第 36 卷 第 3 期
2021 年 6 月

天津科技大学学报
Journal of Tianjin University of Science & Technology

Vol. 36 No. 3
Jun. 2021

DOI:10.13364/j.issn.1672-6510.20200164

# 基于 YOLOv3 的轻量化高精度多目标检测模型

陈晓艳[1]，任玉蒙[1]，张东洋[1]，洪 耿[1]，许能华[2]，闫潇宁[2]
（1. 天津科技大学电子信息与自动化学院，天津 300222；2. 深圳市安软科技股份有限公司，深圳 518131）

**摘 要**：针对当前目标检测模型在边缘设备中的应用占用内存过大、无法达到实时性要求的问题，提出一种基于 YOLOv3 的轻量化多目标检测模型. 采用 MobileNet 网络进行点卷积和深度可分离卷积运算提取图像特征，显著降低了模型的参数量. 同时，为了保证目标检测精度，在训练过程中不仅采用 CIOU（complete intersection over union）目标框回归损失函数，而且在损失函数中引入 Focal loss，减少正负样本分布不平衡所造成的误差；引入 Label Smoothing 调整真实样本标签类别在计算损失函数时的权重，有效抑制过拟合问题. 经 3.5 万个实际场景数据训练，本文提出的改进模型在行人和车辆的检测精度上分别达到 47.3%和 69.67%，模型大小仅为 YOLOv3 的 40%，实现了理想检测精度水平下的模型轻量化.

**关键词**：多目标检测；轻量化模型；YOLOv3；CIOU；Focal loss
**中图分类号**：TP391 **文献标志码**：A **文章编号**：1672-6510（2021）03-0033-06

# Lightweight High Precision Targets Detection Model Based on YOLOv3

CHEN Xiaoyan[1], REN Yumeng[1], ZHANG Dongyang[1], HONG Geng[1],
XU Nenghua[2], YAN Xiaoning[2]
（1. College of Electronic Information and Automation, Tianjin University of Science & Technology, Tianjin 300222, China;
2. Shenzhen Softsz Co., Ltd., Shenzhen 518131, China）

**Abstract**: Aiming at the problem that the current target detection algorithm occupies too much memory in the application of the edge device and cannot meet the real-time requirements, this article proposes an improved lightweight targets detection model based on YOLOv3 algorithm. MobileNet was adopted to carry out point convolution and deep separable convolution for features extracting, which significantly reduces the number of parameters of the model. Meanwhile, in order to ensure the accuracy of target detection, complete intersection over union (CIOU) bounding box regression loss function was used in the training process. In addition, Focal loss was introduced to reduce the errors caused by the unbalanced distribution of positive and negative sample distributions. Moreover, Label Smoothing was taken as an optimized strategy to adjust the weight of the real sample label category in the calculation of the loss function, which is helpful to avoid the overfitting problem. After training on 35 000 actual scene data, the proposed model improves 47.3% and 69.67% detection accuracy of pedestrians and vehicles respectively, and the model size is 40% of YOLOv3, thus achieving the targets detection with lightweight model and satisfying precision.

**Keywords**: targets detection; lightweight model; YOLOv3; CIOU; Focal loss

当前行人车辆检测算法在边缘设备中的应用占用内存过大，且无法达到实时性要求. 随着智慧城市建设和人工智能及大数据技术的迅猛发展，实现各种场景中的行人和车辆等重要目标的精准检测成为智慧城市的重要任务. 目标检测一直被认为是计算机视觉领域中最具挑战性的研究课题之一，这是由于它

**收稿日期**：2020-10-12；**修回日期**：2021-01-05
**基金项目**：天津市重点研发计划科技支撑重点项目（18YFZCGX00360）
**作者简介**：陈晓艳（1973—），女，天津人，教授，cxywxr@tust.edu.cn

需要在给定图像中精确定位特定目标类的对象，并为每个检测目标分配一个对应类的标签[1-2].

基于图像分类的传统目标检测方法是在检测图像中提取若干区域，用训练好的分类器逐个判断每个区域的所属类别[3]. 检测过程包含图像预处理、特征提取、特征分类及后处理等阶段.

