Computer Science and Application 计算机科学与应用, 2021, 11(4), 962-974
Published Online April 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.114099

Hans汉斯

基于轻量级神经网络的食材识别方法研究

黄颖康，曾　碧

广东工业大学，广东 广州
Email: 443839706@qq.com, zb9215@gdut.edu.cn

收稿日期：2021年3月21日；录用日期：2021年4月15日；发布日期：2021年4月22日

摘　要

随着计算机视觉技术的快速发展，基于深度学习的目标检测技术已广泛应用于诸多领域。由于目前目标检测算法模型复杂，计算量大，无法应用于嵌入式设备中，为了满足在嵌入式设备中使用食材识别功能的需求，提出了对目标识别模型YOLOv3的改进方法，将轻量化神经网络MobileNet应用于YOLOv3中，把YOLOv3的主干网络darknet53替换为MobileNet；然后采用Cluster-NMS算法，配合中心距离法和加权平均法提升网络的准确度。通过收集得来的食材数据集和VOC 2007数据集对网络进行对比实验。实验表明，改进后的网络模型，既能满足其迁移到嵌入式设备的轻量级需求，而且无论在识别速度和精度上都有提升，满足在嵌入式设备实现食材识别的功能。

关键词

深度学习，轻量化，非极大抑制，食材识别，目标检测

Research on Food Ingredients Identification Method Based on Lightweight Neural Network

Yingkang Huang, Bi Zeng

Guangdong University of Technology, Guangzhou Guangdong
Email: 443839706@qq.com, zb9215@gdut.edu.cn

Received: Mar. 21st, 2021; accepted: Apr. 15th, 2021; published: Apr. 22nd, 2021

Abstract

With the rapid development of computer vision technology, object detection technology based on

文章引用: 黄颖康, 曾碧. 基于轻量级神经网络的食材识别方法研究[J]. 计算机科学与应用, 2021, 11(4): 962-974.
DOI: 10.12677/csa.2021.114099

deep learning has been widely used in many fields. Due to the complexity of the current target detection algorithm model, the amount of calculation is large, model can not be applied to embedded devices. In order to meet the needs of using food ingredients identification function in embedded devices, an improved method of object recognition model YOLOv3 is proposed. The lightweight neural network MobileNet is applied in YOLOv3, and the main network darknet53 of YOLOv3 is replaced by MobileNet. And then Cluster Non-Maximum Suppression (Cluster-NMS), Distance NMS and Weighted NMS are used to improve the accuracy of the neural network. Through comparative experiments by testing neural network with food ingredients data set and VOC 2007 data set, the improved network model meets the needs of network migration to embedded devices, improves the accuracy of the neural network and the ability of food ingredients recognition, and realizes the function of food identification in embedded devices.

Keywords

Deep Learning, Lightweight, NMS, Food Ingredients Identification, Object Detection

1. 引言

随着社会的发展使得生活水平不断提高，人们对衣食住行的要求逐渐增高。其中吃是体现生活质量的一个重要方面。随着信息领域和物流领域的迅速发展，人们能够通过网络等渠道采购到各种丰富的食材。以往人们对食材种类的认知以及对应烹饪菜谱是依靠经验得来，如何使用嵌入式移动设备(比如：手机)自动识别采购得来的食材，配合移动端的菜谱应用软件，就能生成各种美味食品的烹饪方法，对满足人们对美食的追求具有现实意义。但这也对食材识别模型和识别速度提出了更高的要求。

近年来，随着人工智能技术的发展，深度学习得到了很大的关注。深度学习与传统手工设计目标特征的方法不同，深度学习以端到端的方式训练深度神经网络，实现自动提取目标的深层特征，避免人为设计的干扰，同时，深层特征与传统特征相比，深层特征可以多层次表示目标，从浅层的局部特征到深层的全局特征，具有更强的鲁棒性和表达能力。随着深度神经网络发展，出现了许多经典的用于目标识别的神经网络模型，包括 R-CNN [1]、Fast RCNN [2]、SSD [3]、YOLO 系列[4] [5] [6]等，但是为了让模型达到更高的精度，大多数神经网络模型制造得更深，复杂度更高高，难以应用在真实场景中，所以必须要对模型进行轻量化处理，以达到移植到各种嵌入式设备的要求[7] [8] [9] [10]。

为了提高嵌入式设备对食材图片识别的效率和能力，同时需要满足存储空间和功耗的限制，设计适用于嵌入式设备的轻量化深度神经网络架构是解决该问题的关键。在最近的研究中，人们对建立轻量级和高精度的神经网络越来越感兴趣[11] [12] [13] [14]，因此，轻量级神经网络架构的设计得到了学术界和工业界的广泛关注，也提出了一些典型的方法[15] [16]，主要包括三个方向，分别是：(1) 人工设计轻量化神经网络模型；(2) 基于神经网络架构搜索(Neural Architecture Search，NAS)的自动化神经网络架构设计；(3) 神经网络模型的压缩。

对比目标检测的其他神经网络模型，YOLOv3 在速度与效果方面表现不错，因此针对食材识别任务，本文提出一种基于聚类加权中心非极大值抑制的轻量级 YOLOv3 方法(Cluster-Weighted-Distance NMSlightweight YOLOv3, CWDNMS-lightweightYOLOv3)。采用 YOLOv3 作为主要的网络结构，使用轻

