

# Fruit and Vegetable Identification System Based on Deep Learning

Tong Xue, Hui Chen, Na Xu

College of Information, Zhejiang Sci-Tech University, Hangzhou Zhejiang  
Email: 769895711@qq.com, 975005846@qq.com, 1733545914@qq.com

Received: Sep. 20<sup>th</sup>, 2019; accepted: Oct. 5<sup>th</sup>, 2019; published: Oct. 12<sup>th</sup>, 2019

---

## Abstract

In the processing of sorting agricultural products, the classification of fruits and vegetables mostly stays in the traditional classification mode. This mode increases the labor cost on the one hand and the inconvenience to the workers on the other hand. Thanks to the continuous development of computer vision technologies, replacing labor with machines has gradually become possible. The paper studies the model structure and principle used in the development of fruit and vegetable recognition systems. Tensorflow is applied as core deep learning framework and OpenCV is used as in image processing. Based on the collected images of fruits and vegetables, a model with high accuracy was obtained through multiple adjustments and training. Image datasets, model parameter adjustment, training results and application performance are visualized and analyzed in the system. Four functional modules of image recognition, object detection, semantic segmentation and real-time detection are realized in the Web APP.

## Keywords

Image Recognition, Object Detection, Semantic Segmentation, Real-Time Detection

---

# 基于深度学习的果蔬识别系统

薛桐, 陈辉, 徐娜

浙江理工大学信息学院, 浙江 杭州  
Email: 769895711@qq.com, 975005846@qq.com, 1733545914@qq.com

收稿日期: 2019年9月20日; 录用日期: 2019年10月5日; 发布日期: 2019年10月12日

---

## 摘要

在农产品分拣过程中, 果蔬的分类大多还停留在传统的分类模式, 由人工进行果蔬筛选, 这不但增加了

成本,还可能给劳动者带来不便的体验。得益于计算机视觉相关技术的不断发展,以机器代替人工劳动的运营模式逐渐成为可能。本文探讨了以Tensorflow深度学习框架为核心,利用OpenCV进行图像处理的果蔬识别系统。基于收集的果蔬图像数据,通过多次调参与训练获得了准确率较高的模型。在系统构建上对图像数据集的使用、模型参数调整、训练结果以及应用性能进行了可视化设计与分析,通过Web App实现并展示了图像识别、目标检测、语义分割、实时检测四大功能模块。

## 关键词

图像识别, 目标检测, 语义分割, 实时检测

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着社会信息化程度的提升和 AI 技术的广泛普及,人们对美好生活的向往迫切要求社会向智能化、便捷化发展[1]。计算机视觉技术在人民生活的各方各面起着越来越重要的作用,果蔬识别技术作为其中的一环,与商品零售、智能农业、智慧医疗等方向密切联系,既可有效减少劳动力开销,又能使人民在享受便捷精致生活服务的同时得到膳食引导[2],其广泛需求推动着相关领域的发展。

果蔬识别技术凭借其高效便捷的特性广泛应用于智能农业和产品分拣中。卡内基梅隆大学机器人学院科学家已提出建立农业机器人队伍来实现对农作物生产的辅助管理,通过 AI 技术判断农作物的成熟度等特性[3];比利时已通过视觉识别技术识别成熟果蔬并采摘,对未成熟果蔬的成熟时间给以预测[4];中国部分地区在茶叶采摘过程中利用视觉技术识别茶树嫩芽并实现定位及采摘以保证叶片完整性。

本文以果蔬识别为中心,探索智能分类功能的实现与应用,并结合基于深度学习的计算机视觉技术与传统互联网技术进行项目应用与展示,完成一个较为完善的果蔬识别系统。

## 2. 数据集

论文研究中通过数据增强技术取得 2500 张源自自行拍摄的果蔬图像作为主要训练集,16854 张源自 Kaggle 平台 Fruits360 dataset 中的果蔬图像作为辅助训练集。

考虑到图像像素大小对识别的准确率会产生一定的影响,需要选择较为合适的像素大小对图像进行输入来达到提高准确率的目的[5]。由于所用计算机性能一般,为提升训练速度并使后期调用模型能够有较快的速度,选用像素大小分别为  $299 \times 299$ 、 $300 \times 300$ 、 $400 \times 400$ 、 $500 \times 500$ 、 $600 \times 600$ 、 $700 \times 700$  的图像数据集测试图像像素大小对识别准确率的影响,验证结果如图 1 所示,对比分析选择大小为  $400 \times 400$  像素的图像数据集作为开发数据集。

## 3. 图像识别

如图 2 所示[6],图像识别我们采用了常规卷积模型和 Inception v3 模型。在常规卷积模型构建上选用辅助训练集进行训练,包含三类不同品种苹果、黄桃、樱桃、荔枝、山竹、橘子、梨、火龙果共 10 类果蔬图像共 16854 张图像,其中 80% 作为训练集,20% 作为测试集。由于数据量较多,训练所需时间较长,考虑到计算机性能、时间因素等多方面原因,论文采用的网络较为简单,使用综合性能较好的

AdamOptimizer 作为优化器，分别采用神经元为  $32 \times 64$ 、 $64 \times 16$  的卷积层，使用  $3 \times 3$  卷积核，步长为 2 的 max\_pool，128 个神经元的全连接层和 softmax 回归层进行实现。