HOG-SVM(histogram of oriented gradient，HOG；support vector machine，SVM)被描述为传统的目标检测算法中最成功的行人目标检测方法，并在一些实际场景中得到了广泛的运用[4]；但该方法对物体的轮廓特征提取不够准确，且特征提取能力不够稳定. Krizhevsky 等[5]提出的 AlexNet 网络，开启了基于卷积神经网络(convolutional neural networks，CNN)的目标检测研究. 二阶检测器 R-CNN 系列算法[6-7]提出的以样本候选框提取特征的目标检测算法受到极大的关注，目标检测精度明显提高，但是由于选择性搜索得到的有效候选框的数量太多，卷积神经网络计算量十分庞大，要得到理想的检测精度，对时间和算力的要求显著提高.

近几年，YOLO(you only look once)系列算法[8-10]给目标检测领域带来了全新的思路，即将分类任务和定位任务进行合并，图片经过一次特征提取之后就可以获取目标的位置和类别. 尤其是 YOLOv3 模型，它使用了 end-to-end 的设计思路，将整张图片进行特征提取，并将目标检测问题转换成单一的回归问题，可直接计算出多种目标的分类结果与位置坐标，既保证了检测的速度又保证了检测的精度[10].

然而，YOLOv3 模型有 235 MB 内存，参数量约 65 MB，检测速度不够理想，在模型的轻量化和处理速度上存在很大的提升空间. 模型的轻量化意味着对终端图像处理芯片的需求降低，更便于在嵌入式设备上部署，这对智慧城市的建设具有十分重要的意义，不仅节约建设成本，而且提升城市安防的智能化水平[11]. 本文在 YOLOv3 基础上提出一种轻量化高精度模型，使其在目标检测保持高精度的同时，大幅减小模型，提升处理速度. 主要针对特征提取网络、交并比计算以及损失函数进行了改进：将 MobileNet 作为特征提取网络，即将标准卷积替换成点卷积和深度可分离卷积，降低大量参数量；采用 CIOU(complete intersection over union)进一步精确计算目标框和预测框的距离，并反映两者的重合度大小；在损失函数中引入了 Focal loss，解决正负样本分布不平衡以及简单样本与复杂样本不平衡所造成的误差；同时在训练过程中引入标签平滑(Label Smoothing)抑制过拟合.

## 1　目标检测算法

YOLOv3 网络的基本结构分为两个部分，如图 1 所示，特征提取部分和预测部分. 其中特征提取网络 Darknet53 由 1 个 CBL 层和 5 个残差块组成. CBL 层是由卷积层、BN 层和 LeakyRelu 激活函数组成. 残差块结构如图 1 绿框所示，由 2 个 CBL 单元和 1 个残差边组成. 而预测部分融合了 3 种不同尺度的边界框. Darknet53 提取出来的 3 个特征图经过卷积层加强特征提取之后进行上采样，再次进行特征融合来丰富不同尺度的特征.

