



## 改进 SSD 算法及其在地铁安检中的应用

张 震, 李孟洲, 李浩方, 马军强

(郑州大学 电气工程学院, 郑州 450001)

**摘 要:** 针对传统 SSD 算法在检测小目标时容易漏检且检测精度不高的问题, 提出一种改进算法。对 SSD 算法中各尺度特征进行尺寸大小不变的卷积操作, 将卷积前后对应的特征进行轻量级网络融合, 从而生成新的金字塔特征层, 并加入基于残差模块的检测单元避免增加网络模型容量和运算复杂度, 同时增强对小尺度目标的检测能力。基于 PASCAL-VOC2007 小目标数据集的实验结果表明, 与传统 SSD、YOLOv3、Faster RCNN 等算法相比, 在 PASCAL-VOC2007 小目标数据集中, 该算法的 mAP 指标较传统 SSD 算法提高 8.5%, 较 Faster RCNN 算法提高 3.9%, 较 YOLOv3 提高 2%, FPS 达到 83 frame/s, 其检测地铁安检图片的 mAP 达到 77.8%。

**关键词:** SSD 算法; 网络融合; 金字塔特征层; 残差模块; 检测单元; 目标检测

开放科学(资源服务)标志码(OSID):



**中文引用格式:** 张震, 李孟洲, 李浩方, 等. 改进 SSD 算法及其在地铁安检中的应用[J]. 计算机工程, 2021, 47(7): 314-320.

**英文引用格式:** ZHANG Z, LI M Z, LI H F, et al. Improved SSD algorithm and its application in subway security detection[J]. Computer Engineering, 2021, 47(7): 314-320.

## Improved SSD Algorithm and Its Application in Subway Security Detection

ZHANG Zhen, LI Mengzhou, LI Haofang, MA Junqiang

(School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China)

**[Abstract]** In the detection of small objects, the traditional SSD algorithm tends to miss the targets, and its detection accuracy is reduced. To address the problem, an improved algorithm is proposed. Each scale feature in the SSD algorithm is convoluted with the size being unchanged, and the corresponding features before and after the convolution are fused by using a lightweight network to generate a new set of pyramid feature layers. Then a detection unit based on the residual module is added to avoid increasing the network model capacity and computational complexity, and to enhance the detection ability of small targets. The experimental results on the PASCAL-VOC2007 small target data set show that, compared with traditional SSD, YOLOv3, Faster RCNN algorithms etc., the mAP of the proposed algorithm is 8.5% higher than that of the traditional SSD, 3.9% higher than that of the Faster RCNN algorithm, and 2% higher than that of the YOLOv3 algorithm. The proposed algorithm increases the FPS to 83 frames/s, and the mAP of the subway security image check to 77.8%.

**[Key words]** SSD algorithm; network convergence; pyramid feature layer; residual module; detection unit; object detection

DOI: 10.19678/j.issn.1000-3428.0058283

### 0 概述

近年来,随着互联网大数据的快速发展、计算机运算能力的提升以及国内外在深度学习研究上的突破,目标检测技术日趋成熟。目前,将目标检测与实际应用场景结合是一个热门方向<sup>[1]</sup>,但较少用于地铁安检方面,地铁安检仍采用人为识别的方法。由

于安检工作人员的熟练程度以及精神状态都可能会影响安检的准确率<sup>[2]</sup>,因此将目标检测用于地铁安检具有重要的研究意义。

GIRSHICK 等于 2014 年提出了 R-CNN (Regions with CNN Features)<sup>[3]</sup>算法,促使深度学习在目标检测领域快速发展。此后, SPP-Net (Spatial Pyramid Pooling Networks)、YOLO<sup>[4-5]</sup>、YOLO2、Fast RCNN<sup>[6]</sup>、

**基金项目:** 国家重点研发计划“公共安全风险防控与应急技术装备”重点专项(2018YFC0824XXX)。

**作者简介:** 张 震(1966—),男,教授,主研方向为多媒体信息安全、图像处理、模式识别;李孟洲、李浩方、马军强,硕士研究生。

**收稿日期:** 2020-05-11 **修回日期:** 2020-07-06 **E-mail:** zhangzhen66@126.com

Faster RCNN<sup>[7]</sup>、SSD<sup>[8]</sup>、DSSD<sup>[9]</sup>等算法的出现也进一步促进了目标检测的发展。目前主流的目标检测算法大致分为两类:一类是基于R-CNN进行展开,以Faster RCNN为代表的两阶段算法;另一类是以YOLO和SSD为代表的一阶段算法。两阶段算法先进行区域选取,再对所选区域进行分类,一阶段算法则是对产生的区域同时进行这两项操作。两阶段算法的检测精度略高于一阶段算法,但检测速度很慢,因此,两阶段算法不适用于地铁安检这类对实时性要求较高的应用场景。在一阶段算法中,SSD算法已被广泛应用于目标检测并取得了不错的效果,如文献[10]将SSD应用于舰船目标检测,文献[11]将SSD应用于柑橘实时分类检测,文献[12]将SSD应用于果园行人实时检测。