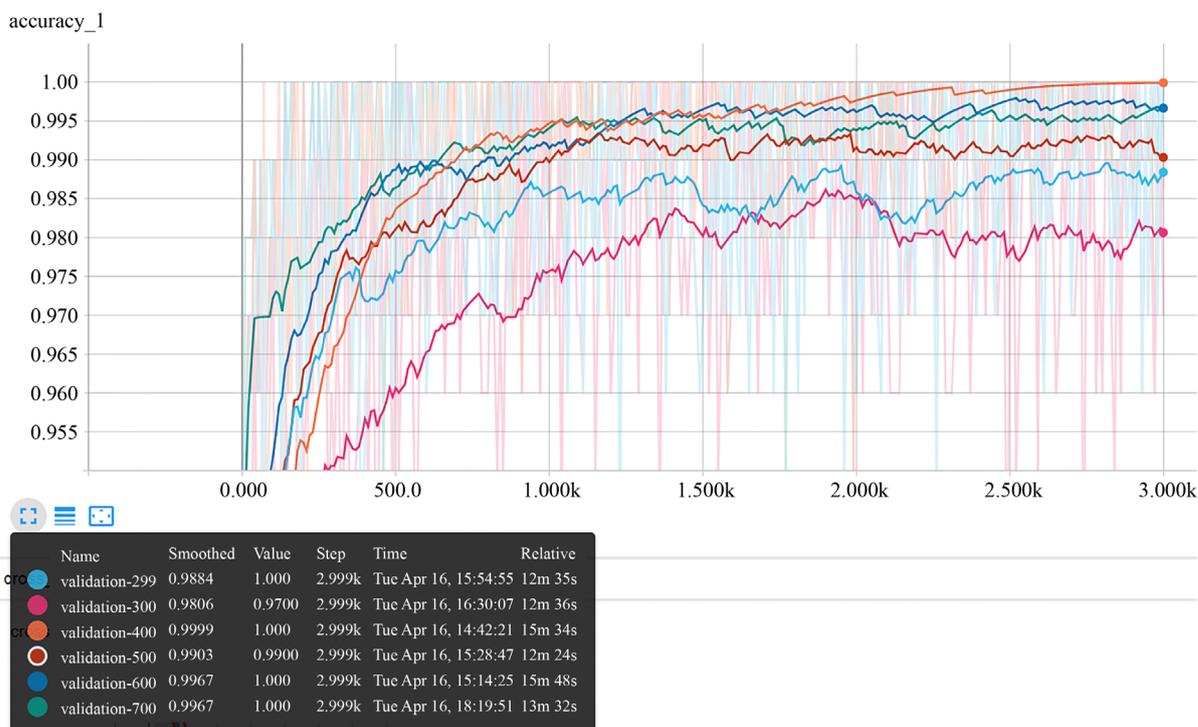


Figure 1. Effect sits on verification accuracy of images of different pixel sizes

图 1. 不同像素大小图像对验证准确率影响图

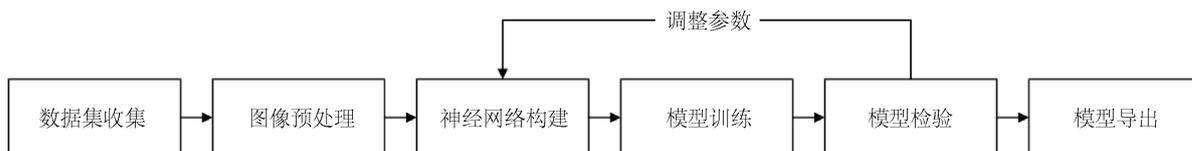


Figure 2. General convolution model feature implementation flowchart

图 2. 常规卷积模型功能实现流程图

根据如图 3 训练结果所示，64-16 结构能比 32-64 结构取得更好的效果，在测试时二者训练集准确率分别为 0.9654 和 0.9318，测试集准确率分别为 0.9231 和 0.8990，选择 64-16 结构进行后续优化，分别使用值为 0.9、0.85、0.8 的 dropout 来降低过拟合。如表 1 所示当 dropout 值为 0.9 时效果较好，既提升了测试集准确率，又缩小了测试集与训练集准确率的差距，降低了过拟合。

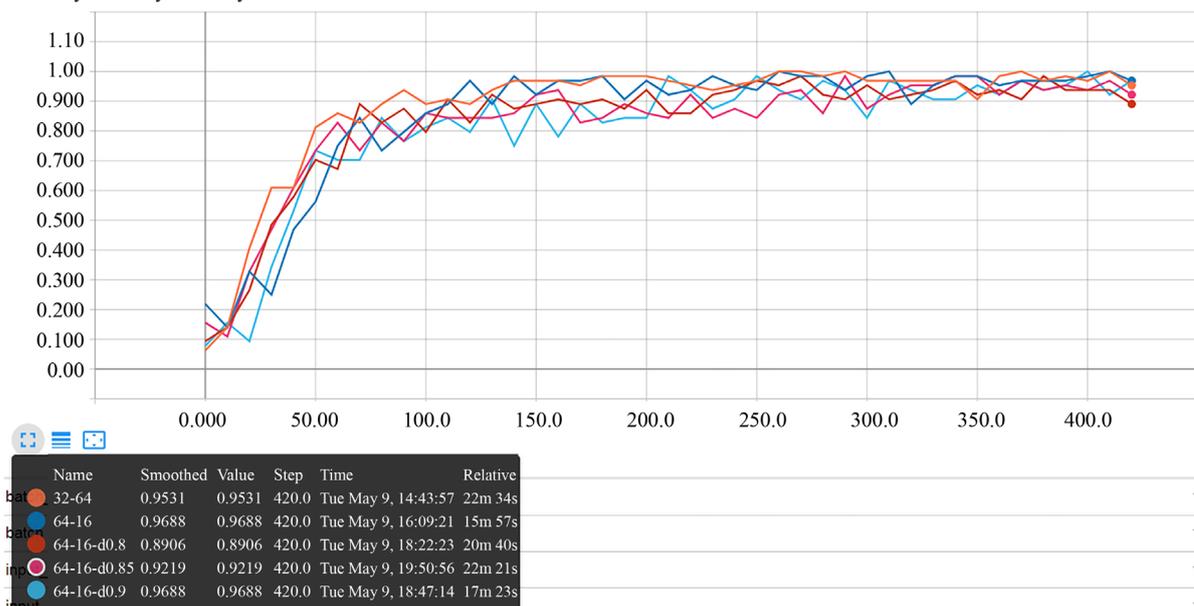
在 Inception v3 模型构建上选用主要数据集进行训练，包含香蕉、橘子、黄桃、柠檬、梨子五种较为相近的水果共 2500 张图像，将 80% 的数据集作为训练集，10% 的数据集作为验证集，10% 的数据作为测试集。