量化神经网络 MobileNet [17]对 YOLOv3 进行轻量化处理，能够有效减少整个网络的参数量，提升整个模型的运行效率，使模型能够迁移到移动端或其他嵌入式设备中。除此之外为了进一步提升模型的精度和效率，对传统有缺陷的非极大值抑制(Non-Maximum Suppression, NMS)算法作出改进，使用交并比(Intersection-over-Union, IoU)矩阵运行方式，并且加入了中心距离惩罚项改变了仅使用 IoU 作为判断检测框是否抑制的依据，采用加权平均法得到了更准确的最高得分检测框，进一步增强食材识别的效果。

2. 轻量级神经网络的构建

2.1. YOLOv3

YOLOv3 网络结构图如图 1 所示，把图片输进 YOLOv3，网络会对图片分别进行 32 倍降采样、16 倍降采样、8 倍降采样，输出三种尺寸的特征图。但是在进行 16 倍降采样检测时会先对 32 倍降采样的特征图进行上采样后再与 16 倍降采样的特征图结合输出最终的 16 倍降采样特征图，8 倍降采样的操作与 16 倍降采样的操作相同，这样既可以提高非线性处理能力，增加泛化性能以提高网络精度，又能减少参数提高实时性。例如输入 416 * 416 的图片，会输出 13 * 13、26 * 26、52 * 52 的特征图。YOLOv3 的降采样操作不使用池化层，而是将卷积步骤中的步长设置为 2 以达到降采样的目的，这样能够降低池化操作所带来的梯度负面效果。

图片经过 32 倍降采样后，特征图的感受野最大，适合检测较大的目标；图片经过 16 倍降采样后，特征图的感受野适中，适合检测一般大小的目标；图片经过 8 倍降采样后，特征图的感受野最小，适合检测较小的目标。YOLOv3 的这种多尺度预测的特点，能够适应各种大小的识别目标。

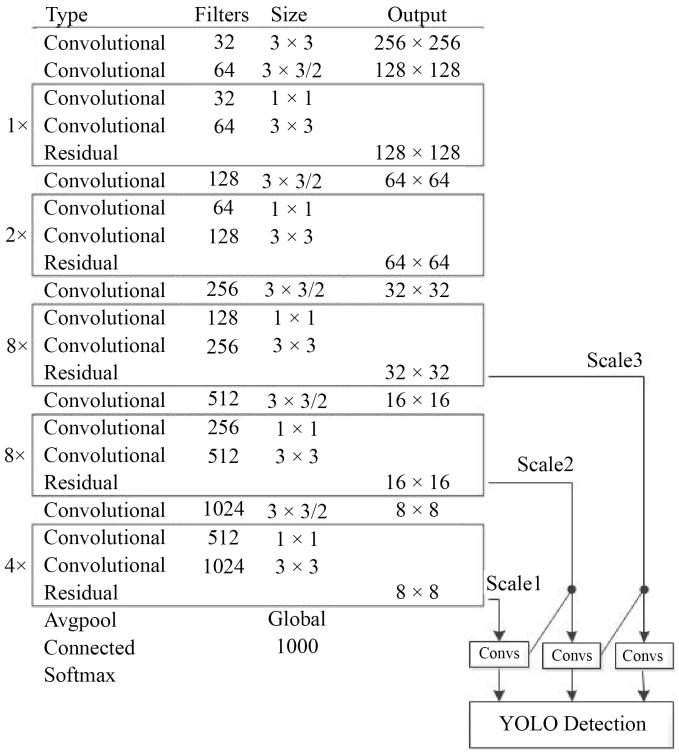

Figure 1. Yolov3 network structure diagram
图 1. YOLOv3 网络结构图

YOLOv3 最后输出的特征图里，每一个网络单元预测 3 个不同尺寸的检测框，检测框的尺寸通过对

输入图片里的目标进行聚类得到，因为 YOLOv3 输出 3 种征图，每种特征图有三种尺寸的检测框，所以 YOLOv3 一共有 9 种尺寸的检测框检测不同大小的检测目标，能够提升网络的泛化能力。最终特征图里，每一个网络单元含有检测框的坐标信息，检测框的高度和宽度以及目标预测的置信度。

YOLOv3 的主干网络是 darknet53，上图左边部分。darknet53 网络有 52 层卷积操作，采用 3 * 3 大小的卷积核，分别在第 2、5、10、27、44 层结构处进行降采样的操作。

2.2. MobileNet

MobileNet 是一种轻量及高效的神经网络，相比于 darknet53，它的网络模型较小，准确度高以及网络结构图有一定的相似之处。

MobileNet 使用深度可分离卷积代替传统卷积，深度可分离卷积区别于传统卷积，在于卷积的方式和构成不一样。深度可分离卷积操作时一个通道的卷积核对一个通道的特征图进行卷积后，直接作为该通道的输出特征图，而传统卷积操作时是对每一通道特征图卷积的结果进行叠加后作为该通道的输出特征图。

