计

YOLOv3 原文中运用残差网络（ResNet)[26]模型搭建出 53 层的神经网络，考虑到使用过深的网络模型会增加检测时间，文中只需要检测苹果这一类目标，并不需要深层神经网络提取特征，所以本文运用类似于 VGG 的网络模型搭建出 13 层网络。同时 YOLOv3 原文中设置 3 种不同尺寸的锚点，预测这 3 种不同尺寸锚点的网格大小分别是 13、26、52，其中最小预测框的大小则是 8×8（图像尺寸除以网格大小即 416/52），但是实际检测密集苹果的时候只需要检测出大于分辨率 16×16 的苹果，过小的苹果距离太远，无法作为采摘机器人的目标，同时使用 52×52 个预测网格使得预测张量过于庞大增加检测时间，所以本文使用 2 种不同尺度的锚点（13×13,26×26），提高对小物体的检测能力的同时不会增加检测时间。

本文设计的神经网络使用 3×3 的卷积层来提取图像信息，用 2×2 的池化层来降低数据维度，当输入矩阵到

达 52×52×64 大小时并行出另一条神经网络，其第一条神经网络最后输出 13×13×18 的预测张量，第二条神经网络的开始经过 3×3×256 的卷积层得到 26×26×256 矩阵与第一条神经网络的 26×26×128 矩阵进行特征融合，最后输出 26×26×18 的预测张量，最后对 2 个输出结果做并集处理，获得苹果的预测信息，其具体结构如图 2 所示。



注：x、y 为 x 轴、y 轴预测坐标，w 为预测宽度，h 为预测高度，Pr 为预测置信度。

Note: x and y are predicted coordinates in x-axis and y-axis; w is predicted width; h is predicted height; Pr is predicted confidence.

图 2 13 层的 YOLOv3 网络

Fig.2 13-layer YOLOv3 neural network

1.2.3 YOLO 损失函数

本文 YOLO 在训练神经网络时优化的是损失函数，通过 BP（back propagation)不断更新模型，使得 Loss 不断减少。

$$\begin{aligned}\text{Loss Function} = {} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\ & \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] + \\ & \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \\ & \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 + \\ & \sum_{i=0}^{S^2} 1_{\text{i}}^{obj} \sum_{(C \in \textbf{classes})} (p_i(C) - \hat{p}_i(C))^2 \end{aligned} \quad (1)$$

式中 x_i 和 y_i 为预测目标的 x 和 y 轴坐标，$\hat{x}_i$ 和 $\hat{y}_i$ 为实际目标的 x 和 y 轴坐标；w_i 和 h_i 为预测目标的宽和高，$\hat{w}_i$ 和 $\hat{h}_i$ 为实际目标的宽和高；C_i 为预测目标的类别，$\hat{C}_i$ 为实际目标的类别。第四项是算法对背景的置信度的预测。第五项是对物体类别的预测 $Pr(\text{Class}_i|\text{Object})$，$p_i(C)$为预测目标的置信度，$\hat{p}_i(C)$ 为实际目标的置信度；λ_{coord} 和 λ_{noobj} 是为了平衡各项 Loss 的比重所设定的权值，分别为 5 和 0.5。S 为每一个网格单元，B 为每一个边界框，obj 为含有目标，noobj 为不含目标。

2 模型的训练与调整

训练模型所使用的数据集一共 515 张，包含有 3 725 个各种情况下的苹果。其中 400 张用来训练模型，115 张用来验证模型，比例约为 4∶1。

2.1 试验运行平台

本文中所有的训练和测试均在同一台笔记本电脑进行，电脑的主要配置为 PC Intel Core(TM) i7-7700HQ CPU@2.80 GHz、6 GB 的 GPU GeForce GTX 1060 和 16 GB 的运行内存。所有的程序均是由 C++语言编写并调用 CUDA、Cudnn、OPENCV 库并在 Windows10 系统下运行，训练和测试模型时有分别单独使用 CPU 和 GPU。

训练时采用 64 个样本作为一个处理单元，每次更新权值时用 BN（batch normalization）[27]进行正则化，并且在每一层中加入丢弃层（dropout），以保持每层提取特征的相互独立，其丢弃比例为 0.5，动量（Momentum）设置为 0.9，权值衰减（decay）设置为 0.000 5，起始学习率（learning rate）设为 0.001，并在每迭代 5 000 次后衰减 10 倍，并且在一开始为了稳定模型设置了一个 burn_in 参数，其值为 1 000，实际计算如下：

$$\text{学习率} = \text{学习率} \times (\text{迭代次数}/\text{burn\_in})^4 \quad (2)$$

2.2 数据集预处理

在图像中均匀选取不同情况下的苹果，并用矩形框对图像中的苹果进行人工标注用于 YOLO 的训练和测

试。标注信息采用 PASCAL VOC 数据集的格式进行保存，其中包含目标苹果的类别和外包边框。然后进行归一化处理，将目标的实际数据除以图像的宽度和高度，这样得到的数据都在 0~1 的范围之内，使得在训练的时候能够更快地读取数据，并且能够训练不同尺寸的图像，其具体格式为 5 个参数作一组数据，分别包含 index（类别的序列），x（目标中心的 x 坐标），y（目标中心的 y 坐标），w（目标的宽），h（目标的高）。如式 3 所示，其中 x_{max}、y_{max} 为边框右下角坐标，x_{min}、y_{min} 为边框左上角坐标，width、height 为图像宽和高。

$$\begin{cases} x=\dfrac{x_{max}+x_{min}}{2width}, y=\dfrac{y_{max}+y_{min}}{2height} \\ w=\dfrac{x_{max}-x_{min}}{width}, h=\dfrac{y_{max}-y_{min}}{height} \end{cases} \tag{3}$$