本文提出一种改进的SSD算法,通过轻量级网络融合对传统SSD算法中的金字塔特征层进行特征增强,同时加入一种基于残差模块的检测单元以避免增加网络模型容量,从而在提高检测精确度的同时保证识别速度。

## 1 SSD目标检测

SSD目标检测算法在YOLO的基础上融合了RPN的思想。该算法在多个不同尺度上对输入图片进行检测,且在各特征层上都使用不同的卷积模型,从而使模型具有更高的识别准确率。此外,作为一阶段算法,其检测速度比两阶段算法快得多。

虽然SSD能够改善传统算法对物体尺度敏感、对尺度变化大的物体泛化能力差的缺点,但仍存在对小目标检测精度低的问题<sup>[13-15]</sup>。对此,多种改进的SSD算法相继被提出<sup>[16-18]</sup>,如DSSD、DSOD等。DSSD将SSD的基础网络由VGG-16改为ResNet,并且引入反卷积层用于传递信息,尽管提高了识别小目标的准确率,但是网络结构复杂,使识别速度变慢,实时性变差。DSOD先将基础网络换为DenseNet,再融合两层特征图,不仅加快了训练速度,而且提高了检测性能。

### 1.1 传统SSD的网络结构

传统SSD检测模型结构分为两部分:第一部分是利用卷积神经网络来提取输入图像特征信息的分类网络VGG-16;第二部分是后端的多尺度检测网络。由于浅层的特征图适用于小目标检测,而深层的特征图适用于大目标检测,因此多尺度检测网络需要在不同深度的特征图上进行回归计算和极大值抑制,并输出最终结果。传统SSD网络结构如图1所示。

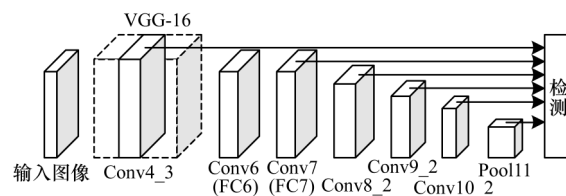


图1 传统SSD网络结构

Fig.1 Structure of traditional SSD network

### 1.2 传统SSD特征图默认边框尺度

SSD算法通过利用多尺度的方法来获取特征图。假设检测时采用的特征图层数为 $n$ ,那么第 $k$ 个默认框比例 $s_k$ 可表示为:

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{n-1} (k-1), k \in [1, n] \quad (1)$$

其中: $s_{\min}$ 表示特征层的默认框占输入图像的最小比例,一般取为0.2; $s_{\max}$ 表示特征层的默认框占输入图像的最大比例,一般取为0.95。为加强默认框适应检测物体大小的能力,SSD算法对相同特征层设置了不同宽高比的默认框。本文选择默认框的宽高比尺寸为 $r=\{1, 2, 3, 1/2, 1/3\}$ ,其中,默认框的宽度 $w_k^{(m)}$ 和高度 $h_k^{(m)}$ 分别为:

$$\begin{cases} w_k^{(m)} = s_k \sqrt{r^{(m)}} \\ h_k^{(m)} = s_k \sqrt{r^{(m)}} \end{cases} \quad (2)$$

其中, $m$ 表示5种默认框的宽高比尺寸, $m \in \{1, 2, 3, 4, 5\}$ 。

当默认宽高比等于1时,增加1个默认框比例 $s'_k$ :

$$s'_k = \sqrt{s_k s_{k+1}} \quad (3)$$

因此,SSD在每个特征图共产生6个默认框,第6个默认框的宽度和高度表示为:

$$w_k^{(6)} = h_k^{(6)} = \sqrt{s_k s_{k+1}} \quad (4)$$

设定默认框中心坐标为 $\left(\frac{a+0.5}{|f_k|}, \frac{b+0.5}{|f_k|}\right)$ ,其中, $|f_k|$ 代表第 $k$ 个特征图的尺寸, $a, b \in \{0, 1, \dots, |f_k|-1\}$ 。特征图的默认框坐标与原始图像的坐标映射关系如下:

$$x_{\min} = \frac{c_x - \frac{w_b}{2}}{w_{\text{feature}}} w_{\text{image}} = \left( \frac{a+0.5}{|f_k|} - \frac{w_k}{2} \right) w_{\text{image}} \quad (5)$$