模型构建所使用的优化器分别选用较为常见的 Gradient Descent Optimizer 和 Adam Optimizer。其中 Gradient Descent Optimizer 分别采用值为 0.005、0.01 的学习率训练，Adam Optimizer 分别采用值为 0.0001、0.0003、0.0005 的学习率进行训练，训练集训练情况如图 3 所示。当训练 3000 步时 Adam-0.0005 结构和 Adam-0.0003 结构准确率较为突出且较快达到收敛，而两个 GradientDescent 结构均训练效果较差。

**Table 1.** Convolution model effect comparison**表 1.** 卷积模型效果对比

| 结构     | 32-64  | 64-16  | 64-16-d0.9 | 64-16-d0.85 | 64-16-d0.8 |
|--------|--------|--------|------------|-------------|------------|
| 训练集准确率 | 0.9318 | 0.9651 | 0.9601     | 0.9388      | 0.9414     |
| 测试集准确率 | 0.8990 | 0.9231 | 0.9349     | 0.9133      | 0.8720     |

accuracy/accuracy/accuracy

**Figure 3.** General convolution model training accuracy diagram**图 3.** 常规卷积模型训练准确率展示图

如图 4 所示, 在训练 3000 步时验证集中各结构均基本收敛。Adam 结构网络在该案例上明显优于 Gradient Descent 结构网络, 均达到了较高的准确率, 由于学习率的不同达到收敛时间有所差异。如图 5 所示。在测试集上测试最终结果时, Adam-0.0001 结构、Adam-0.0003 结构、Adam-0.0005 结构均达到了 98.8% 准确率, GradientDescent-0.005 结构和 GradientDescent-0.01 结构均达到 98.4% 准确率。最终选择 Adam-0.0005 结构生成的模型完成系统应用功能开发。

#### 4. 目标检测

分析测试中使用了 Faster RCNN 模型[7]和 SSD 模型[8], 通过对二者进行分析, 选用综合时间、mAP 性能较好的 faster\_rcnn\_inception\_v2\_coco 模型、ssd\_mobilenet\_v1\_coco 模型和 ssd\_inception\_v2\_coco 模型作为预训练模型对主要数据集进行测试, 在 loss 值趋势不再降低时停止训练。

如图 6、图 7 所示, ssd\_mobilenet\_v1 结构的 loss 相比 ssd\_inception\_v2 结构的 loss 值较低, 训练效果相比较好, 因此选用 ssd\_mobilenet\_v1 结构进行后续对比。

如图 8 所示, 对 ssd\_mobilenet\_v1 结构与 faster\_rcnn\_inception\_v2 结构进行目标检测结果对比, faster\_rcnn\_inception\_v2 结构的 loss 值明显低于 ssd\_mobilenet\_v1 结构的 loss 值。如图 9 所示, 准确率测试结果中 faster\_rcnn\_inception\_v2 结构也取得了明显的优势。这两个模型的检测准确率有一定差异, 但基于 Faster RCNN 和 SSD 的基本特性, 该 SSD 模型虽然准确率较低, 但在模型调用速度上有一定优势,