实际上深度学习需要大量的数据，实际情况下 500 幅图像远远不够使用。于是本文在训练之前对数据集进行数据增强来增加训练数据[28-29]，其中色调的变化范围从 1 到 1.5 倍，曝光的变化范围从 1 到 1.5 倍，色量的变化范围从 0.9 到 1.1 倍，以此最终生成了 51 500 幅图像以供训练使用。

在训练时还需要用 K-means 聚类计算出当前数据集锚点的预训练数值，其中 K-means 使用的是欧氏距离，13 层网络使用的 6 个锚点为（50×66）、（74×99）、（91×125）、（113×154）、（140×190）、（220×284），在第一条网络上使用前 3 个锚点，第二条网络上使用后 3 个锚点。

2.3　模型的测试与评估

最终模型一共训练 12 000 次，耗时 4 h，一共训练使用了 768 000 幅图像（在 51 500 幅图像中随机抽取并重复使用），其 Loss 变化图如图 3 所示。

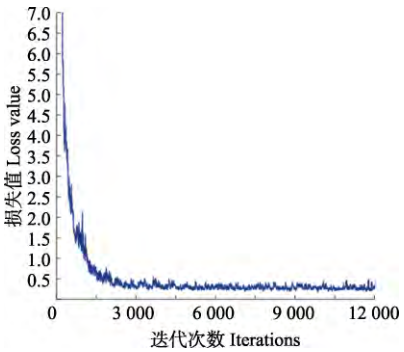


图 3　Loss 值随着迭代次数的变化曲线

Fig.3　Loss value curve changes with iterations

可以看出模型在前 1 000 次迭代中迅速拟合，Loss 快速变小，然后到 4 000 次迭代之后渐渐稳定，只有稍许的振荡。

在训练模型中，平均每隔 100 次迭代输出一次权值（weights），模型的好坏并不是迭代的次数越多越好，过多的训练可能会导致过拟合，所以需要对训练出来的模型进行测试和评估。

在本文中采用客观的评判标准来评估苹果识别系统，本文使用准确率（precision），召回率（recall），mAP（mean average precision）还有 IOU（intersection over union）来评估本文训练出来的模型。并且为模型找出合适的阈值（threshold），通过模型预测的置信度来选择合适的目标。

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{4}$$

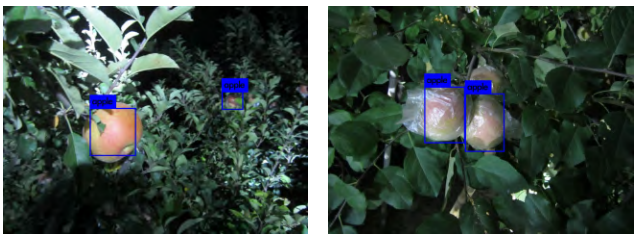
$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{5}$$

$$\text{mAP}=\frac{1}{C}\sum_{k=i}^{N}P(k)\Delta R(k) \tag{6}$$

式中 TP 为真实的正样本数量，FP 为虚假的正样本数量，FN 为虚假的负样本数量，C 为类别的数量，N 为引用阈值的数量，k 为阈值，$P(k)$为准确率，$R(k)$为召回率。

首先找到 mAP 值最高的权值文件，这样就先筛选出整体性能足够高的模型。然后在单独找出的模型中不断地调整阈值，衡量准确率、召回率和 IOU 的值，使之能够按照自身的需求去检测当前环境下的苹果。对于准确率和召回率，在本文的系统中，由于需要找出图中主要的苹果，忽略那些特别小的苹果，所以在准确率和召回率之间优先选择准确率。对于 IOU，因为苹果属于小型目标，本文只需要它的中心点，所以对 IOU 要求也不高。

如果在阈值选择不合适的情况下，就会出现如图 4 的结果，当阈值过低时误将远处的小苹果检测出来，当阈值过高的时候漏掉了后面被部分遮挡的第 3 个苹果。



a. 阈值过低的情况
a. Threshold is too low

b. 阈值过高的情况
b. Threshold is too high

图 4　阈值对检测的影响

Fig.4　Impact of threshold on detection

3　试验数据结果分析

模型训练好后需要对其性能进行评估以找到最优模型，本节用 mAP 值评估模型的整体性能，然后在 mAP 值最高的模型中通过比较不同阈值下准确率、召回率和 IOU 的变化选择模型的最优阈值。之后对实际不同环境下的苹果的检测效果进行试验，并与其他算法进行比较。

3.1　最优模型的寻找

在本文的试验中，训练一共迭代了 12 000 次，每 100 次迭代输出一个模型，所以一共得到 120 个模型，需要在这 120 个模型中找出一个 mAP 值最高的模型。