$$y_{\min} = \frac{c_y - \frac{h_b}{2}}{h_{\text{feature}}} h_{\text{image}} = \left( \frac{b+0.5}{|f_k|} - \frac{h_k}{2} \right) h_{\text{image}} \quad (6)$$

$$x_{\max} = \frac{c_x + \frac{w_b}{2}}{w_{\text{feature}}} w_{\text{image}} = \left( \frac{a+0.5}{|f_k|} + \frac{w_k}{2} \right) w_{\text{image}} \quad (7)$$

$$y_{\max} = \frac{c_y + \frac{h_b}{2}}{h_{\text{feature}}} h_{\text{image}} = \left( \frac{b+0.5}{|f_k|} + \frac{h_k}{2} \right) h_{\text{image}} \quad (8)$$

其中:\$(c\_x, c\_y)\$代表特征层上默认框的中心坐标; \$w\_b\$ 和 \$h\_b\$ 分别代表默认框的宽度和高度; \$w\_{feature}\$ 和 \$h\_{feature}\$ 分别代表特征层的宽度和高度; \$w\_{image}\$ 和 \$h\_{image}\$ 分别代表原始图像的宽度和高度。求得的 \$(x\_{min}, y\_{min}, x\_{max}, y\_{max})\$ 代表第 \$k\$ 层特征图上中心坐标为 \$\left(\frac{a+0.5}{|f\_k|}, \frac{b+0.5}{|f\_k|}\right)\$、宽度和高度分别为 \$w\_k\$ 和 \$h\_k\$ 的默认框映射到原始图像的具体坐标。

### 1.3 传统 SSD 的损失函数

SSD 算法的损失函数可表示:

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \quad (9)$$

其中: \$x\$ 为当前预测框的类别匹配信息; \$c\$ 代表标注类别; \$l\$ 代表预测边界框坐标; \$g\$ 代表标注边界框坐标; \$N\$ 代表匹配到的 default box 的数量; \$L\_{conf}(x, c)\$ 代表分类损失; \$L\_{loc}(x, l, g)\$ 代表位置损失; \$\alpha\$ 代表两者的权重, 本文将其设置为 1。

\$L\_{loc}(x, l, g)\$ 借鉴 Faster RCNN 的位置回归函数 smooth\$\_{L1}\$, 可表示为:

$$L_{loc}(x, l, g) = \sum_{i \in P_{os}} \sum_{m \in B_{ox}} x_{ij,k} S_{L1}(l_{i,m}(l_{i,m} - \hat{g}_{j,m})) \quad (10)$$

其中: \$x\_{ij,p} = \{1, 0\}\$ 代表当前第 \$i\$ 个预测框匹配类别 \$p\$ 的第 \$j\$ 个目标框真值; \$P\_{os}\$ 代表样本中正例; \$B\_{ox}\$ 代表预测框中心的坐标及其宽和高的集合; \$l\_{i,m}\$ 代表先验框的预测值; \$\hat{g}\_{j,m}\$ 代表真实框的位置参数。\$S\_{L1}\$ 可表示为:

$$S_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & \text{其他} \end{cases} \quad (11)$$

\$L\_{conf}(x, c)\$ 用交叉熵损失函数可表示为:

$$L_{conf}(x, c) = - \sum_{i \in Pos} (x_{ij,p} \lg \hat{c}_{i,p}) - \sum_{i \in Neg} \lg \hat{c}_{i,0} \quad (12)$$

其中: \$\hat{c}\_{i,0}\$ 代表预测框中没有物体的概率; \$N\_{eg}\$ 代表样本中的负例; \$\hat{c}\_{i,p}\$ 为利用 softmax 函数计算出的概率值。\$\hat{c}\_{i,p}\$ 可表示为:

$$\hat{c}_{i,p} = \frac{\exp c_{i,p}}{\sum_p \exp c_{i,p}} \quad (13)$$

### 1.4 传统 SSD 目标检测存在的不足

传统 SSD 算法低层次的定位效果好但是分类精度低, 又由于 6 个不同特征图相互独立, 因此其对小目标的检测效果一般。造成以上情况的主要原因是:

1) 特征图维度会随着 CNN 层数的增加而变小, 进而得到语义特征更加明显的结果, 同时位置信息也变得更加模糊。如果采用的是 SSD\_300×300 的模型, 则主要依赖 Conv4\_3 检测小目标, 虽然其分辨率