在实时性和 mAP 综合性能上可能超过该 Faster RCNN 模型。

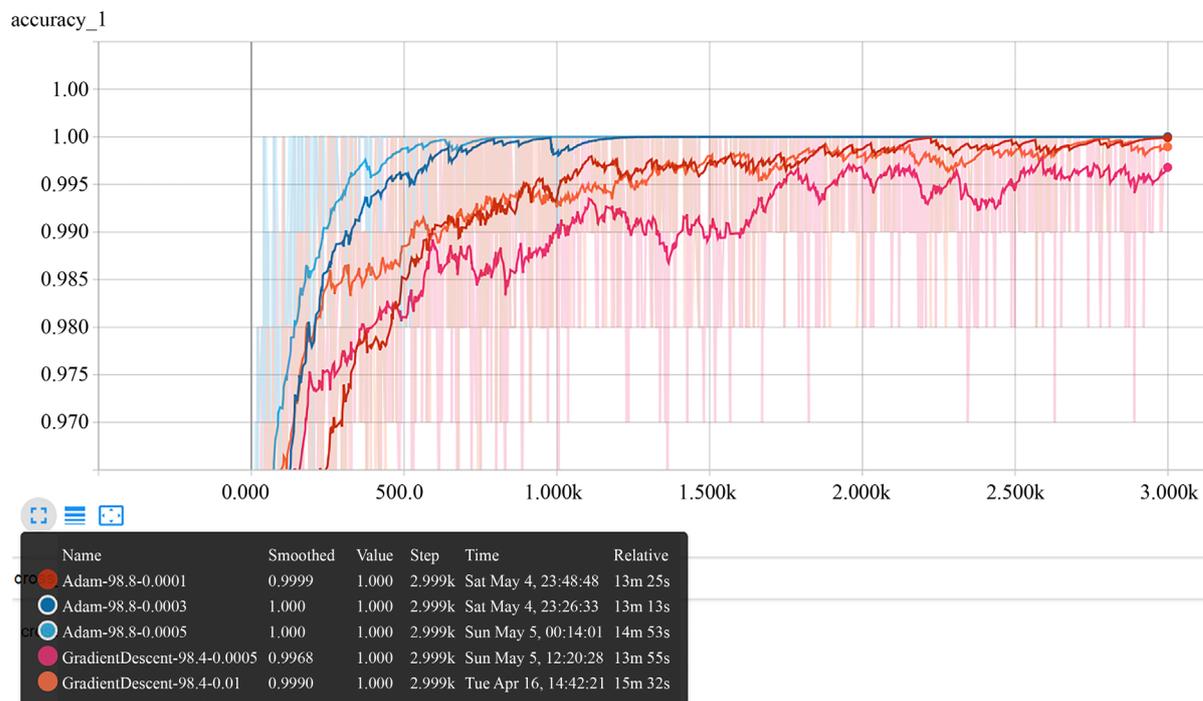


Figure 4. Inception v3 training accuracy trends

图 4. Inception v3 训练准确率趋势图

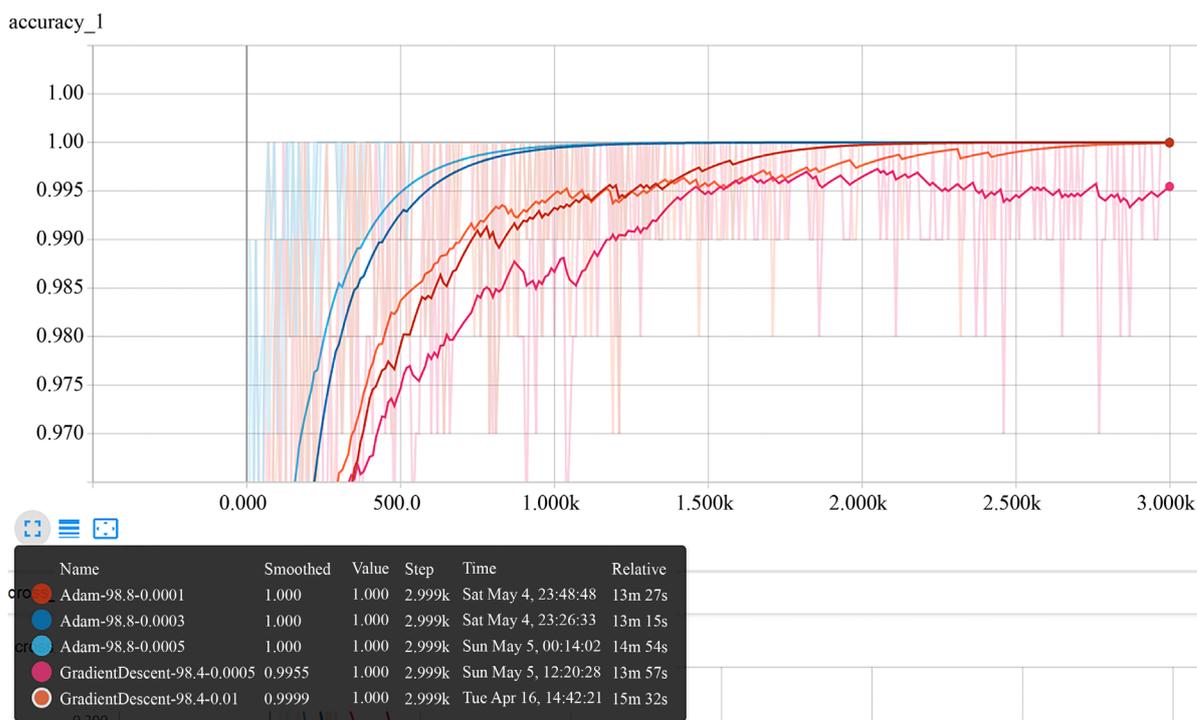


Figure 5. Inception v3 verification accuracy trend chart

图 5. Inception v3 验证准确率趋势图

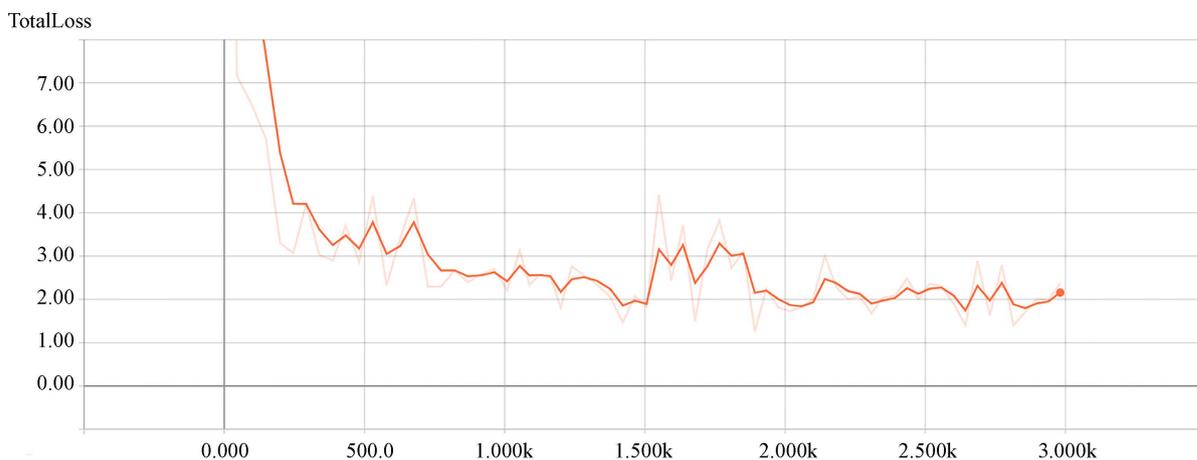


Figure 6. ssd\_mobilenet\_v1 loss trend chart  
图 6. ssd\_mobilenet\_v1 loss 趋势图

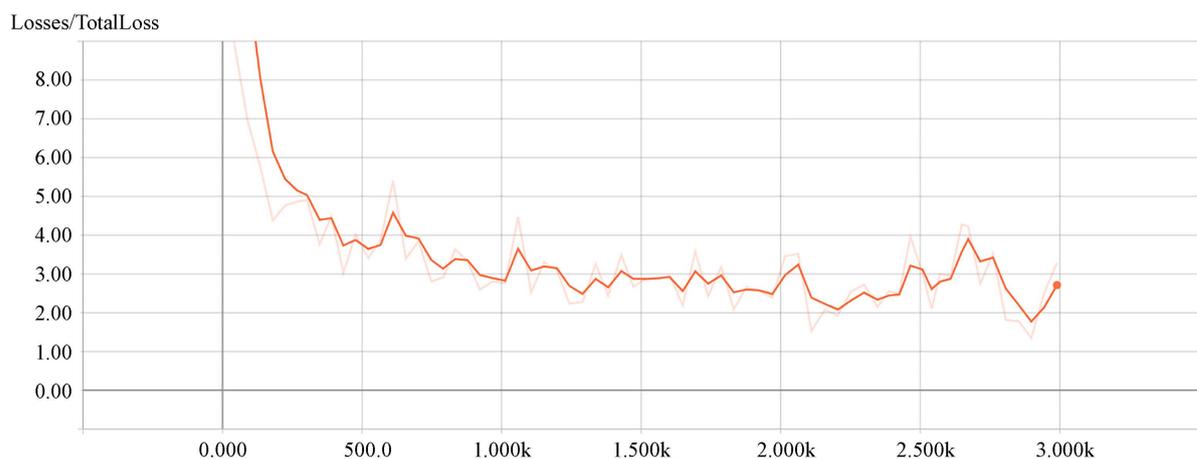


Figure 7. ssd\_mobilenet\_v2 loss trend chart  
图 7. ssd\_inception\_v2 loss 趋势图

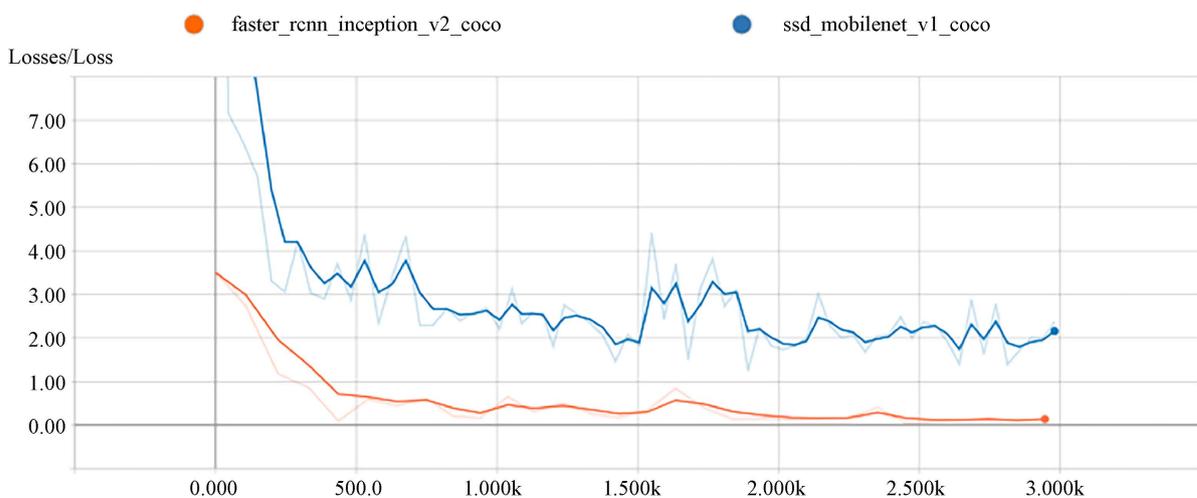


Figure 8. Target detection loss comparison  
图 8. 目标检测 loss 对比图

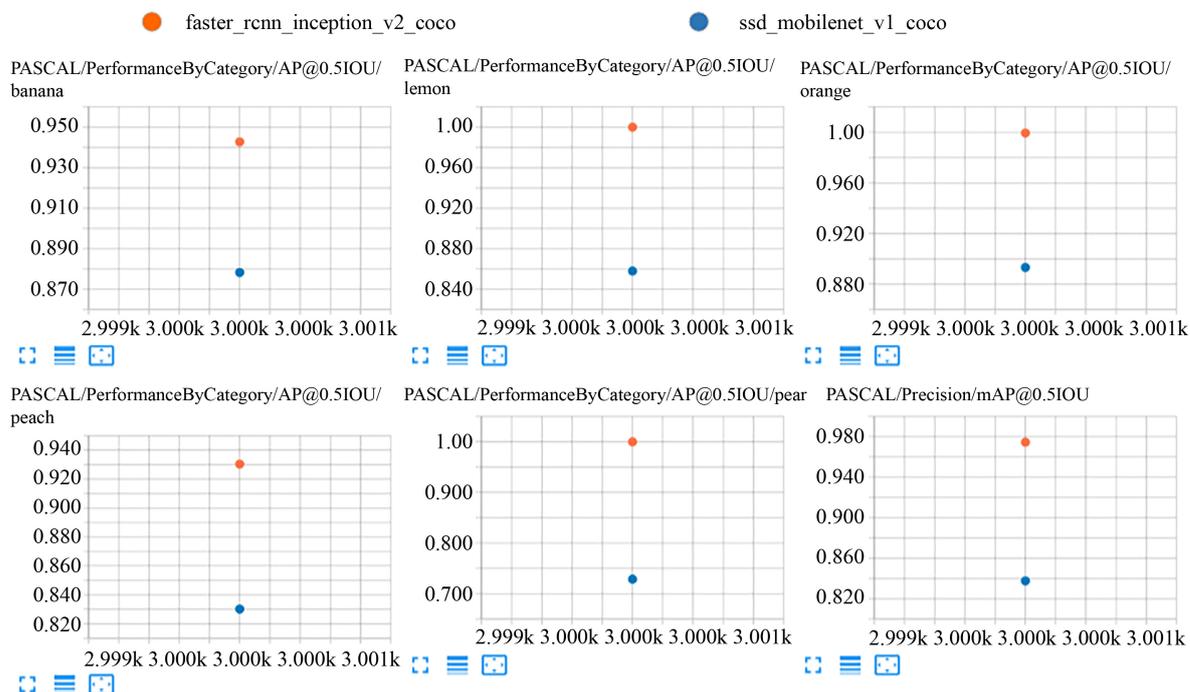


Figure 9. Target detection model accuracy comparison

图 9. 目标检测模型准确率对比图