计算出来的试验结果如图 5 所示，纵坐标为 mAP 值，其范围从 0 到 100%，横坐标为迭代次数，每 100 次迭代即一个模型为一个单位。

从图 5 中可以看出，当迭代次数达到 4 000 次时 mAP

已经稳定，不再发生变化，其中 mAP 最大值为 87.71%，这就是本文选用的模型。

将稳定后的模型选出后，画出其 P-R 曲线如图 6 所示，由于 mAP 很大，所以 PR 曲线将近覆盖了整个坐标系。

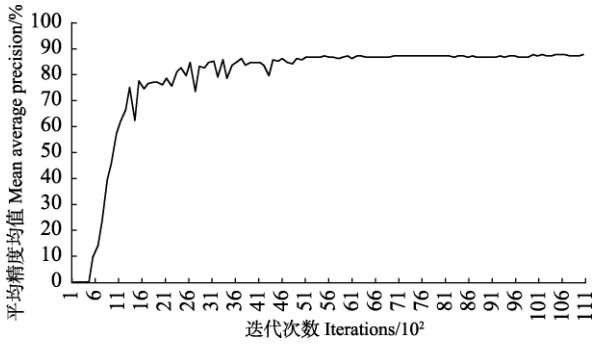


图 5 平均精度均值随迭代次数的变化

Fig.5 Mean average precision changes with iterations

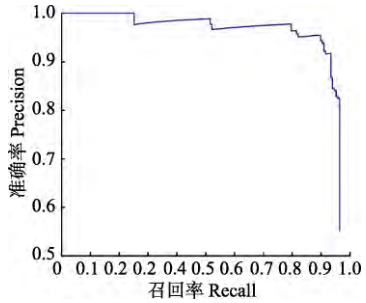


图 6 P-R 曲线

Fig.6 P-R curve

3.2 类别阈值的选择

接下来需要找出本文在实际使用模型时的阈值，算法预测出目标的置信度之后需要使用预设的阈值对其进行筛选。同一个模型下不同的阈值，它所能检测到目标时的准确率、召回率和 IOU 都是不同的，在苹果识别系统中，本文的优先级为准确率>召回率>IOU。同时本文在判断中参照用于辅助的评估值 $F1$[30]，$F1$ 能同时兼顾准确率和召回率：

$$F1=\frac{2\text{Precision}\cdot\text{Recall}}{(\text{Precision}+\text{Recall})} \tag{7}$$

试验结果如图 7 所示，其纵坐标的范围从 0 到 1，横坐标为阈值从 0 到 1 的变化。

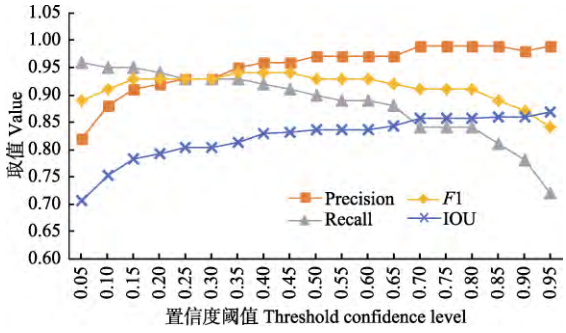


图 7 不同阈值下参数的变化

Fig.7 Changes in various parameters at different thresholds

由图 7 知，阈值选用 0.5，此时的准确率和召回率都达到 97%和 90%，$F1$ 值为 0.93，IOU 达到 83.61%，模型的表现最好。

3.3 实际效果的检验

为了进一步验证模型的有效性，需要在各种实际条件下检测算法的检测效率，本节试验以果实数目、光照角度、果实阶段、采摘时间为控制变量，对比 YOLOv3、YOLOv2、Faster RCNN 和 HOG+SVM[31] 4 种算法在上述条件下的检测效果，并用 $F1$ 进行算法性能的评估。

3.3.1 不同果实数目的对比试验

在机器人实际采摘过程中，随着摄像头距离果树距离的变化，苹果的数目和大小也会随之改变。当苹果数目较少、尺寸较大时，作为识别对象较为清晰完整，识别难度低。但在多目标图像中，由于尺寸的减少和数目的增多，会出现粘连和遮挡的情况，识别难度较大。因此设置不同数目下苹果检测的对比试验，分别分 1 个苹果、多个苹果和密集苹果，对比 4 种算法在不同数目苹果下的检测性能，检测效果如图 8 所示。

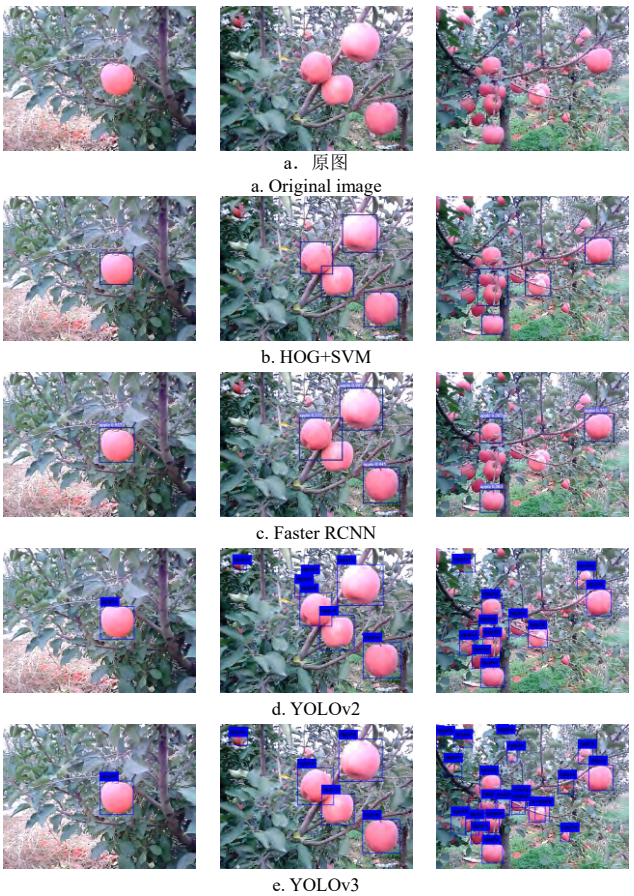


图 8 4 种算法对不同数量苹果的检测效果

Fig.8 Detection effect of 4 algorithms on different numbers of apples

试验测试集一共 336 张图像包含 1 410 个苹果，按照果实数量分为 3 类，其中 1 个苹果 75 幅图像包含 75 个苹果，多个苹果 203 幅图像包含 576 个苹果，密集苹果 58 幅图像包含 759 个苹果，其中密集苹果中最小苹果的分辨率不低于 15×15,尺寸过小忽略不计。每类随机抽取 30 幅图像作为试验测试集，用 4 种不同的算法检测出正