YOLOv3 的基本单位由 3 * 3 普通卷积层，批量归一化(Batch Normalization, BN)层和 Relu 激活函数层构成，如图 2 所示。其中，模型通过 3 * 3 普通卷积层对输入图片进行特征提取；BN 层通过对数据进行归一化的操作，不仅提升模型训练速度，加快收敛过程，还能防止模型训练时容易出现的过拟合现象。Relu 激活函数是非线性函数，引入模型能够增加模型的非线性因素，增加神经网络各层之间的非线性关系，增强模型的表达能力；Relu 激活函数相对于其他激活函数的优势是：计算量少；不容易出现梯度消失的问题；缓解过拟合问题的发生。

MobileNet 的基本单位是深度可分离卷积[18]，如图 2 所示，深度可分离卷积由 3 * 3 和 1 * 1 卷积层，BN 层和 Relu 激活函数层构成，通过 3 * 3 深度卷积提取特征图的特征，通过 1 * 1 卷积操作控制输出特征图的通道数。

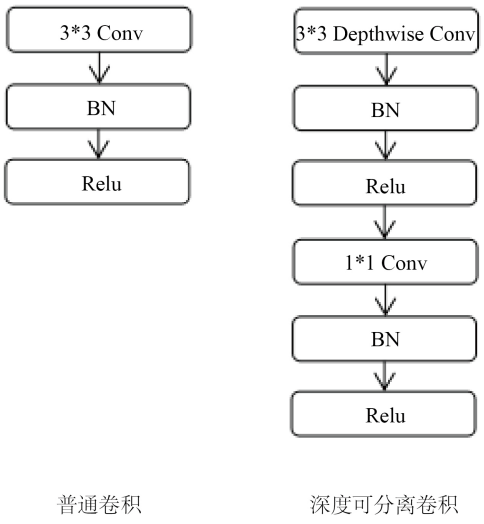

Figure 2. Comparison of depth separable convolution and ordinary convolution
图 2 深度可分离卷积与普通卷积比较图

假设输入和输出的特征图大小为 D_F，其中输入特征图通道数为 E，输出特征图通道数为 N，卷积核大小为 D_K，则深度可分离卷积和普通卷积的计算量之比公式为：

$$\frac{D_K \times D_K \times E \times D_F \times D_F + E \times N \times D_F \times D_F}{D_K \times D_K \times E \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{1}$$

通道数 N 一般较大，卷积核大小一般为 3*3，所以深度可分离卷积的计算量大约只有普通卷积的计算量的九分之一[19]。

MobileNet 网络结构图如图 3 所示，整个网络有 27 层卷积操作，并且在 1，4，8，12，24 层进行降采样操作。整个网络与 darknet53 网络相似，但是 MobileNet 除了第一层卷积，其余卷积使用了深度可分离卷积操作，减少了网络计算量，使 MobileNet 移植到 YOLOv3 提供了可能性。

层类型/步长	卷积核大小	输入尺寸
普通卷积/2	3×3×3×32	416×416×3
深度卷积/1	3×3×32	208×208×32
点卷积/1	1×1×32×64	208×208×32
深度卷积/2	3×3×64	208×208×64
点卷积/1	1×1×64×128	104×104×64
深度卷积/1	3×3×128	104×104×128
点卷积/1	1×1×128×128	104×104×128
深度卷积/2	3×3×128	104×104×128
点卷积/1	1×1×128×256	52×52×128
深度卷积/1	3×3×256	52×52×256
点卷积/1	1×1×256×256	52×52×256
深度卷积/2	3×3×256	52×52×256
点卷积/1	1×1×256×512	26×26×256
5×(深度卷积/1+ 点卷积/1)	3×3×512 1×1×512×512	26×26×512 26×26×512
深度卷积/2	3×3×512	26×26×512
点卷积/1	1×1×512×1024	13×13×512
深度卷积/1	3×3×1024	13×13×1024
点卷积/1	1×1×1024×1024	13×13×1024
均值池化/1	7×7	13×13×1024
全连接/1	1024×1000	1×1×1024
softmax/1	无尺寸	1×1×1000

Figure 3. MobileNet network structure diagram
图 3. MobileNet 网络结构图

把 MobileNet 代替 darknet53 移植到 YOLOv3 网络中，得到 YOLOv3-MobileNet 网络结构，如图 4 所示。

图 4 中 CBR 为普通卷积，如图 2 左侧部分所示，DBR 为深度可分离卷积，如图 2 右侧部分所示，Conv 为普通卷积操作，不包括 BN 算法和激活函数。

从图 4 中可得在 MobileNet 部分第 6 层后输出 8 倍降采样的特征图，第 12 层后输出 16 倍降采样的特征图，最后输出 32 倍降采样的特征图。各个倍数的特征图经过卷积或上采样等操作后，输出最终的特征图。

YOLOv3-MobileNet 网络结构通过将 MobileNet 代替 darknet53 移植到 YOLOv3 的主干网络中减少整个网络的冗余程度，参数量从 61 M 降到了 24 M。YOLOv3-MobileNet 整体结构与 YOLOv3 相似，并且在降低网络复杂度的同时，保留了许多 YOLOv3 的特性，例如多尺度预测等。

3. NMS 算法设计

NMS，是目标检测的重要部分，大部分神经网络，包括 YOLOv3，在最后输出特征图中保存着大量的信息，其中含有大量的检测框信息，每个检测框有各自的置信度，用来表达检测框属于某个种类的可