## 5. APP 实现

1. APP 应用模块主要用于对系统功能的可视化应用展示和性能分析。通过对图像识别模块、目标检测模块训练生成的模型进行对比分析，分别选择较优模型投入 APP 应用模块进行实际应用，更直观地展示果蔬识别情况，对模型的优化和模型调用算法的改善提供分析依据。APP 应用模块在对两种识别模式进行常规实现的基础上，对实际问题进行思考，延伸拓展实现语义分割模块和视频实时检测功能模块。系统实现架构如图 10 示。

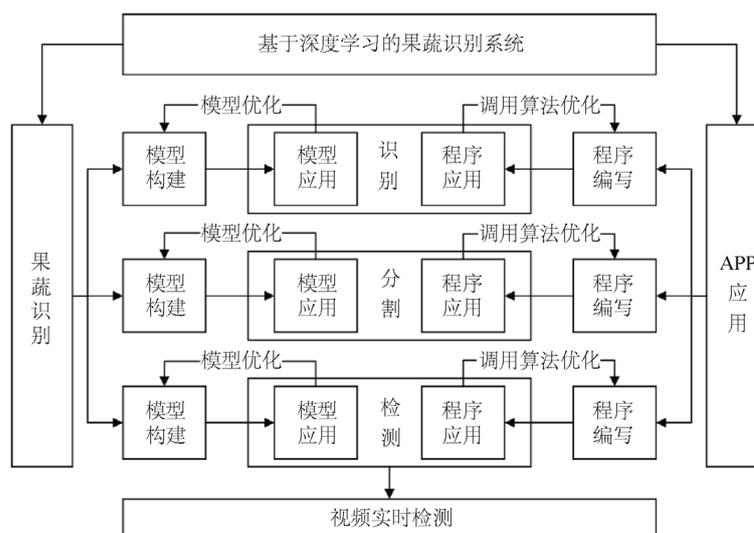


Figure 10. System implementation architecture diagram

图 10. 系统实现架构图

为了提高 APP 应用的灵活性, APP 可使用相册图片和摄像头拍摄两种方式上传图片, 并允许许多张图片上传识别。APP 对输入的图像数据进行后端存储, 便于数据集扩充, 为后期模型的调整与优化做准备。当系统完成识别时, 对结果进行可视化输出展示, 提供识别准确率、识别时间等参数供模型优化和模型调用算法优化进行参考。APP 主页、图像识别模块、目标检测模块效果图如图 11、图 12、图 13 所示。



Figure 11. App home effects  
图 11. APP 主页效果图



Figure 12. Image recognition effect  
图 12. 图像识别效果图



Figure 13. Target detection effect chart  
图 13. 目标检测效果图