样本数，综合总样本数，未被检测样本数和误检测的样本数，计算出准确率和召回率得出 *F*1 的值。重复以上步骤 3 次，取平均值，最后再将 3 类结果平均得到综合效果，其最终结果如表 1 所示。

从表 1 可以看出，YOLOv3 算法检测效果均高于其余 3 个算法，苹果数目越少，*F*1 值越高，数目由 1 增到 10 个以下时，*F*1 值也没有降低太多。但是遇到密集苹果的时候，*F*1 值降低了将近 20 个百分点。因为密集苹果的图像里摄像头距离苹果的距离过远，导致苹果尺寸不一，同时有粘连和遮挡的情况，导致有一些苹果没有被识别，同时因为卷积神经网络本身固有的特性，不断的卷积会在深层网络丢失高分辨率时小目标的特征，而 Faster RCNN 是基于区域推荐的算法，在 RPN 网络输出矩形候选区域时就已经丢失了小目标的区域，之后进行分类并输出预测结果自然不会出现小苹果。但是 YOLOv3 在浅层网络提前提取出特征用于第二条神经网络的检测，并且输出尺寸更小的栅格预测结果，降低了深层卷积对小目标的影响，因此在密集苹果上高出 YOLOv2 算法 4.45 个百分点。从综合结果来看，YOLOv3 算法能够胜任不同数目苹果的检测。

表 1　4 种算法对不同数量苹果图像的试验结果

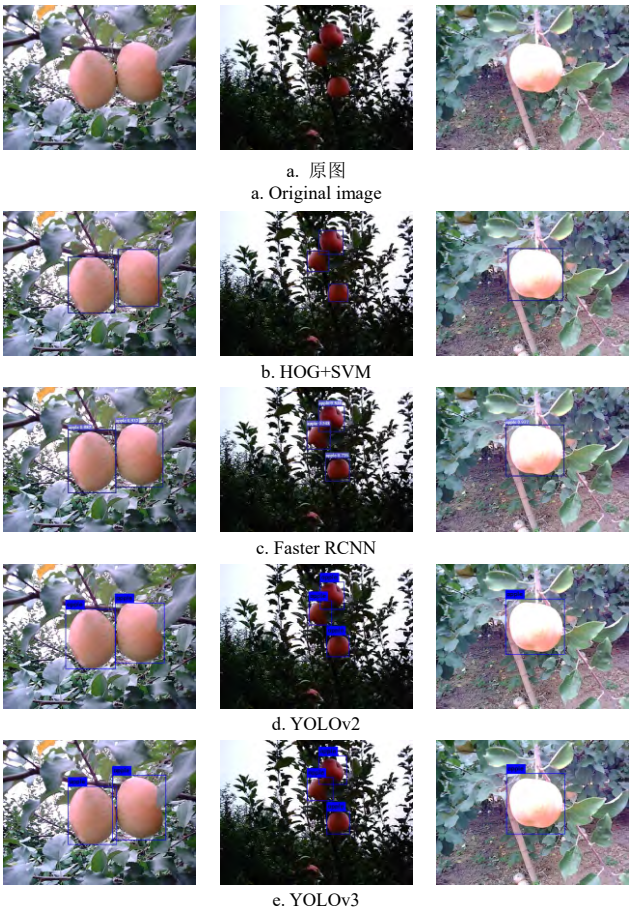
Table 1　Experimental results of 4 algorithms for images with different numbers of apples

苹果个数 Apples number	检测算法 Algorithm	*F*1/%			
		重复 1	重复 2	重复 3	平均值 Average
1	HOG+SVM	86.95	83.94	89.42	86.77
	Faster RCNN	95.23	96.14	95.67	95.68
	YOLOv2	96.56	96.62	95.76	96.31
	YOLOv3	97.17	97.41	96.89	97.15
>2~10	HOG+SVM	83.23	80.57	79.63	81.14
	Faster RCNN	87.49	86.34	86.92	86.91
	YOLOv2	93.17	91.41	93.89	92.82
	YOLOv3	93.67	92.53	94.43	93.54
>10	HOG+SVM	41.43	37.36	39.33	39.37
	Faster RCNN	66.29	68.23	64.85	66.45
	YOLOv2	70.72	69.59	71.79	70.70
	YOLOv3	75.47	72.24	77.74	75.15
平均 Average	HOG+SVM	70.53	67.29	69.46	69.09
	Faster RCNN	83.00	83.57	83.01	83.01
	YOLOv2	86.81	85.87	87.14	86.60
	YOLOv3	88.77	87.39	89.68	88.61

3.3.2　不同光照角度的对比试验

本节试验将拍摄时苹果的光照角度作为控制变量，分别有侧光、逆光和顺光，其中侧光的苹果 124 幅图像 357 个苹果，逆光 67 幅图像 149 个苹果，顺光 87 幅图像 145 个苹果，由于密集苹果的样本影响较大，所以这里选取图像时不会考虑密集苹果样本，其具体效果如图 9 所示，统计结果如表 2 所示。