信程度。大量的检测框互相重叠，需要 NMS 算法抑制大量的检测框，确保检测框的正确预测某个种类。NMS 的作用就是选取置信度高的检测框，抑制重叠度高的检测框。

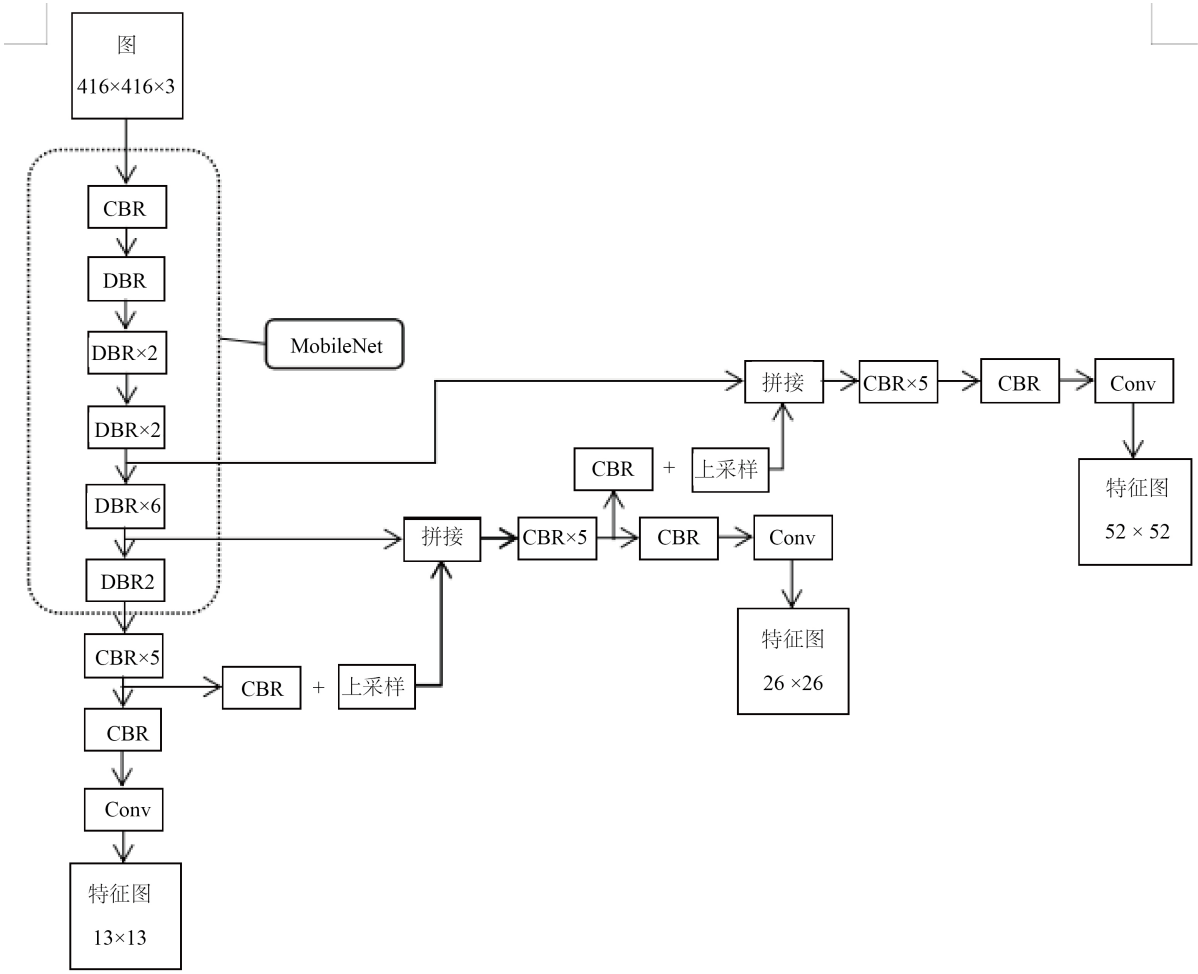

Figure 4. YOLOv3-MobileNet network structure diagram
图 4. YOLOv3-MobileNet 网络结构图

其算法流程如下:

1) 对检测框按置信度从高到低进行排序，得到集合 M;

2) 选取集合 M 中置信度最高的检测框 b，遍历其余检测框；使与 b 的 IoU 大于阈值的其余检测框置信度设为 0，因为它们判断为属于同一种类;

3) 从集合 M 中剔除 b 重复步骤 2 和 3 直到遍历完集合 M。

这种算法的缺点是：IoU 阈值设置高了，会导致抑制错误检测框的效果较差，设置低了，容易把重叠的不同种类的检测框给抑制了。

3.1. Cluster-NMS

NMS 算法在计算 IoU 方面的公式为：

$$(n-1)+(n-2)+\cdots+1=\frac{1}{2}n^2-\frac{2}{n} \tag{2}$$

想要加速 NMS，就要将 IoU 计算并行化。

IoU 矩阵为：

$$\boldsymbol{X}=\mathrm{IoU}(B,B)=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix},\quad x_{ij}=\mathrm{IoU}(B_i,B_j) \tag{3}$$

式(3)中，$\boldsymbol{X}$ 为 IoU 矩阵，B 为检测框集合，其中 $B=\{B_i\}_{i=1\text{ to }n}$，集合 B 按照置信度降序排列，也就是 B_1 是最高得分框，B_n 为最低得分框。