## 6. 实验结果分析

本文实验环境为 Windows 操作系统，GTX950M 显卡，使用 Python-3.5 和 Tensorflow-gpu-1.9.0 进行神经网络模型构建，使用 Django-1.11 实现 Web APP 进行系统可视化应用展示。通过对不同模型及其相关因素进行分析，结论如下：

(1) 输入图像大小对性能的影响。输入图像的大小直接影响 APP 功能完成所需时间，图像越大，识别所需处理的数据就越多。APP 在后台将输入的图像数据 resize 为  $400 \times 400$  大小，一方面防止图像过大时影响性能，另一方面使用与模型数据集相似的图像数据易于实现更好的识别准确率。

(2) 图像识别应用模块性能分析与优化。通过对模型调用算法的调整，测试发现进行图像识别时模型的调用花费了较多的时间，因此考虑能否对模型调用时间进行分离，使模型在 APP 上有更好的应用性能。重复测试 10 次模型调用时间分离前后识别 1 张、2 张图像所需时间情况如图 14 示。

将每种测试结果的后五次数值取均值，结果如表 2、表 3 所示，对比可见分离模型调用对缩短识别时间能起到较好的作用，单张识别速度可提升约 119%，2 张识别速度可提升约 85%，随着单次识别的图像数目增加，识别所需总时间增加，模型调用分离对速度的提升将逐渐减小。

(3) 目标检测应用模块性能分析与优化。

faster\_rcnn\_inception\_v2 结构模型与 ssd\_mobilenet\_v1 结构模型目标检测耗时对比。SSD 模型在综合性能上常优于 Faster RCNN 模型，在本系统对比中由于部分原因未能够使用相当模型进行对比，仅以 faster\_rcnn\_inception\_v2 结构模型和 ssd\_mobilenet\_v1 结构模型进行对比供参考作用。