较高, 而且先验框的尺寸和检测目标更接近, 但是特征表达的能力却仅来源于前 10 层卷积层, 显然不能很好地捕捉深层次语义信息。

2) 随着卷积层数的增加, 输入图像的感受野和映射区域也会扩大, 卷积感受野过大会降低精确定位目标的性能, 并且当映射区域过大时, 如果出现在检测区域包括多个检测目标的情况, 则无法有效地区分目标。

针对以上问题, 本文提出融合轻量级网络来改进传统 SSD 算法, 通过加入基于残差模块的检测单元来避免增加网络模型容量以及运算复杂度, 同时提高检测精确度。

## 2 改进的 SSD 算法

本文改进的 SSD 算法整体网络结构框图如图 2 所示, 首先对 Conv4\_3、Conv7、Conv8\_2、Conv9\_2、Conv10\_2 (简记为 ConvY\_Y) 这 5 个特征层进行卷积操作 (由于 Pool11 的特征层尺寸很小, 包含语义信息较多, 因此无需对其进行卷积操作)。为避免特征融合后再进行特征降维操作, 要求进行卷积操作后生成的 Conv4\_3\_0、Conv7\_0、Conv8\_2\_0、Conv9\_2\_0、Conv10\_2\_0 (简记为 ConvY\_Y\_0) 的特征维数不超过原始的特征维数<sup>[19]</sup>。然后将 Conv4\_3 和 Conv4\_3\_0、Conv7 和 Conv7\_0、Conv8\_2 和 Conv8\_2\_0、Conv9\_2 和 Conv9\_2\_0 以及 Conv10\_2 和 Conv10\_2\_0 这 5 对两两融合, 依次得到 Conv4\_3\_1、Conv7\_1、Conv8\_2\_1、Conv9\_2\_1 和 Conv10\_2\_1, 并与 Pool11 形成最终的金字塔特征层<sup>[19]</sup>。

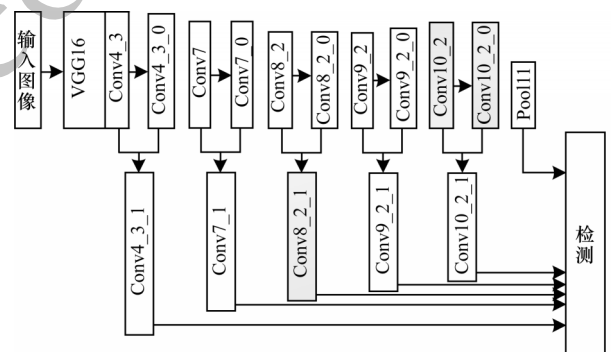


图 2 整体网络结构框图

Fig.2 Block diagram of overall network structure

ConvY\_Y\_0 比 ConvY\_Y 具有更强的语义信息和更大的卷积特征尺度。同时, 由于本文提出的卷积没有改变补边的特征尺寸, 因此能更好地保留原特征图的边缘信息。所以, 本文最终形成的金字塔特征层与传统 SSD 算法的金字塔特征层相比, 增强了特征层的语义信息, 改善了 SSD 算法对小物体识



别率低的问题。由于生成的特征层相比于最初的特征层特征维数没有升高,特征图的边缘特征没有损失,因此在特征融合时高层特征不仅无需调整尺寸,而且不用进行降维操作,相比于传统的特征融合操作,本文采用的特征融合方法更具优势。上述改进虽然提高了检测精度,但是难免会使检测网络复杂化,从而降低识别速率。针对以上情况,本文新加入基于残差模块的检测单元,从而实现特征图维度与主分支维度的统一,避免网络模型容量和运算复杂度增加。

## 2.1 特征融合策略

特征对应元素相加与特征连接是特征融合策略中最流行的2种融合方式<sup>[16,20]</sup>。基于特征连接的特征融合方式,本文设计一种轻量级特征融合方式对金字塔特征层进行特征增强。

在进行特征融合时,要求ConvY\_Y\_0与ConvY\_Y尺寸相同且ConvY\_Y\_0的维数不高于ConvY\_Y,因此,在进行特征融合的操作时不必调整特征图的尺寸,也不用对融合后的特征进行降维,从而大幅简化了操作。

在进行特征融合时,设 $x$ 为输入特征, $f(x)$ 表示对 $x$ 进行卷积操作, $y$ 表示 $x$ 卷积后的特征,则 $y=f(x)$ 。如果 $(x,y)$ 表示特征 $x$ 和特征 $y$ 进行首尾相连的操作,那么相应的特征连接操作可表示为 $F_{\text{concat}}=(x,y)$ ,表示特征 $x$ 和特征 $y$ 进行首尾相连操作后得到的特征。