因为 $\mathrm{IoU}(B_i,B_j)=\mathrm{IoU}(B_j,B_i)$,所以 $\boldsymbol{X}$ 为对称矩阵，且 B_{ii} 没有意义，所以 $\boldsymbol{X}$ 可以简化为：

$$\boldsymbol{X}=\begin{bmatrix} 0 & x_{12} & x_{13} & \cdots & x_{1n} \\ 0 & 0 & x_{23} & \cdots & x_{2n} \\ 0 & 0 & 0 & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{4}$$

对 $\boldsymbol{X}$ 执行按列取最大值操作，得到一维张量 $\boldsymbol{b}=[b_1,b_2,\cdots,b_n]$，对 $\boldsymbol{b}$ 中元素进行二值化，大于 IoU 阈值的元素取 0，小于 IoU 阈值的元素取 1。最后张量 $\boldsymbol{b}$ 中 b_i 为 0 代表第 i 个检测框抑制，为 1 代表第 i 个检测框保留。

但是这种方法会导致检测框被过度抑制[20]。例如，矩阵 $\boldsymbol{X}$ 第二行 B_2 是与比它低分的检测框之间的 IoU 数值，若 b_2 为 0，代表第二高分的检测框被抑制了，也就是说矩阵 $\boldsymbol{X}$ 第二行应该为 0，但矩阵 $\boldsymbol{X}$ 第二行依然有数值并且该数值有可能是某列的最大值，有可能在二值化时被设为 0，造成该列的检测框被抑制。

为了解决这种缺点采取 Cluster-NMS 算法：

1) 使矩阵 $\boldsymbol{C}_1=\boldsymbol{E}\times\boldsymbol{X}$，第一次迭代用单位矩阵 $\boldsymbol{E}$ 左乘矩阵 $\boldsymbol{X}$ 得到矩阵 $\boldsymbol{C}_1$，按照前面方法得到 $\boldsymbol{C}_1$ 的一维张量 $\boldsymbol{b}^1$；

2) 把得到的一维张量 $\boldsymbol{b}^1$ 展开成对角矩阵 $\boldsymbol{A}_1$，也就是使 $\boldsymbol{A}_1$ 的对角线元素与一维张量 $\boldsymbol{b}^1$ 的元素相同，其余元素为 0，例如 $\boldsymbol{b}^1=[1\quad 0\quad 1\quad \cdots\quad 1]$，得到以下矩阵：

$$\boldsymbol{A}_1=\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{5}$$

3) 使 $\boldsymbol{C}_2=\boldsymbol{A}_1\times\boldsymbol{X}$，得到 $\boldsymbol{C}_2$ 的一维张量 $\boldsymbol{b}^2$ 和 $\boldsymbol{A}_2$，重复步骤(2)(3)直到 $\boldsymbol{b}^{n-1}=\boldsymbol{b}^n$。

如果对角矩阵 $\boldsymbol{A}$ 中某行为 0，则与矩阵 $\boldsymbol{X}$ 相乘后该行元素全为 0，代表该行被抑制了。Cluster-NMS 算法最后的结果和传统 NMS 算法的结果一样，但是速度比传统的 NMS 快[21]。

3.2. 中心距离惩罚法(DIoU-NMS)

中心距离惩罚法的核心思想是相邻检测框的中心约接近得分最大的检测框，越有可能是冗余的检测框[22]。检测框示意图如图 5 所示。

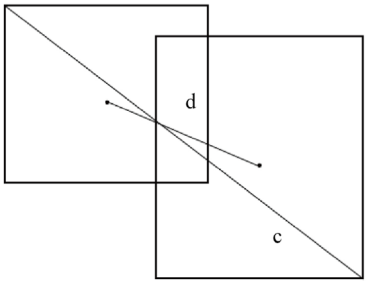

Figure 5. Diagram of two adjacent detection frames
图 5. 两个相邻检测框示意图

$$\mathrm{DIoU} = \mathrm{IoU} - \left(\frac{d^2}{c^2}\right)^{\beta} \tag{6}$$

式(6)中 d 为两检测框中心的距离，c 为两检测框最远的距离，参数 β 控制惩罚项对 IoU 的影响程度。

用 DIoU 代替 IoU 作为 NMS 的评判准则，当 $\beta \to \infty$，惩罚项趋向 0，此时 DIoU=IoU，DIoU-NMS 与传统 NMS 效果相当；当 $\beta \to 0$，大部分检测框都没有被抑制。

使用 DIoU-NMS 能有在一定程度缓解传统 NMS 算法中抑制遮蔽物体检测框的问题，并且 DIoU 最差的效果，即惩罚项趋向于 0，效果也不会比传统 NMS 效果差，对 NMS 只有改进的效果。在阈值不变的情况下，会有更多的检测框保留下来，提升检测的召回率。

3.3. 加权平均法(Weighted NMS)

传统 NMS 得到的置信度最高的检测框定位不一定是最精确的，也就是说冗余的检测框有可能比置信度最高的检测框定位更好，所以采取加权平均法是对坐标进行加权平均[23]，公式：

$$M = \frac{\sum_i w_i B_i}{\sum_i w_i}, \quad B_i \in \{B \mid \mathrm{IoU}(M, B) \geq \text{阈值}\} \cup \{M\} \tag{7}$$