从图 9 可以看出，苹果在侧光下纹理清楚，表面光照强度均匀，检测难度较小。逆光时，苹果和枝叶的亮度明显降低，并且两者之间分界线也不明显。顺光时，苹果表面一部分亮度增强，脱离苹果本身颜色呈现亮白色，几乎没有纹理特征。后 2 种情况下苹果的检测难度明显增加。



a. 原图
a. Original image

b. HOG+SVM

c. Faster RCNN

d. YOLOv2

e. YOLOv3

图 9　4 种算法对不同光照下苹果的检测效果

Fig.9　Detection effect of 4 algorithms on apples under different illumination

表 2　4 种算法对不同光照下苹果的试验结果

Table 2　Experimental results of 4 algorithms for apples under different illumination

光照角度 Illumination angles	检测算法 Algorithm	*F*1/%			
		重复 1	重复 2	重复 3	平均值 Average
侧光 Side light	HOG+SVM	87.57	85.94	89.42	87.64
	Faster RCNN	96.36	95.76	96.21	96.11
	YOLOv2	97.25	97.38	96.89	97.17
	YOLOv3	98.16	98.36	98.93	98.48
逆光 Back light	HOG+SVM	80.52	78.44	81.69	80.22
	Faster RCNN	85.34	81.49	83.27	83.37
	YOLOv2	94.59	94.32	93.71	94.21
	YOLOv3	95.12	95.39	94.51	95.01
顺光 Natural light	HOG+SVM	86.65	88.74	85.25	86.88
	Faster RCNN	91.43	93.27	89.87	91.52
	YOLOv2	96.28	94.26	97.65	96.06
	YOLOv3	97.10	96.74	97.38	97.07
平均 Average	HOG+SVM	84.91	84.37	85.45	84.91
	Faster RCNN	91.04	90.17	89.78	90.33
	YOLOv2	96.04	95.32	96.08	95.81
	YOLOv3	96.79	96.83	96.94	96.85

从表 2 可以看出,YOLOv3 和 YOLOv2 算法 *F*1 值普遍较高，这 3 种情况下 *F*1 的值没有明显差别，Faster RCNN 算法的 *F*1 值总体比 YOLOv3 低大概 5 个百分点，2 种算法在 3 种情况 *F*1 值梯度一致，这是因为同是深度神经网络强大的泛化能力，能够在颜色特征不一样的情况下依然检测到苹果。逆光时模型表现最差，原因是逆光时缺少亮度，直接消去了部分苹果，使得模型 *F*1 值降低。

3.3.3 不同果实生长阶段的对比试验

本节试验将拍摄时苹果的成长阶段作为控制变量，分别有成熟、未成熟和套袋，其中成熟的苹果 95 幅图像 249 个苹果，未成熟的苹果 84 幅图像 167 个苹果，套袋的苹果 99 幅图像 235 个苹果，同样不考虑密集苹果样本，其具体效果如图 10 所示，统计结果如表 3 所示。

注：从左至右图中分别为未成熟、成熟、套袋苹果。
Note: Apples in image from left to right are Immature, mature and bagged.

图 10 4 种算法对不同阶段苹果的检测效果
Fig.10 Detection effect of 4 algorithms on apples in different stages

从图 10 可以看出，成熟的苹果纹理清晰，颜色与背景明显分离。未成熟的苹果颜色较淡，其颜色与背景相近。套袋的苹果外面都包裹了一层透明薄膜，在光线照射下生成了一个个亮斑，腐蚀了苹果表明的颜色特征和纹理特征，同时塑料薄膜空余的部分多出来让目标整体不再呈现圆形，形状特征也变得模糊。

从表 3 可以看出，YOLOv3 算法的 *F*1 值比 Faster RCNN 高出 3 点百分点，后 3 种算法在 3 种情况下的 *F*1 值差异不大，只有成熟苹果比其他 2 种情况下高了 2~3 个百分点。甚至套袋情况下的苹果 *F*1 值没有发生下降，经分析虽然塑料薄膜让苹果的特征变得模糊，但是薄膜本身的特征也很明显，其与背景相比有很大的差异，这使得套袋苹果实际变得容易识别。猜想算法在同时分辨套袋的苹果和其他水果的时候，才会发生误检测的情况。

表 3 4 种算法对不同生长阶段苹果的试验结果
Table 3 Experimental results of 4 algorithms on different growth stages of apple

苹果阶段 Apple stages	检测算法 Algorithm	*F*1/%			
		重复 1	重复 2	重复 3	平均值 Average
成熟 Ripe	HOG+SVM	92.87	89.21	90.46	90.85
	Faster RCNN	96.32	95.25	96.94	96.17
	YOLOv2	96.67	95.69	95.32	95.89
	YOLOv3	97.17	96.80	96.72	96.89
未成熟 Immature	HOG+SVM	86.41	87.74	86.19	86.78
	Faster RCNN	93.12	94.45	94.37	93.98
	YOLOv2	94.03	95.26	94.63	94.64
	YOLOv3	95.92	96.12	95.61	95.88
套袋 Bagged	HOG+SVM	78.61	81.98	77.28	79.29
	Faster RCNN	91.85	89.38	87.49	89.57
	YOLOv2	93.62	92.54	94.25	93.46
	YOLOv3	95.86	96.51	96.34	96.24
平均 Average	HOG+SVM	85.96	86.31	84.64	85.64
	Faster RCNN	93.76	96.47	92.93	93.24
	YOLOv2	94.77	94.49	94.73	94.66
	YOLOv3	96.31	96.47	96.22	96.33

3.3.4 不同采摘时间的对比试验

本节试验将拍摄时苹果的拍摄时间作为控制变量，分别有白天、傍晚和夜晚，其中白天的苹果 131 幅图像 306 个苹果，傍晚的苹果 71 幅图像 193 个苹果，夜晚的苹果 76 幅图像 152 个苹果，同样不考虑密集苹果样本，试验方法与上节相同，其具体效果如图 11 所示，统计结果如表 4 所示。