特征连接的特征融合方式如图3所示,将ConvY\_Y\_0的特征维数统一设置为128维,低于ConvY\_Y特征层的特征维数,然后将ConvY\_Y与ConvY\_Y\_0串联得到ConvY\_Y\_1,将这种特征融合称为轻量级网络融合。

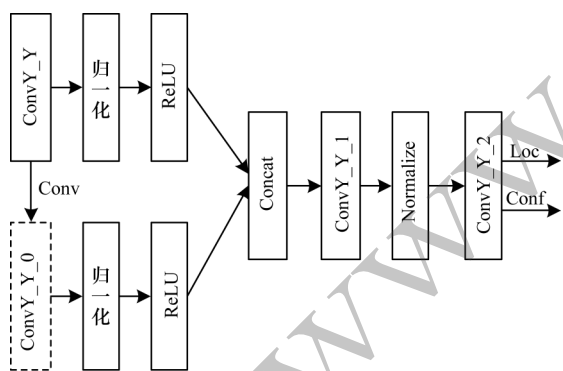


图3 特征连接的特征融合方式

Fig.3 Feature fusion mode of feature connection

## 2.2 基于残差模块的检测单元

本文通过加入一种基于残差模块的检测单元来避免增加网络模型的容量以及运算复杂度,并防止直接用主干网络来预测目标<sup>[21]</sup>。残差模块分为短接分支和主分支2个部分,其结构如图4所示。主分支共包含3个卷积层,即2个 $1\times 1$ 的卷积层和1个 $3\times 3$

的卷积层;短接分支主要作用是用1个 $1\times 1$ 的卷积层使特征图与主分支的维度达到统一<sup>[21]</sup>。在网络训练过程中,每个卷积层后都添加了1个非线性激活层,此做法不仅可以降低预测单元所产生的梯度值对主干网络形成的影响,而且还可以起到缓冲的作用。同时,由于残差模块的加入,预测层的网络深度加深,能有效改善浅层特征图语义表征能力较弱的问题<sup>[21]</sup>。本文新加入残差模块的作用虽然与插入 $3\times 3$ 卷积模块基本相同,但是残差模块中的 $1\times 1$ 卷积层可以将特征图调整到比较低的维度<sup>[21]</sup>。因此,本文新插入的残差模块要比直接插入1个 $3\times 3$ 卷积模块占用的模型参数低得多。

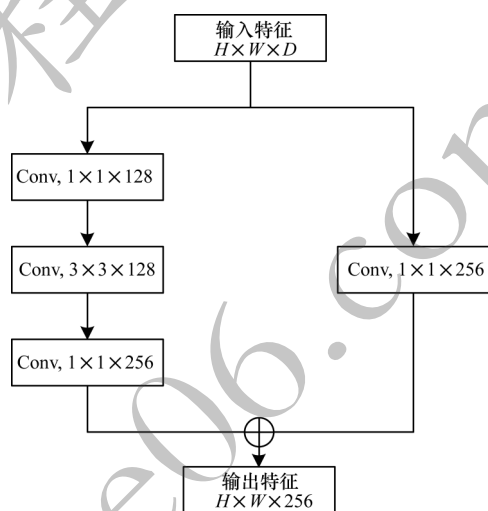


图4 ResBlock结构

Fig.4 Structure of ResBlock

本文采用类似Faster RCNN的anchor机制,Faster RCNN采用的是 $3\times 3$ 的卷积核,因为模型参数与运算量会随着卷积核的减小而降低,所以本文尝试采用 $1\times 1$ 的卷积核,经过实验发现 $1\times 1$ 的卷积核不仅没有明显降低检测精度,反而能够提高网络的运算速率。此外,由于加入的残差模块可以起到统一维度的效果,因此即使是不同预测单元上的分类器也能够在训练过程中达到参数共享。相比于传统SSD直接在网络卷积层上面提取目标,然后再针对不同的检测单元训练出其对应的分类器而言,本文所采取的操作要更为简单实用。综上,本文新加入的残差模块能够大幅降低网络模型大小,从而提高网络的运算速率。

## 2.3 网络模型的训练策略

在网络模型训练的过程中,需要对训练数据集进行翻转、旋转以及亮度与尺度变化,从而提高网络模型的稳健性。由于本文算法形成了新的金字塔特征层,与传统SSD相差较多,因此不能直接在已经训练好的SSD模型上修改。本文以VGG-16网络作为改进SSD算法的基础框架,训练位置损失函数采用

smooth L1,训练分类损失函数采用 Log loss。此外,本文算法的学习率、提取各个特征层所对应的 Box 长宽比以及挖掘决策困难样本均和传统 SSD 算法相同,IOU 值设置为 0.5。