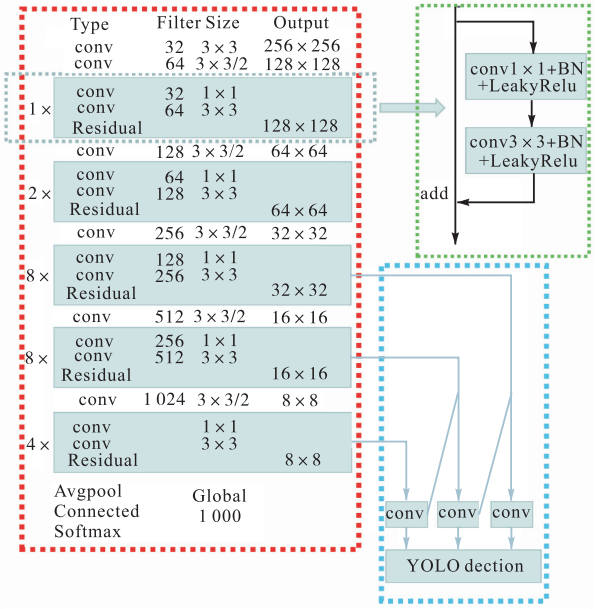


**图 1　YOLOv3 网络结构图**

**Fig. 1　YOLOv3 network structure diagram**

本文算法是在 YOLOv3 算法的基础上进行改进，其检测流程是先输入大小为 418×418 的图片，将图片划为 $S\times S$ 个网格，检测每个网格是否含有目标中心点. 若有目标中心点，则计算该物体属于某一类的后验概率并同时预测多个目标边框. 每个被预测的边框包含 5 个参数，分别为目标边框的中心点坐标 $(x,y)$、宽高 $(w,h)$ 和置信度评分 $S_i$. 将置信度小于阈值的边界框置零，采用非极大值抑制算法筛除冗余的边界框，最终确定目标的位置. 本文中的 YOLO-A 模型是将 YOLO 的特征提取网络替换成 MobileNet. YOLO-B 模型在 YOLO-A 的基础上将

IOU(intersection over union)改进为 CIOU、在损失函数中引入 Focal loss 以及引入 Label Smoothing.

### 1.1　特征提取网络

MobileNet 是 Google 公司在 2017 年提出用于移动端和边缘设备中的轻量级网络. 它提出的一种新的卷积思路是将标准卷积替换成点卷积和深度可分离卷积，大大地降低了模型的计算量. 深度可分离与标准卷积的卷积核运算如图 2 所示.

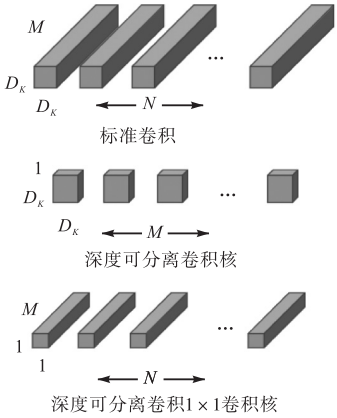


**图 2　标准卷积与深度可分离卷积**

**Fig. 2　Standard convolution with depthwise separable convolution**

假设输入一张大小为 $D_F \times D_F \times M$ 的图片，输出图片大小为 $D_F \times D_F \times N$，卷积核大小为 $D_K \times D_K$. 计算量的计算公式为

$$\mathrm{FLOPS} = (C_i \times D_K \times D_K) \times C_o \times D_F \times D_F \tag{1}$$

式中：FLOPS 表示计算量，$C_i$ 为输入通道，$C_o$ 为输出通道，$D_K \times D_K$ 为卷积核大小，$D_F \times D_F$ 为输出图片大小.

若采用标准卷积，则计算量为式(2)；若采用深度可分离卷积，则计算量为式(3).

$$O_1 = D_K \times D_K \times M \times N \times D_F \times D_F = M \times D_F \times D_F \times (D_K \times D_K \times N) \tag{2}$$

$$O_2 = D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F = M \times D_F \times D_F \times (D_K \times D_K + N) \tag{3}$$

本文算法的特征提取网络中 $N$ 的值远大于 1. 由公式计算可得，深度可分离的计算量远远小于采用普通卷积所带来的计算量.

### 1.2　CIOU 精选预测框

交并比 IOU 用于表示目标框和预测框的交集和并集之比，它不仅可以用来确定正样本和负样本，还可以反映预测框的检测效果[12]. 若预测框和真实框相近，IOU 值就大，反之 IOU 的值就小. IOU 示意图如图 3 所示，计算公式为

$$\mathrm{IOU} = \frac{A \cap B}{A \cup B} \tag{4}$$

采用 IOU 度量两个框之间的距离和重合程度，在训练过程中存在两种极端的情况：其一，当两个框没有交集，即 IOU = 0 时，无法表示两者的距离，也无法反映两者的重合度大小；其二，如图 4(a)和(b)，其 IOU 值虽然相等，但拟合程度是完全不一样的.