从图 11 可以看出，苹果在白天表现正常，其颜色和纹理特征都表现完好。傍晚时苹果特征与逆光时相似，这里就不再赘述。夜晚时，由于光源的缺少，不同于白天逆光时还有漫反射光对苹果的照射，苹果大部分部位都呈现黑色，这种情况下苹果无论颜色、性质、或者纹理特征都已近乎没有，仅仅有残缺的被照到一部分的苹果。

从表 4 可以看出，后 3 种算法在不同时间的检测效果差不多，并且都在白天时效果最佳，同时光线越暗，*F*1 值越小。但是在夜晚下相比白天下降了 10 个百分点，有了大幅下降，经分析，因为亮度偏低，不仅苹果表面变暗，背景及其枝叶也变得很暗，这让苹果一部分几乎融入背景，没有区别。而且辅助用的人工光源的探照范围有限，被照射的部分与没有被照射的部分相比亮度差异更大，使没有被照射的苹果更加难以被识别。

综上所述，HOG+SVM 算法虽然在各种环境下都能检测到苹果，但是 *F*1 值普遍较低，本文改进的 YOLOv3 在各种情况下均优于 Faster RCNN，与 YOLOv2 相比虽然相差不大（但也高出 1～2 个百分点），但是密集苹果情况下明显优于 YOLOv2。4 组对比试验证明文本的算法

能够适应绝大多数情况下苹果的目标检测，使得搭载此视觉算法的果树采摘机器人的全天候工作变得可能。

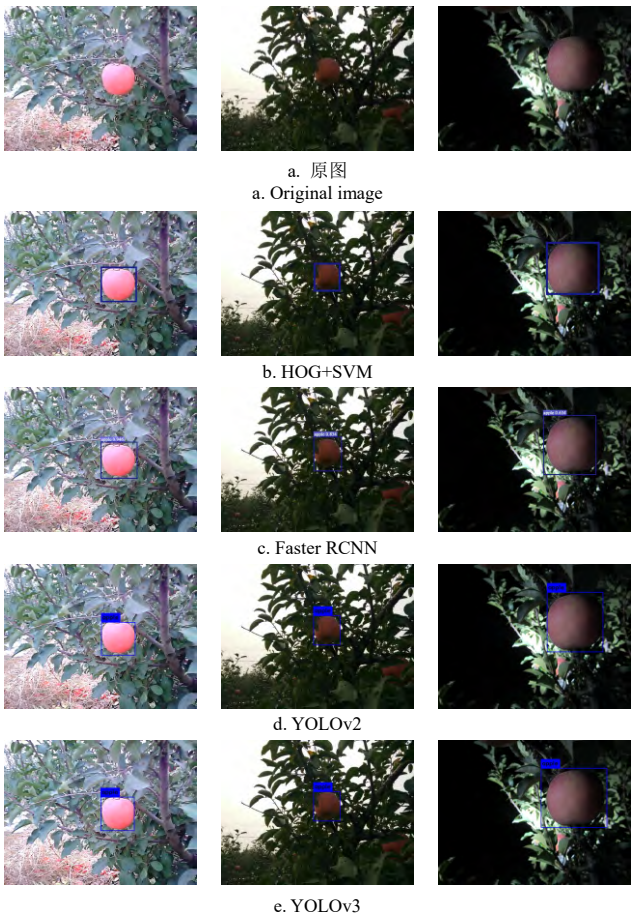


图 11　4 种算法对不同时间苹果的检测效果

Fig.11　Detection effect of 4 algorithms on apples at different times

表 4　4 种算法对不同时刻苹果的试验结果

Table 4　Experimental results of 4 algorithms for apples at different time

时间 Times	检测算法 Algorithm	*F*1/%			
		重复 1	重复 2	重复 3	平均值 Average
白天 Daytime	HOG+SVM	90.57	88.45	92.93	90.65
	Faster RCNN	96.48	95.27	95.36	95.70
	YOLOv2	94.64	97.03	95.28	95.65
	YOLOv3	97.11	97.41	96.86	97.12
傍晚 Nightfall	HOG+SVM	85.17	81.28	83.24	83.23
	Faster RCNN	92.75	91.24	90.63	91.54
	YOLOv2	93.82	93.35	92.57	93.25
	YOLOv3	95.38	94.33	93.90	94.53
夜晚 Night	HOG+SVM	73.51	76.31	77.85	75.89
	Faster RCNN	85.36	83.87	88.91	86.04
	YOLOv2	88.60	89.79	86.39	88.26
	YOLOv3	87.22	91.48	85.81	88.17
平均 Average	HOG+SVM	83.08	82.01	84.67	83.26
	Faster RCNN	91.53	90.12	91.63	91.09
	YOLOv2	92.35	93.39	91.41	92.39
	YOLOv3	93.23	94.40	92.19	93.28

3.4　实际检测时间的检验

除了评估算法的准确率，还需要测试算法在实际检测目标时所用时间，由于 YOLOv3 算法在做卷积运算前需要对图像进行缩放到固定尺寸，所以不同分辨率的图像在本文算法下检测时间的差异只与缩放过程所用时间有关，其余流程并没有区别。本文试验测试用图像分辨率均为 640×480，考虑到苹果识别系统是搭载在果树机器人的工控机之中的，有的工控机只有 CPU，为了减少硬件成本，需要模型能够在单 CPU 的情况下运行，所以本文分别在 GPU 和 CPU 下进行了测试，检测的精度没有变化，检测时间上 CPU 较 GPU 变长。