如图 15 所示，10 次测试中使用 ssd\_mobilenet\_v1 结构模型进行目标检测耗时明显少于使用 faster\_rcnn\_inception\_v2 结构模型，各使用测试的后五次结果取均值，faster\_rcnn\_inception\_v2 结构模型平均耗时 15.31 秒，ssd\_mobilenet\_v1 结构模型平均耗时 8.22 秒，后者耗时仅为前者 53.7%，且亦有较高的准确率，有较好的应用性能。

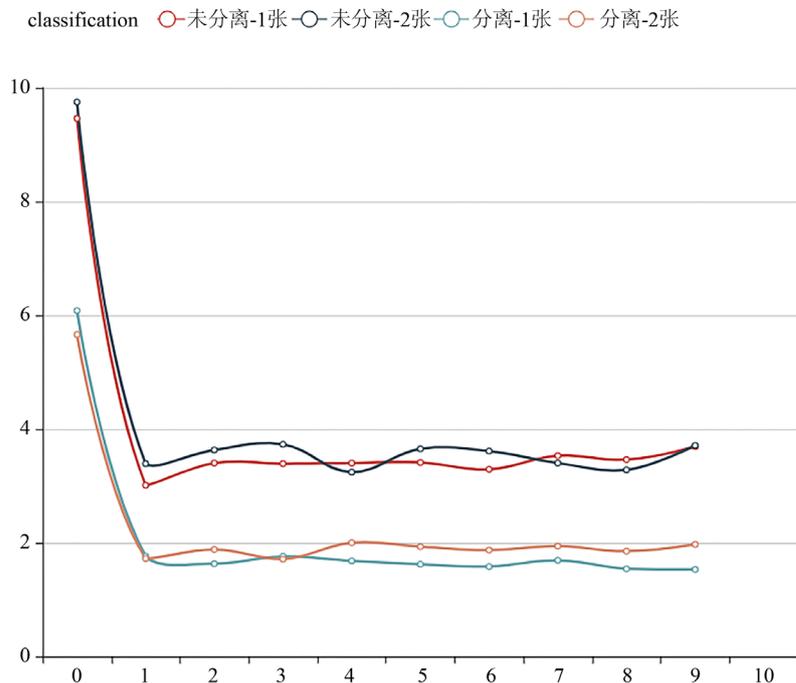


Figure 14. Time comparison diagram before and after image recognition calls optimization

图 14. 图像识别调用优化前后时间对比图

Table 2. 1 comparison of recognition time before and after image adjustment

表 2. 1 张图像调整前后识别时间对比

| 方式      | 未分离-1 张 | 分离-1 张 | 速度提升率 |
|---------|---------|--------|-------|
| 平均时间(秒) | 3.48    | 1.59   | 119%  |

Table 3. Comparison of recognition time before and after image adjustment

表 3. 2 张图像调整前后识别时间对比

| 方式    | 未分离-2 张 | 分离-2 张 | 速度提升率 |
|-------|---------|--------|-------|
| 速度提升率 | 3.53    | 1.91   | 85%   |

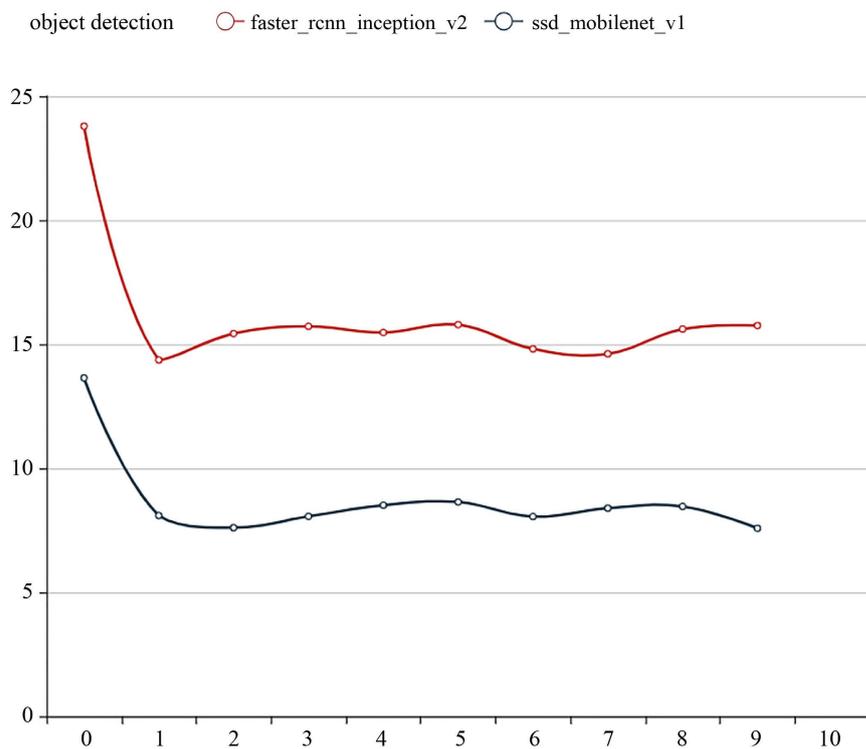
模型调用算法优化。为使对比结果更为直观，该部分选用了目标检测耗时较长的 Faster RCNN 模型进行测试。将所有图像数据一起送入目标检测算法，从原先的每张图像数据进行一次模型调用转化为所有图像数据进行一次模型调用，极大缩短了目标检测功能实现时间，提升了功能实现效率。重复测试 10 次模型调用算法优化前后识别 1 张、2 张图像所需时间情况如图 16。

将每种测试结果的后五次数值取均值，结果如表 4 所示，对比可见分离模型调用对缩短识别时间能起到较好的作用，2 张图像的识别速度可提升约 83%，分离模型调用后每张图像的检测时间约 0.93 秒，随着单次识别图像数目增加，识别速度提升率将更高。