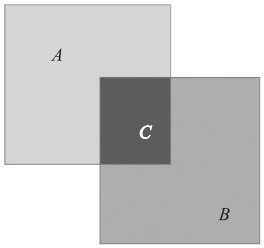


**图 3　IOU 示意图**

**Fig. 3　The schematic diagram of IOU**

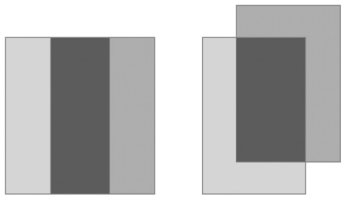


**图 4　目标框和预测框重叠情况示意图**

**Fig. 4　Schematic diagram of overlap between target box and prediction box**

针对上述情况，采用 CIOU 度量目标框和预测框的距离与重合程度. CIOU 在充分利用尺度不变性的基础上，融合目标框与预测框之间的距离及其重合程度[13]，并将预测框长和宽的比值作为惩罚项，从而使预测框的效果更加稳定. CIOU 示意图见图 5.

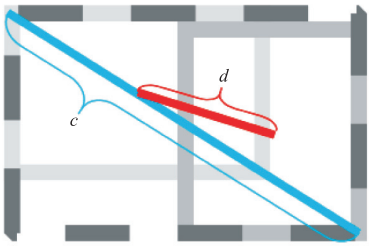


**图 5　CIOU 示意图**

**Fig. 5　The schematic diagram of IOU**

CIOU 公式为

$$\mathrm{CIOU} = \mathrm{IOU} - \frac{\rho^2(b, b^{\mathrm{gt}})}{c^2} - \alpha v \tag{5}$$

其中：$\alpha$ 为权重函数，$v$ 为衡量长宽比的相似性参数，$\alpha$ 和 $v$ 的公式为

$$\alpha=\frac{v}{1-\mathrm{IOU}+v} \tag{6}$$

$$v=\frac{4}{\pi^2}\left(\arctan\frac{w^{\mathrm{gt}}}{h^{\mathrm{gt}}}-\arctan\frac{w}{h}\right)^2 \tag{7}$$

图中：$d=\rho^2(b,b^{\mathrm{gt}})$ 是预测框和真实框的中心点的欧氏距离，$b=(x,y,w,h)$ 是真实框的信息，$b^{\mathrm{gt}}=(x^{\mathrm{gt}},y^{\mathrm{gt}},w^{\mathrm{gt}},h^{\mathrm{gt}})$ 是预测框的信息；$c$ 是包含预测框和真实框的最小矩形区域的对角线距离.

### 1.3　损失函数

在 YOLOv3 模型中，由于产生的先验预测框绝大部分都不包含目标，这就造成了正负样本比例失衡，导致大量的负样本影响损失函数，少量正样本的关键信息不能在损失函数中发挥正常的作用. 借鉴 Focal loss 损失函数概念[14]，通过减少负样本在样本总量的权重，使得模型在训练时更专注于难分类的样本，即正样本可以发挥出在损失函数中的作用.

Focal loss 是在交叉熵损失函数基础上进行修改，以二分类（即 0 和 1 两个类）交叉熵损失函数为例，表达式为

$$\begin{aligned}&\mathrm{loss}=-y\log y'-(1-y)\log(1-y')=\\&\begin{cases}-\log y', & m=0\\-\log(1-y'), & m=1\end{cases}\end{aligned} \tag{8}$$

令 $y_t=\begin{cases}y' & m=1\\1-y' & m=0\end{cases}$，则 $\mathrm{loss}=-\log y_t$.

$y'$是经激活函数的输出概率，在 0 ~ 1 之间，对于输出为 1 的正样本，若输出概率越大则损失函数越小；对于输出为 0 的负样本而言，若输出概率越小则损失函数越小. 由于输出为 0 的负样本太多，导致对损失函数的优化比较缓慢，难以获得损失函数的最优解. 为了降低负样本在损失函数中所占的权重，充分发挥正样本在损失函数中所占的比重，即加入平衡系数 $\alpha$ 以降低负样本的比重[15]，得到的损失函数为

$$\mathrm{loss}=-\alpha\log y_t \tag{9}$$

在此基础上又增加一个动态缩放因子 $\gamma$，自动降低简单样本的损失，帮助模型更好地训练困难的样本，最终形成的 Focal loss 为

$$\mathrm{loss}=-\alpha(1-y_t)^{\gamma}\log y_t \tag{10}$$

### 1.4　Label Smoothing

在处理人车分类问题时，当使用最小化交叉熵损失函数更新模型参数时，模型的泛化能力弱，容易导致过拟合，所以先引入一个与样本无关的分布 $u(i)$（平滑因子），将标签 $m$ 修正为 $m'$，见式（11），达到抑制过拟合的目的.