式(7)中权重 $w_i = s_i \mathrm{IoU}(M, B_i)$。

通过 Weighted NMS，对检测框进行加权平均，能够得到定位效果较准确的最高置信度检测框，能够较为稳定地提升检测的召回率和精度。

在 Cluster-NMS 的基础上使用中心距离惩罚法和加权平均法对 NMS 进行进一步的改进，结合以上三种 NMS 改进算法得到聚类加权中心非极大值抑制算法(Cluster-Weighted-Distance NMS, CWDNMS)，该算法可以使神经网络性能在效率和精度方面有所提升。

4. 实验结论与分析

4.1. 实验步骤与方法

实验的硬件平台是：

操作系统：Windows 10；

CPU：英特尔 i7-6700 3.4 GHz；

内存：16 GB；

显卡：英伟达 GTX 1070。

软件平台：Pycharm、Anaconda3。

实验步骤：收集食材数据集，对食材数据集进行筛选，对筛选后的食材数据集进行预处理，预处理包括划分食材数据集为训练集，验证集，测试集，对数据集进行标定。在软件平台上构建YOLOv3-MobileNet 网络结构，编写 Cluster-Weighted-Distance NMS 算法程序。把训练集送进YOLOv3-MobileNet 网络结构进行训练，得到相关的参数，把测试集送进训练后网络中，得到网络的精度和速度，评估网络的泛化能力。使用 VOC 2007 数据集代替食材数据集进行上述实验步骤，得出结果与食材数据集进行结果对比；把两种数据集分别送进 YOLOv3 和是否使用 Cluster-Weighted-Distance NMS 算法的网络中，进行实验结果的对比和分析。主要分析对 YOLOv3 进行轻量化操作后，模型的检测速度是否提升了，改进后的 NMS 算法，模型的检测精度是否提升了。

VOC 2007 数据集是为图像识别和分类提供了一整套标准化的优秀的数据集。VOC 2007 数据集共包含：训练集 5011 张图片，测试集 4952 张图片，共 9963 张图片，共包含 20 个种类，其中包括人类、单车、汽车等常见的种类。

自制食材数据集包括 2300 张图片，共包含 10 个种类，每个种类 200 到 300 张图片，种类分别是茄子、鸡蛋、肉、青菜、西红柿、西兰花、玉米、洋葱、萝卜、胡萝卜。标注工具为 LabelImg，方式是人工标注，图片来源于实景拍摄。

4.2. 实验结果比较

表 1 是以 VOC2007 作为数据集对各种网络评估的结果。可以从表中得知 YOLOv3-MobileNet 模型的速度为 36 ms，YOLOv3 模型的速度为 61 ms，YOLOv3-MobileNet 模型比 YOLOv3 模型快了近一倍。在精度方面，使用 IoU 为 0.5 作为判断基准，使用 NMS 的 YOLOv3 模型的准确度为 65.84%，使用 NMS 的 YOLOv3-MobileNet 的模型准确度为 62.66%，YOLOv3 使用了 MobileNet 代替 darknet53 作为主干网络，在精度方面有略微的下降，但是在速度方面快了近一倍。模型采用 CWD-NMS 算法后，检测准确度提升 3%左右。

Table 1. Comparison of VOC 2007 operation results
表 1. VOC 2007 运行结果比较

模型	参数量/M	速度/ms	mAP@IoU = 0.5
YOLOv3 + NMS	61	61	65.84%
YOLOv3 + CWDNMS	\	\	68.12%
YOLOv3-MobileNet + NMS	24	36	62.66%
YOLOv3-MobileNet + CWDNMS	\	\	65.96%

表 2 是自制的食材数据集对各种网络评估的结果。可以从表中得知 YOLOv3-MobileNet 模型的速度为 28 ms，YOLOv3 模型的速度为 62 ms，YOLOv3-MobileNet 模型比 YOLOv3 模型快了一倍多。在精度方面，使用 IoU 为 0.5 作为判断基准，使用 NMS 的 YOLOv3 模型的准确度为 69.29%，使用 NMS 的 YOLOv3-MobileNet 的模型准确度为 68.44%，YOLOv3 使用了 MobileNet 代替 darknet53 作为主干网络，在精度方面有略微的下降，但是在速度方面快了一倍多。模型采用 CWD-NMS 算法后，检测准确度提升 7%左右。

Table 2. Comparison of VOC 2007 operation results
表 2. VOC 2007 运行结果比较

模型	参数量/M	速度/ms	mAP@IoU = 0.5
YOLOv3 + NMS	61	61	69.29%
YOLOv3 + CWDNMS	\	\	76.05%
YOLOv3-MobileNet + NMS	24	28	68.44%
YOLOv3-MobileNet + CWDNMS	\	\	75.21%

结合表和表中数据分析可得出结论是，YOLOv3-MobileNet 对比与 YOLOv3，使用 MobileNet 代替 YOLOv3 原有的 darknet 作为主干网络，参数减少了一大半，速度从 61 ms 左右上升到 30 ms 左右，快了一倍，精度只是略微下降；模型采用 CWD-NMS 算法后，精度提升了 7%。使用 MobileNet 作为主干网络的 YOLOv3 模型达到轻量化的效果，精度只略微下降，采用 CWD-NMS 算法后，模型有不错的提升。