本文使用的 GPU 型号为 GTX1080 Ti,基于 Ubuntu16.04 操作系统,在 TensorFlow 平台上完成对改进 SSD 算法的训练以及测试的工作。由于设备的限制,本文提出的算法采取单 GPU 训练。

### 3 实验与结果分析

本文算法主要解决了传统 SSD 算法对小目标识别率低的问题,并新加入基于残差模块的检测单元避免增加网络模型容量以及运算复杂度。实验通过对 mAP、FPS 的测试来验证本文算法对传统 SSD 的改进程度。此外,将本文训练好的算法用于地铁安检图像的检测,以验证其应用性能。

#### 3.1 实验步骤

首先将在 ImageNet 训练好的 VGG-16 网络作为本文算法的基础框架,然后在 PASCAL-VOC2007 的训练集与验证集上来训练本文算法,并以 mAP 和 FPS 作为性能指标在其测试集上检测本文算法对小目标检测的能力,最后与性能优异的改进 SSD 算法、YOLOv3 算法和 Faster RCNN 算法进行比较。

#### 3.2 评价指标

本文使用的评价指标为平均准确率均值(mAP)与检测速率(FPS)。其中,平均准确率均值是所有类别平均准确率(AP)的平均值,检测速率为每秒处理图像的帧数。

平均准确率定义为:

$$AP = \int_0^1 P(R) dR \quad (14)$$

其中, $R$  代表召回率, $P$  代表准确率。

#### 3.3 PASCAL-VOC2007 的测试结果

PASCAL-VOC2007 是 1 个标准的小目标数据集,其具体的物体类别见表 1。该数据集提供的图片集共分为 20 种类别。

表 1 PASCAL-VOC2007 具体的物体类别

Table 1 PASCAL-VOC2007 specific object categories

目录	类别
交通工具	飞机,自行车,船,公共汽车,小轿车,摩托车,火车
动物	鸟,猫,牛,狗,马,羊
室内	瓶子,椅子,餐桌,盆栽植物,沙发,电视
人类	人

为对比本文算法与传统 SSD、DSSD、DSOD、YOLOv3 和 Faster RCNN 的检测性能,从 PASCAL-VOC2007 数据集中挑选 189 具有代表性的图片,共涉及 11 种物体类别,分别为飞机、船、小轿车、鸟、猫、狗、马、羊、瓶子、盆栽植物和人。对这些图片进行相应的处理后,标注物体的 ground truth 共有 1 601 个。分别采用传统 SSD 算法、DSSD 算法、DSOD 算法、YOLOv3 算法、Faster RCNN 算法和本文算法进行目标检测实验,mAP 指标测试结果如表 2 所示,FPS 指标测试结果如表 3 所示。其中,本文算法与传统 SSD 算法是在单个 1080 Ti GPU 测试的,而其他算法则是在单个 Titan X GPU 上测试的。

表 2 不同算法在 PASCAL-VOC2007 数据集上的 mAP 指标

Table 2 mAP indicators of different algorithms on PASCAL-VOC2007 data set

算法	mAP 指标	检测精度										
		飞机	船	小轿车	鸟	猫	狗	马	羊	瓶子	盆栽植物	人
SSD	0.526	0.543	0.438	0.706	0.519	0.535	0.548	0.516	0.538	0.368	0.505	0.573
DSSD	0.619	0.621	0.563	0.771	0.611	0.608	0.612	0.609	0.630	0.557	0.599	0.623
DSOD	0.601	0.605	0.537	0.765	0.592	0.586	0.598	0.591	0.612	0.531	0.581	0.609
YOLOv3	0.591	0.598	0.532	0.749	0.571	0.562	0.591	0.584	0.605	0.527	0.578	0.599
Faster RCNN	0.572	0.569	0.481	0.727	0.558	0.543	0.572	0.562	0.584	0.526	0.576	0.592
本文算法	0.611	0.614	0.553	0.766	0.603	0.598	0.604	0.601	0.622	0.546	0.592	0.617

表 3 不同算法在 PASCAL-VOC2007 数据集上的 FPS 指标

Table 3 FPS indicators of different algorithms on PASCAL-VOC2007 data set

算法	GPU 配置	输入图像尺寸	FPS 指标/(frame·s <sup>-1</sup> )
SSD	Titan X	300 像素×300 像素	85.0
	1080 Ti	300 像素×300 像素	46.0
DSSD	Titan X	321 像素×321 像素	9.5
DSOD	Titan X	300 像素×300 像素	17.4
YOLOv3	Titan X	608 像素×608 像素	51.0
Faster RCNN	Titan X	600 像素×1 000 像素	7.0
本文算法	1080 Ti	300 像素×300 像素	83.0