模型在 GPU 下检测 115 幅图像消耗 1.92 s，在 CPU 下检测 115 幅图像消耗 12.1 s，即为 60 帧/s 和 15 帧/s，平均每幅图像检测时间为 16.69 ms 和 105.21 ms。由于在采摘流程里只在开始的初始定位，在采摘过程中并不需要频繁刷新，所以能够满足实际要求。

4　结　论

本文提出了基于深度神经网络 YOLOv3 算法的苹果识别方法。试验表明该模型的检测精度高、速度快、在复杂环境下鲁棒性高，选用的 13 层网络在真实环境下进行检测，平均每幅图像在 GPU 和 CPU 下的检测时间为 16.69 和 105.21 ms，准确率和召回率分别达到 97%和 90%以上。

1）针对果园中复杂的识别环境，提出用优化的 YOLOv3 深度神经网络定位苹果，与 YOLOv2 相比在密集苹果条件下 *F*1 值高出 4.45 个百分点，其余条件下高出 1~2 个百分点。

2）对比了 HOG+SVM、Faster RCNN、YOLOv2 和 YOLOv3 在不同果实数目、光照角度、果实阶段、采摘时间下的检测结果，其 *F*1 值总体比 HOG+SVM 高出约 10 个百分点，比 Faster RCNN 高出约 5 个百分点，本文使用算法检测精度优势更为显著。

3）由于深度学习需要大量的计算能力，在实际使用中还需要考虑功耗和稳定问题，因此未来还需要在实际机器上运行试验，在不损失精度的情况下缩减网络模型，减少功耗。

[参　考　文　献]

[1] Kapach K, Barnea E, Mairon R, et al. Computer vision for fruit harvesting robots–state of the art and challenges ahead[J]. International Journal of Computational Vision and Robotics, 2012, 3(1/2): 4－34.

[2] 吕继东，赵德安. 苹果采摘机器人目标果实快速跟踪识别方法[J]. 农业机械学报，2014，45(1)：65－72.
Lu Jidong, Zhao De'an. Fast tracking and recognition method for target fruits of apple picking robots[J]. Transactions of the Chinese Society for Agricultural Machinery, 2014, 45(1): 65－72. (in Chinese with English abstract)

[3] 赵德安，刘晓洋，陈玉，等. 苹果采摘机器人夜间识别方法[J]. 农业机械学报，2015，46(3)：15－22.
Zhao Dean, Liu Xiaoyang, Chen Yu, et al. Night recognition

method of apple picking robot[J]. Transactions of the Chinese Society for Agricultural Machinery, 2015, 46(3): 15－22. (in Chinese with English abstract)

[4] Ji W, Zhao D, Cheng F, et al. Automatic recognition vision system guided for apple harvesting robot[J]. Computers & Electrical Engineering, 2012, 38(5): 1186－1195.

[5] Stajnko D, Lakota M, Hočevar M. Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging[J]. Computers and Electronics in Agriculture, 2004, 42(1): 31－42.

[6] Wachs J P, Stern H I, Burks T, et al. Low and high-level visual feature-based apple detection from multi-modal images[J]. Precision Agriculture, 2010, 11(6): 717－735.

[7] Rakun J, Stajnko D, Zazula D. Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry[J]. Computers and Electronics in Agriculture, 2011, 76(1): 80－88.

[8] Aggelopoulou A D, Bochtis D, Fountas S, et al. Yield prediction in apple orchards based on image processing[J]. Precision Agriculture, 2011, 12(3): 448－456.

[9] 宋怀波，张卫园，张欣欣，等. 基于模糊集理论的苹果表面阴影去除方法[J]. 农业工程学报，2014，30(3)：135－141.
Song Huaibo, Zhang Weiyuan, Zhang Xinxin, et al. Apple surface shadow removal method based on fuzzy set theory[J]. Transactions of the Chinese Society of Agricultural Engineering(Transactions of the CSAE), 2014, 30(3): 135－141. (in Chinese with English abstract)

[10] Kurtulmus F, Lee W S, Vardar A. Green citrus detection using ‘eigenfruit’, color and circular Gabor texture features under natural outdoor conditions[J]. Computers and Electronics in Agriculture, 2011, 78(2): 140－149.

[11] Linker R, Cohen O, Naor A. Determination of the number of green apples in RGB images recorded in orchards[J]. Computers and Electronics in Agriculture, 2012, 81: 45－57.

[12] Arivazhagan S, Shebiah R N, Nidhyanandhan S S, et al. Fruit recognition using color and texture features[J]. Journal of Emerging Trends in Computing and Information Sciences, 2010, 1(2): 90－94.

[13] Xu Y, Imou K, Kaizu Y, et al. Two-stage approach for detecting slightly overlapping strawberries using HOG descriptor[J]. Biosystems engineering, 2013, 115(2): 144－153.

[14] 卢军，桑农. 变化光照下树上柑橘目标检测与遮挡轮廓恢复技术[J]. 农业机械学报，2014，45(4)：76－81.
Lu Jun, Sang Nong. Detection of citrus targets and restoration of concealed contours in trees under changing light[J]. Transactions of the Chinese Society for Agricultural Machinery, 2014, 45(4): 76－81. (in Chinese with English abstract)

[15] Zhao C, Lee W S, He D. Immature green citrus detection based on colour feature and sum of absolute transformed difference (SATD) using colour images in the citrus grove[J]. Computers and Electronics in Agriculture, 2016, 124: 243－253.