4.3. 实验结果分析

从表 2 可知，使用自制食材数据集对 YOLOv3-MobileNet 模型进行训练，并使用 CWD-NMS 算法提升精度，模型的精度达到 75.21%，使用 VOC 2007 数据集对 YOLOv3-MobileNet 模型进行训练，并使用 CWD-NMS 算法提升精度，得到的模型精度达到 65.96%。自制食材数据集在该模型中表现比 VOC 2007 数据集好。

由图 6 可得在 VOC 2007 的全部种类中，火车种类准确度达到 87%，盆栽种类准确度只达到 30%；超一半种类的准确度达到 60%以上，有五个种类准确度达到 80%及以上。

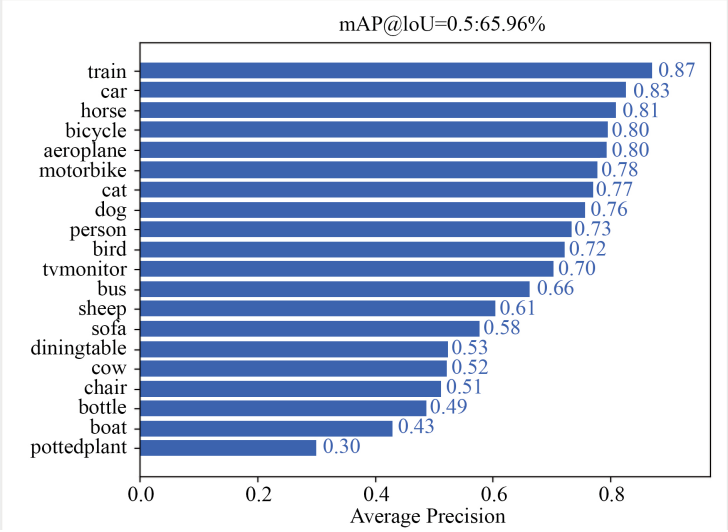

Figure 6. Various class accuracy maps of VOC 2007 dataset
图 6. VOC 2007 数据集的各种类准确度图

由图 7 可得食材数据集中洋葱和青菜的检测准确度很好，鸡蛋和玉米的检测准确度很低，尤其是玉米的准确度只有 36%。

最终检测效果由图 8 所示，图片中含有青菜、鸡蛋、胡萝卜食材，全部品种都检测出来，青菜的检测置信度有 100%，鸡蛋的检测置信度只有 41%。

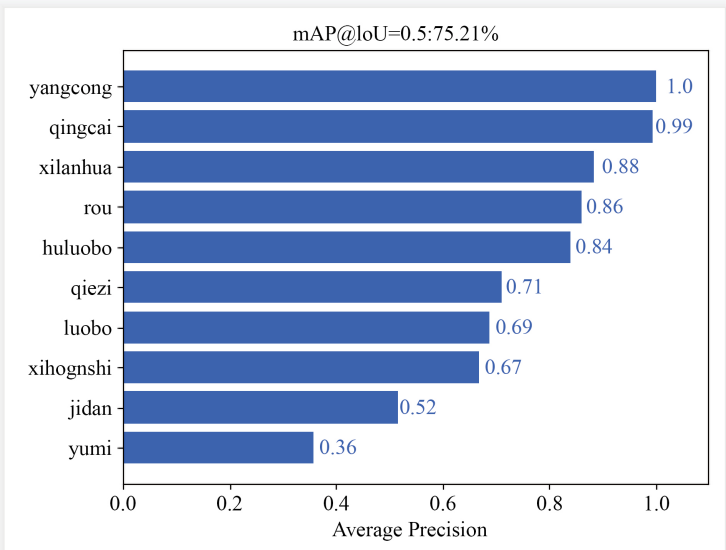

Figure 7. Various class accuracy maps of food material dataset
图 7. 食材数据集的各种类准确度图

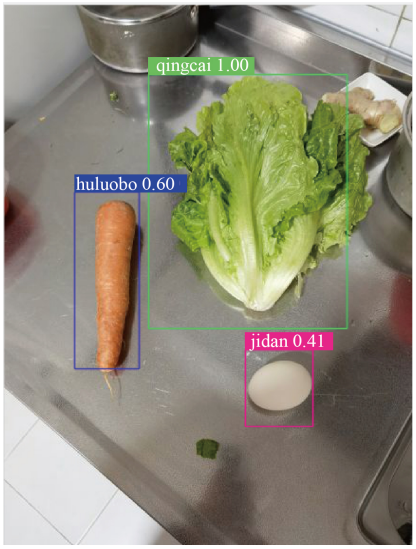

Figure 8. Food detection picture
图 8. 食材检测图片

通过分析准确率及食材数据集可得出以下特点及其有待改进的地方：

1) 食材数据集中的食材图片较为简单，背景较为单一，模型容易过拟合，泛化性不足，可能会在复杂背景的食材检测中表现不佳，其中有部分食材种类的检测准确度较低，应该收集更多复杂的食材数据，提升模型的泛化性。