(4) 深入优化模型调用算法思考：

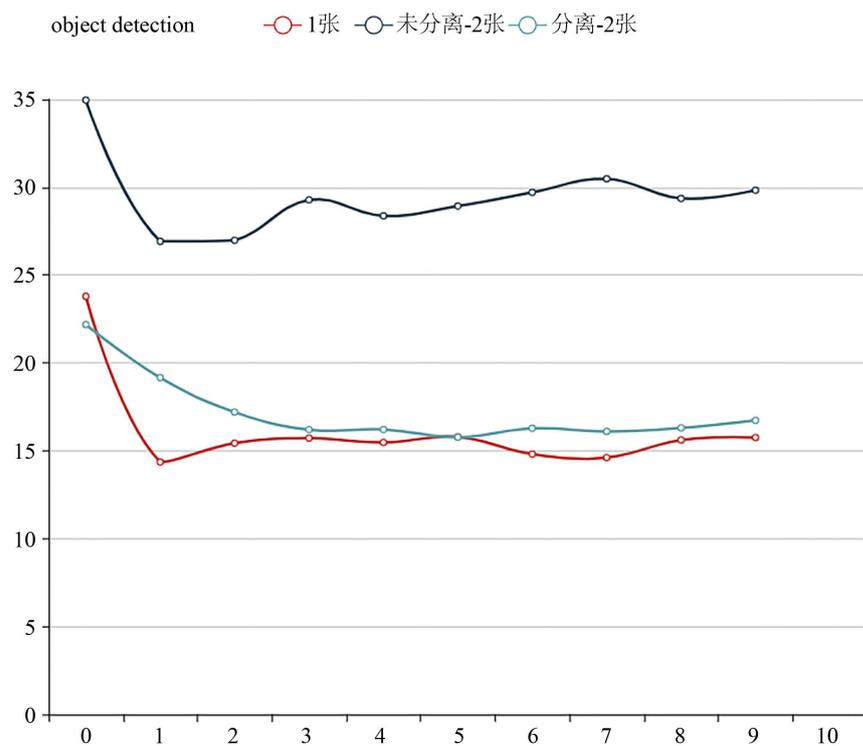
模型调用需要消耗大量时间，在输入图像较少时占用时间更加明显，如果可以使模型在系统环境中长存，使每次进行识别或检测操作时可直接调用，将有效提升应用效率。

模型最初几次调用时间明显较长，可尝试在应用程序运行前对模型进行多次预使用，从而达到模型预热的效果。



**Figure 15.** To detect different model time comparison

**图 15.** 标检测不同模型时间对比图



**Figure 16.** Target detection call before and after the time comparison graph

**图 16.** 目标检测调用优化前后时间对比图

**Table 4.** Comparison of detection times before and after adjustment  
**表 4.** 调整前后检测时间对比

| 方式      | 1 张   | 未分离-2 张 | 分离-2 张 | 2 张速度提升率 |
|---------|-------|---------|--------|----------|
| 平均时间(秒) | 15.31 | 29.66   | 16.24  | 83%      |

## 7. 结论

本文基于深度学习的果蔬识别系统研究了模型的构建、分析与应用展示。模型分析模块使用了自行搭建的分析平台,结合 Tensorboard 对模型进行精准分析。在果蔬识别模块分别采用了常规图像识别、目标检测两个功能模块,实现单果蔬分类、多果蔬定位分类,且能达到较好的识别准确率,并将功能延伸到语义分割和视频实时检测上,满足大部分果蔬识别需求。系统采用 Web APP 对识别结果进行直观展示,便于后期分析、改善及应用。虽然在果蔬识别以及其他应用场景中,识别已经能达到较高的准确度,但在缩小训练数据集、提升识别速度上还有待继续提升,以便更好满足人们对美好生活的向往[9]。

## 致 谢

本文的工作得到 2018 国家级创新创业训练项目(编号: 201810338028)和教育部协同创新项目(编号: 201801200045)的资助,特此感谢。

## 参考文献

- [1] 王先庆,雷韶辉. 新零售环境下人工智能对消费及购物体验的影响研究——基于商业零售变革和人货场体系重构视角[J]. 商业经济研究, 2018(17): 5-8.
- [2] 王前程. 基于深度学习的水果图像识别算法研究[D]: [硕士学位论文]. 保定: 河北大学, 2016.
- [3] 驱动中国. 计算机视觉+机器学习进入农业, 未来要用手机种田吗[EB/OL]. <https://mobile.qudong.com/article/435835.shtml>, 2017-09-18.
- [4] 芝麻仓库. 当农业遇上人工智能[EB/OL]. <https://baijiahao.baidu.com/s?id=1596167014452729008>, 2018-03-28.
- [5] 博客园. 深度学习调参技巧[EB/OL]. <https://www.cnblogs.com/ranjiewen/p/7682309.html>.
- [6] DexterLei. TensorFlow 学习笔记: 使用 Inception v3 进行图像分类[EB/OL]. <https://www.jianshu.com/p/cc830a6ed54b>, 2017-10-03.
- [7] Ren, S., He, K., Girshick, R., et al. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans Pattern Anal Mach Intell*, **39**, 1137-1149.
- [8] Liu, W., Anguelov, D., Erhan, D., et al. (2016) SSD: Single Shot MultiBox Detector. *European Conference on Computer Vision*, Springer, Cham, 21-37.
- [9] 前瞻经济学人. 2018 年计算机视觉技术三大发展趋势分析[EB/OL]. <https://www.qianzhan.com/analyst/detail/220/181012-a8dab736.html>, 2018-10-15.