$$m'=(1-\varepsilon)\times m+\varepsilon u(i) \tag{11}$$

其中：$i$ 为标签类别数目，$u(i)$ 的取值 1/$i$，本文中共有人、车两类，故 $u(i)=0.5$；$\varepsilon$ 是伸缩因子，用来调整平滑之后标签数值的大小，有效抑制过拟合. 本文 YOLO-B 网络检测示意图如图 6 所示.

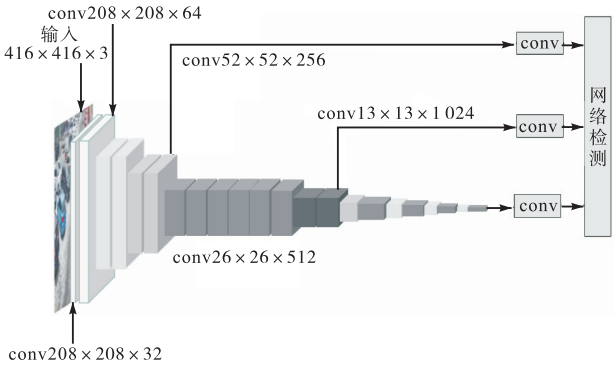


**图 6　YOLO-B 网络检测示意图**

**Fig. 6　YOLO-B schematic diagram of network detection**

## 2　实验及测试结果

本文所有的算法都是基于 Keras 框架实现的，并使用深圳市安软科技有限公司提供的 39 064 张已标注的数据集（以下简称安软数据集），其中验证集包含 3 982 张图片，测试集图片包含 500 张，测试集中人和车两类目标及其目标框数量如图 7 所示，每张图片的大小为 1 920 × 1 080. 实际应用场景包括人和车相交互的近景、远景、白天和夜景. 本文任务是在实际应用场景下对行人和车辆进行快速准确地检测.

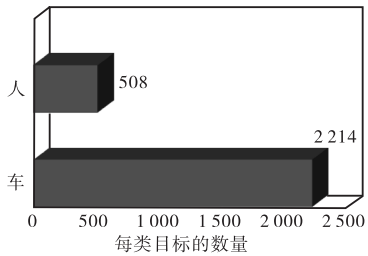


**图 7　测试集两类目标及目标框数量**

**Fig. 7　Number of objects per class and target boxes in the two categories of test set**

训练环境采用的是 Ubuntu 18.04 操作系统，CPU 为 Intel XeonE5-2360，显卡为两块 NVIDIA GEFORCE GTX1080Ti 11 GB. 实验过程中 batch size 设置为 16，即每个 epoch 在训练集中取 16 个样本进行

训练，直至全部样本训练完成一次即为 1 个 epoch. 模型训练时共训练 100 个 epoch. 测试实验环境采用的是 Windows10 操作系统，CPU 为 i5-9400F@2.9 GHz，显卡为 NVIDIA GEFORCE GTX1660 6 GB. 由于数据集所含目标框的数量不同，导致每张图片检测的时间有所不同. 利用 500 张测试集的测试总时间计算平均检测速度(FPS). 采用平均准确率(mAP)及模型大小等指标对目标检测模型进行评估.

为了验证本文 YOLO-B 算法的性能，采用 YOLOv3 原型、YOLO-Slim[16]、YOLOv3-tiny、YOLOv3-D[17]、YOLOv3–剪枝和 YOLO-A 进行对比. 其中，YOLOv3 原型是由 YOLOv3 作者提出没有加以改进的模型；YOLO-Slim 是 YOLOv3 的骨干网络替换成 MobileNetV2 并进行剪枝处理，其目的是为了降低参数量和模型大小；YOLOv3-tiny 是一个公开的目标检测轻量级模型；YOLOv3-D 模型是在 YOLOv3 的基础上引入 CIOU 和 Focal loss，从而提高目标检测的精度，但是模型大小不会发生改变；YOLOv3–剪枝是在 YOLOv3 的基础上降低通道数，从而实现减小模型、降低计算量和增加检测速度的目的.