2) 食材数据集种类只有 10 种，每一种类的数据数量只有 200 多张，样本数量较少，人工数据收集难度大，实用性不高，可以通过数据增强增加样本数量提升实用性。

YOLOv3-MobileNet 模型比 YOLOv3 模型在运行速度上有很大的提升，并且使用 CWD-NMS 算法后模型的准确度有一定的提升；CWDNMS-lightweight YOLOv3 模型在 VOC 2007 数据集的准确度表现不错，并且在食材数据集的准确度表现方面有一定的提升，主要提升空间在于食材数据集的收集以及制作方面，

更好的数据集能够给模型带来更高的准确度。

5. 结论

本文介绍了一种基于轻量级神经网络的食材识别方法，通过比较各个神经网络的表现选择了YOLOv3作为特征提取网络，然后为了满足移植到嵌入式设备的需求，对YOLOv3结构进行轻量化处理，使用轻量化神经网络MobileNet代替原有的darknet53，构建YOLOv3-MobileNet神经网络模型，使模型达到轻量化的目的并且在准确度方面保持优势，最后使用CWD-NMS算法提升模型的准确度，最终的实验结果证明YOLOv3-MobileNet比YOLOv3在不明显降低准确度的前提下快了一倍，使用CWD-NMS算法能够提升模型7%的准确度，并且因为食材的数据集收集及处理方面相对于VOC2007表现更好，所以模型在食材识别方面有一定的准确度提升空间。CWDNMS-lightweight YOLOv3模型已经达到移植其他嵌入式设备对需求，能够应用在嵌入式的菜谱应用程序中，为菜谱应用程序提供食材识别的应用场景。

基金项目

国家自然科学基金项目(项目编号：61702110)；广东省重大科技专项项目(项目编号：2016B010108004)。

参考文献

[1] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2013) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587. https://doi.org/10.1109/CVPR.2014.81

[2] Girshick, R. (2015) Fast R-CNN. 2015 *IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., *et al*. (2016) SSD：Single Shot MultiBox Detector. *Proceedings of European Conference on Computer Vision*, Amsterdam, 8-16 October 2016, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[4] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detectiono. 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[5] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. *Proceedings of* 2017 *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 6517-6525. https://doi.org/10.1109/CVPR.2017.690

[6] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement. arXiv: 1804.02767.

[7] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[8] Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv: 1409.1556.

[9] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A. (2017) Inception-V4, Inception-Resnet and the Impact of Residual Connections on Learning. *Proceedings of the* 31*rd AAAI Conference on Artificial Intelligence*, San Francisco, 4-9 February 2017, 4278–4284.

[10] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016) Rethinking the Inception Architecture for Computer Vision. *Proceedings of the* 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 2818-2826. https://doi.org/10.1109/CVPR.2016.308

[11] Wang, M., Liu, B. and Foroosh, H. (2017) Factorized Convolutional Neural Networks. *Proceedings of the* 2017 *IEEE International Conference on Computer Vision Workshops*, Venice, 22-29 October 2017, 545-553. https://doi.org/10.1109/ICCVW.2017.71

[12] Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J. and Keutzer, K. (2016) SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and< 0.5 MB Model Size. arXiv: 1602.07360.

[13] Wu, J., Leng, C., Wang, Y., Hu, Q. and Cheng, J. (2016) Quantized Convolutional Neural Networks for Mobile Devices. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016,

4820-4828. https://doi.org/10.1109/CVPR.2016.521

[14] Rastegari, M., Ordonez, V., Redmon, J. and Farhadi, A. (2016) XNOR-Net: Imagenet Classification Using Binary convolutional Neural Networks. *European Conference on Computer Vision*, Amsterdam, 8-16 October, 525-542. https://doi.org/10.1007/978-3-319-46493-0_32

[15] Han, S., Mao, H. and Dally, W.J. (2015) Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. arXiv: 1510.00149.

[16] He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J. and Han, S. (2018) AMC: AutoML for Model Compression and Acceleration on Mobile Devices. *Proceedings of the European Conference on Computer Vision*, Munich, 8-14 September, 815-532. https://doi.org/10.1007/978-3-030-01234-2_48

[17] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., *et al*. (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv: 1704.04861,.

[18] Sifre, L. and Mallat, S. (2014) Rigid-Motion Scattering for Texture Classification. arXiv: 1403.1687.

[19] Jin, J., Dundar, A. and Culurciello, E. (2014) Flattened Convolutional Neural Networks for Feedforward Acceleration. arXiv: 1412.5474,.

[20] Bolya, D., Zhou, C., Xiao, F. and Lee, Y.J. (2019) YOLACT: Real-Time Instance Segmentation. *Proceedings of the* 2019 *IEEE International Conference on Computer Vision*, Seoul, 27 October-2 November 2019, 9157-9166. https://doi.org/10.1109/ICCV.2019.00925

[21] Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., *et al*. (2020) Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. arXiv: 2005.03572.

[22] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R. and Ren, D. (2020) Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, 12993-13000. https://doi.org/10.1609/aaai.v34i07.6999

[23] Ning, C., Zhou, H., Song, Y. and Tang, J. (2017) Inception Single Shot Multibox Detector for Object Detection. 2017 *IEEE International Conference on Multimedia & Expo Workshops* (*ICMEW*), Hong Kong, 10-14 July 2017, 549-554. https://doi.org/10.1109/ICMEW.2017.8026312