从表 2 可以看出,本文算法的 mAP 指标相比传统 SSD 算法提高了 8.5%,并且检测物体越小,检测精度提升得越高,比 DSOD 算法的 mAP 提高 1%,比 YOLOv3 算法提高 2%,比 Faster RCNN 算法提高 3.9%。本文算法比 DSSD 算法的 mAP 低 0.8% 的原因是 DSSD 的基础网络是 ResNet-101,该网络比 VGG-16 网络深、性能好,但是结构也更为复杂,大幅降低了检测速率,不适用于对检测速度要求较高的检测场景,从表 3 中也验证了这一结论。

由于本文算法在进行特征融合时既不需要进行降维操作也不需要调整图片尺寸,同时加入了一种基于残差模块的检测单元,可避免增加网络模型容量以及运算复杂度,因此本文算法的FPS相比于传统SSD的85 frame/s仅损耗了2 frame/s,可达到83 frame/s,远高于DSSD、DSOD和Faster RCNN算法,同时也高于YOLOv3的51 frame/s。

3.4 地铁安检图像的检测结果

地铁安检图像中的一些小物体(如指甲刀、打火

机等)往往只占据检测图片中的很小一部分,同时有可能会被别的物体覆盖,增加了检测的难度。本文收集了约550张的地铁安检图像,其中,包含的检测目标数量基本均衡,对所有图像采用多目标标记,然后随机选取总数的30%作为测试集,其余的分为训练集和验证集。分别用传统的SSD算法和本文算法进行目标检测实验,图5为部分的检测结果,从中可以看出本文算法对检测结果的改进。表4为测试各类目标的mAP指标。

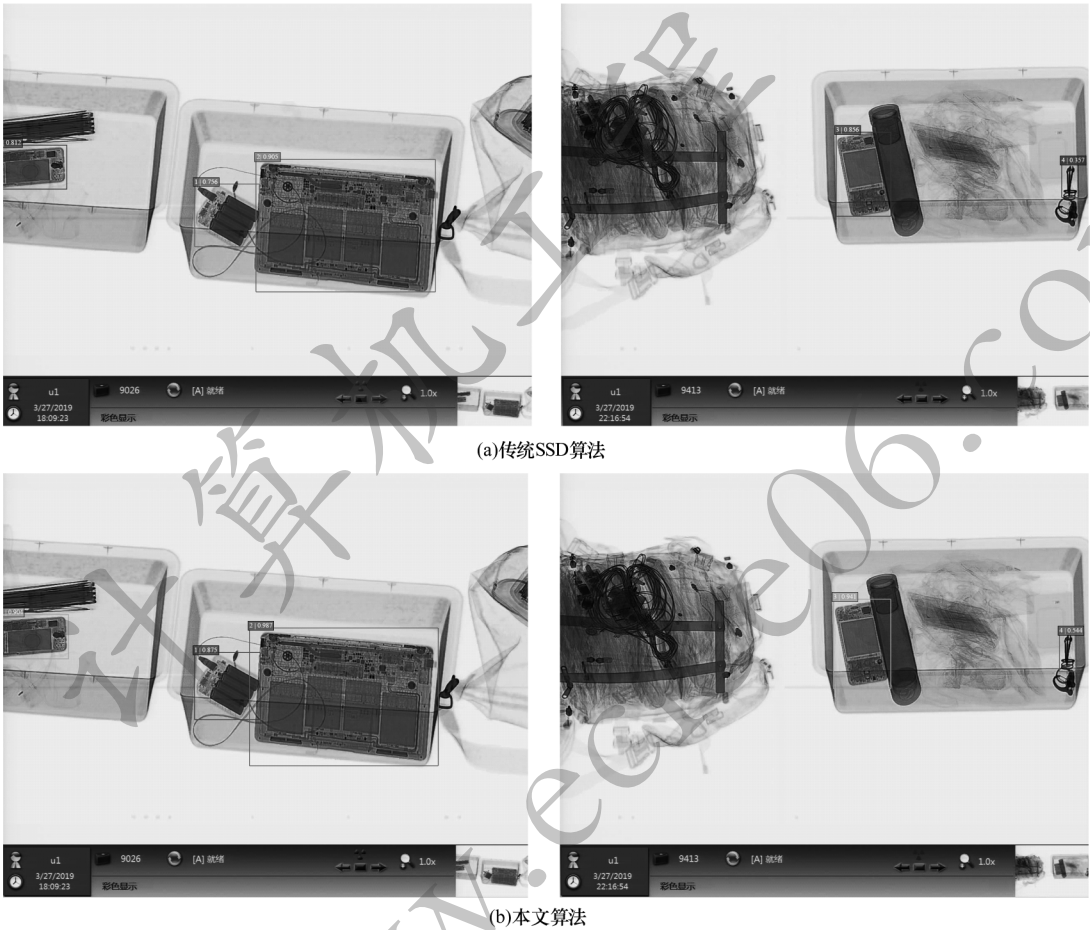


图5 传统SSD算法与本文算法的检测结果对比

Fig.5 Comparison of detection results of the proposed algorithm and traditional SSD algorithm