### 2.1　平均检测速度测试结果

对 7 种模型分别进行训练和测试，平均检测速度见表 1. YOLOv3 的检测速度为 11.2 帧/秒，YOLOv3–剪枝的检测速度为 14.3 帧/秒，YOLOv3-tiny 是一种以 YOLOv3 为基础的轻量化模型，其检测速度达到 25.1 帧/秒，检测速度优势明显. YOLO-Slim 的检测速度略微逊色于 YOLOv3-tiny，而本文提出的 YOLO-A 和 YOLO-B 的检测速度均为 20.6 帧/秒，快于 YOLOv3，检测速度相对提升了 83.9%. 由于 YOLO-B 相对于 YOLO-A 来说参数量相同，故 YOLO-A 和 YOLO-B 的检测速度是一样的.

**表 1　目标检测算法性能评价指标对比结果**

**Tab. 1　Comparison results of performance evaluation indexes of target detection algorithm**

| 模型 | 训练集 | 验证集 | 测试集 | 平均准确率/% | 模型大小/MB | 精确率/% | | 平均检测速度/(帧/秒) |
|---|---|---|---|---|---|---|---|---|
| | | | | | | 行人 | 车辆 | |
| YOLOv3 | 35 082 | 3 982 | 500 | 57.80 | 235 | 45.52 | 70.08 | 11.2 |
| YOLOv3-D | 35 082 | 3 982 | 500 | 58.32 | 235 | 45.69 | 71.02 | 11.2 |
| YOLOv3–剪枝 | 35 082 | 3 982 | 500 | 36.69 | 79.1 | 46.93 | 36.42 | 14.3 |
| YOLO-Slim | 35 082 | 3 982 | 500 | 46.69 | 28.2 | 46.95 | 55.36 | 21.2 |
| YOLOv3-tiny | 35 082 | 3 982 | 500 | 27.23 | 33.2 | 16.56 | 37.89 | 25.1 |
| YOLO-A | 35 082 | 3 982 | 500 | 58.59 | 92.6 | 47.30 | 69.67 | 20.6 |
| YOLO-B | 35 082 | 3 982 | 500 | 59.23 | 92.6 | 48.29 | 70.16 | 20.6 |

### 2.2　平均准确率测试结果和模型大小对比

7 种模型的平均准确率测试结果见表 1. 由表 1 可知：YOLOv3 的平均准确率值为 57.80%，YOLOv3-D 的平均准确率为 58.32%，两者表现很好，但是模型太大；YOLOv3–剪枝、YOLOv3-tiny 和 YOLO-Slim 的模型比较小，但是平均准确率均低于 50%；YOLO-A 和 YOLO-B 的平均准确率比其他 5 种模型都高. YOLO-A 和 YOLO-B 的模型大小相同，由于通过改变损失函数以及其他优化方法使得 YOLO-B 的平均准确率高于 YOLO-A 的，达到 59.23%，网络模型大小仅为原始 YOLOv3 的 40%.

### 2.3　精确率测试结果

在测试集中对行人和车辆分别进行目标检测，检测精度见表 1. 结果表明，无论是检测行人还是检测车辆，YOLO-B 的精确率均高于其他 6 种算法. 通过进行对比得出 YOLO-B 算法综合表现优秀. YOLO-B 训练过程中的 loss 值变化如图 8 所示，随着迭代次数的不断增加，loss 值不断降低，最终达到稳定状态.

为了直观地表达检测效果，在 YOLO-B 算法训练完成之后，使用 Yolo-B 算法对测试集进行测试并对结果进行可视化，测试集的场景均为摄像头真实场景，图 9 为真实场景测试的效果.