[16] 谢忠红，姬长英，郭小清，等. 基于改进 Hough 变换的类圆果实目标检测[J]. 农业工程学报，2010，26(7)：157－162.
Xie Zhonghong, Ji Changying, Guo Xiaoqing, et al. Target detection of fruit-like fruit based on improved Hough transform[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2010, 26(7): 157－162. (in Chinese with English abstract)

[17] 马翠花，张学平，李育涛，等. 基于显著性检测与改进 Hough 变换方法识别未成熟番茄[J]. 农业工程学报，2016，32(14)：219—226.
Ma Cuihua, Zhang Xueping, Li Yutao, et al. Identification of immature tomatoes based on saliency detection and improved Hough transform method [J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2016, 32(14): 219—226. (in Chinese with English abstract)

[18] Girshick R, Donahue J, Darrell 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. 2014: 580－587.

[19] Girshick R. Fast R-CNN[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440－1448.

[20] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[C]// Advances in neural information processing systems. 2015: 91－99.

[21] 熊俊涛，刘振，汤林越，等. 自然环境下绿色柑橘视觉检测技术研究[J]. 农业机械学报，2018，49(4)：45－52.
Xiong Juntao, Liu Zhen, Tang Linyue, et al. Research on green citrus vision detection technology in natural environment[J]. Transactions of the Chinese Society for Agricultural Machinery, 2018, 49(4): 45－52. (in Chinese with English abstract)

[22] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21－37.

[23] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779－788.

[24] 薛月菊，黄宁，涂淑琴，等. 未成熟芒果的改进 YOLOv2 识别方法[J]. 农业工程学报，2018，34(7)：173—179.
Xue Yueju, Huang Ning, Tu Shuqin, et al. Immature mango detection based on improved YOLOv2 [J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2018, 34(7) :173—179. (in Chinese with English abstract)

[25] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017: 6517－6525.

[26] 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. 2016: 770－778.

[27] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International Conference on Machine Learning.

2015: 448－456.
[28] Bargoti S, Underwood J. Deep fruit detection in orchards[C]// IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 3626－3633.
[29] Ding W, Taylor G. Automatic moth detection from trap images for pest management[J]. Computers and Electronics in Agriculture, 2016, 123: 17－28.
[30] Hripcsak G, Rothschild A S. Agreement, the f-measure, and reliability in information retrieval[J]. Journal of the American Medical Informatics Association, 2005, 12(3): 296－298.
[31] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005: 886－893.

Apple positioning based on YOLO deep convolutional neural network for picking robot in complex background

Zhao Dean[1], Wu Rendi[1], Liu Xiaoyang[1], Zhao Yuyan[2]
(1. *School of Electrical and Information Engineering, Jiangsu University, Zhenjiang* 212013, *China*;
2. *School of Electronic and Electrical Engineering, Changzhou Institute of Information Technology, Changzhou* 213164, *China*)

Abstract: Automatic recognition of apple is one of the important aspects for apple harvest robots. Fast apple recognition can improve the efficiency of picking robots. In the actual scene of the orchard, the recognition conditions for apple are complex such as daytime, night, overlap apples, occlusion, bagged, backlighting, reflected light and dense apple, considering which a highly robust and fast visual recognition scheme is required. A fast and stable apple recognition scheme was proposed based on improved YOLOv3 in this paper. The entire image was traversed by a single convolutional neural network (one-stage), dividing an image into a plurality of sub-regions with the same size, and predicting the class of the target and its bounding box in each sub-region. Finally, the non-maximum value suppression was merged into the outer frame of the whole target, and the category and position of the target were returned. In order to improve the detection efficiency, the VGG-like network model was used to replace the original residual network of YOLOv3, and the model size was reduced, in which the 53-layer neural network was compressed into a 13-layer neural network without affecting the detection effect. Taking into account the size of the smallest apple in dense apples images, the anchor points of 3 different sizes were reduced to 2, reducing the final predicted tensor and ensuring that the smallest anchor point could still include the minimum target. The steps in this paper were stated as follows: Firstly, the data set was manually marked, including 400 images for the training set and 115 images for the verification set, including a total of 1 158 apple samples. In addition, in order to increase the generalization ability of the model, the data set was enhanced by adjusting the hue, color amount and exposure of the image, and a total of 51 500 images were generated. Then the initial value of the anchor points was calculated through K-means. Secondly, training the data set, output a model every 100 iterations. For the verification set, the mean average precision (mAP) value of each weight in batches was calculated, selecting the model with the highest mAP value, and finding the appropriate threshold to ensure most preferred precision, recall rate and intersection over union (IOU). The trained model had a mAP which reached up to 87.71%, an accuracy rate up to 97%, a recall rate up to 90%, and an IOU up to 83.61%. Thirdly, the specific performance of the model under image conditions for different fruit number, illumination angle, fruit growth stage and shooting time were verified in additional experimental data sets. The experimental data set consisted of 336 pictures containing 1 410 apple samples. The comparison was performed with algorithms of HOG+SVM, Faster RCNN, YOLOv2, and YOLOv3, with the evaluated index of $F1$ value. The experimental results showed that YOLOv3 performed significantly better than YOLOv2 in dense apples image, and better in other environments than Faster RCNN and HOG+SVM. Finally, the detection accuracy of the algorithm was verified in different hardware environments. The detection time of an image under the GPU was 16.69 ms with 60 frame/s for the actual video, and under the CPU was 105.21 ms with 15 frame/s for the actual video. Since it was positioned only at the beginning of the picking process and it did not require frequently refreshing during the picking process, in which the detection time in this paper was qualified. A reference was provided for the rapid, long-term high efficiency of robots to locate apples in complex environments in this research.

Keywords: harvesters; machine vision; image recognition; deep learning; picking robot; apple recognition; YOLO