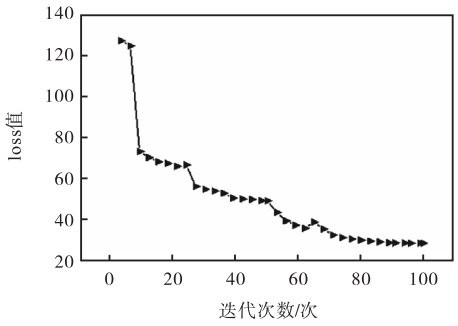


**图 8　YOLO-B 训练过程中的 loss 变化情况**

**Fig. 8　Changes of loss during YOLO-B training**


**图 9　YOLO-B 算法真实场景测试效果图**

**Fig. 9　YOLO-B algorithm real scene test renderings**

## 3　结　语

针对模型轻量化的需求，调整 YOLOv3 的网络结构. 首先在 YOLOv3 的基础上将特征提取网络替换成 MobileNet，大幅度降低模型大小；其次是将 IOU 改进为 CIOU 并在损失函数中引入 Focal loss，并且又引入 Label Smoothing，最终形成 YOLO-B 网络模型. 通过对模型的对比实验以及测试结果显示：网络模型大小仅为原始 YOLOv3 的 40%，并且通过模型优化策略，保证了检测的精度.

本文提出的网络模型在公共安全监控中能有效地做到快速准确的检测要求，为嵌入式平台应用提供了一种高性能的轻量化模型结构. 下一步的工作是将该算法移植部署至多路摄像头终端，进行实际场景的应用.

**参考文献：**


[1] Yin Y，Li H，Fu W. Faster-YOLO：An accurate and faster object detection method[J]. Digital Signal Processing，2020，102：102756.

[2] Wu X，Doyen S，Teven C H H. Recent advances in deep learning for object detection[J]. Neuro Computing，2019，396：39–64.

[3] Zhao Z Q，Zheng P，Xu S T. Object detection with deep learning：A review[J]. IEEE Transactions on Neural Networks and Learning Systems，2019，30(11)：3212–3232.

[4] Aslan M F，Durdu A，Sabanci K，et al. CNN and HOG based comparison study for complete occlusion handling in human tracking[J]. Measurement，2020，158：107704.

[5] Krizhevsky A，Sutskever I，Hinton G. ImageNet classification with deep convolutional neural networks [EB/OL]. [2020–10–11]. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

[6] Girshick R. Fast R-CNN[EB/OL]. [2020–10–11]. https://arxiv.org/abs/1504.08083.

[7] Ren S，He K，Girshick R，et al. Faster R-CNN：Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence，2017，39(6)：1137–1149.

[8] Redmon J，Farhadi A. YOLO9000：Better，faster，stronger[C]// IEEE. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Washington DC：IEEE Computer Society Press，2017：6517–6525.

[9] Redmon J，Divvala S，Girshick R，et al. You only look once：Unified，real-time object detection[C]// IEEE. CVPR2016 Conference on Computer Vision and Pattern Recognition. Washington DC：IEEE Computer Society Press，2016：779–788.

[10] Redmon J，Farhadi A. YOLOv3：An incremental improvement[EB/OL]. [2020–10–11]. https://arxiv.org/pdf/1804.02767.pdf.

[11] Howard A G，Zhu M，Chen B，et al. MobileNets：Efficient convolutional neural networks for mobile vision applications[EB/OL]. [2020–10–11]. https://arxiv.org/abs/1704.04861.

[12] He K，Zhang X Y，Re S Q，et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2015，37(9)：1904–1916.

[13] Zheng Z H，Wang P，Liu W，et al. Distance-IOU loss：Faster and better learning for bounding box regression [EB/OL]. [2020–10–11]. https://arxiv.org/pdf/1911.08287.pdf.

[14] 陈幻杰，王琦琦，杨国威，等. 多尺度卷积特征融合的 SSD 目标检测算法[J]. 计算机科学与探索，2019，13(6)：1049–1061.

[15] Lin T Y，Goyal P，Girshick R，et al. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2020，42(2)：318–327.

[16] 邵伟平，王兴，曹昭睿，等. 基于 MobileNet 与 YOLOv3 的轻量化卷积神经网络设计[J]. 计算机应用，2020，40(S1)：8–13.

[17] 邹承明，薛榕刚. 融合 GIoU 和 Focal loss 的 YOLOv3 目标检测算法[J]. 计算机工程与应用，2020，56(24)：214–222.


**责任编辑：**郎婧