表4 传统SSD算法与本文算法的mAP指标对比

Table 4 Comparison of mAP indicators of traditional SSD algorithm and the proposed algorithm

算法	mAP	power bank	computer	mobile phone	scissors	lighter
SSD算法	0.655	0.743	0.906	0.825	0.331	0.472
本文算法	0.778	0.861	0.987	0.916	0.524	0.603

从图5的检测结果图和表4中的检测结果可以看出,本文算法较传统SSD算法检测更准确,在对小目标(如scissors)上检测精度提升更为明显,mAP提升了12.3%。

4 结束语

本文对SSD算法中各尺度特征进行尺寸大小不变的卷积操作,将卷积前后对应的特征进行轻量级网络融合,进而生成新的金字塔特征层,最后通过加入基于残差模块的检测单元,提升检测精确度。实验结果表明,在用于地铁安检图像检测时,该算法对于小目标的检测精度较传统SSD算法提升明显,并且能够保证检测速率,检测速率达到83 frame/s。后续将对本文算法的网络结构进行优化,进一步加快检测速度,并且尝试将算法应用于其他特定场景中。



## 参考文献

- [1] 吕苗苗,孙建明. 基于改进高斯混合模型的运动图像目标检测算法[J]. 半导体光电, 2019, 40(6): 874-878, 885.
- [2] 曲欣欣. 机场安检人员的可靠性研究[D]. 沈阳: 沈阳航空工业学院, 2009.
- [3] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]//Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2014: 580-587.
- [4] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection [C]//Proceedings of 2016 IEEE International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2016: 779-788.
- [5] REDMON J, FARHADI A. YOLO9000: better, faster, stronger [C]//Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2017: 6517-6525.
- [6] GIRSHICK R. Fast R-CNN [C]//Proceedings of 2015 IEEE International Conference on Computer Vision. Washington D. C., USA: IEEE Press, 2015: 1440-1448.
- [7] REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks [C]//Proceedings of the 28th International Conference on Neural Information Processing Systems. Cambridge, USA: MIT Press, 2015: 91-99.
- [8] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector [C]//Proceedings of European Conference on Computer Vision. Aire-la-Ville, Switzerland: Eurographics Association Press, 2016: 21-37.
- [9] FU C Y, LIU W, RANGA A, et al. DSSD: deconvolutional single shot detector [EB/OL]. (2017-01-23) [2020-07-10]. <http://cn.arxiv.org/abs/1701.06659>.
- [10] 李晖晖,周康鹏,韩太初. 基于CReLU和FPN改进的SSD舰船目标检测[J]. 仪器仪表学报, 2020, 41(4): 183-190.
- [11] 李善军,胡定一,高淑敏,等. 基于改进SSD的柑橘实时分类检测[J]. 农业工程学报, 2019, 35(24): 307-313.
- [12] 刘慧,张礼帅,沈跃,等. 基于改进SSD的果园行人实时检测方法[J]. 农业机械学报, 2019, 50(4): 29-35, 101.
- [13] LIU H, ZHANG L S, SHEN Y, et al. Real time pedestrian detection in orchard based on improved SSD [J]. Journal of Agricultural Machinery, 2019, 50(4): 29-35, 101. (in Chinese)
- [14] CHEN C Y, LIU M Y, TUZEL O, et al. R-CNN for small object detection [C]//Proceedings of Asian Conference on Computer Vision. Berlin, Germany: Springer, 2016: 214-230.
- [15] BELL S, LAWRENCE Z C, BALA K, et al. Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks [C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2016: 2874-2883.
- [16] KONG T, YAO A B, CHEN Y R, et al. HyperNet: towards accurate region proposal generation and joint object detection [C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2016: 845-853.
- [17] CAO G M, XIE X M, YANG W Z, et al. Feature-fused SSD: fast detection for small objects [EB/OL]. (2018-11-27) [2020-07-10]. <https://arxiv.org/ftp/arxiv/papers/1709/1709.05054.pdf>.
- [18] LI Z X, ZHOU F Q. FSSD: feature fusion single shot multibox detector [EB/OL]. (2018-05-17) [2020-07-10]. <https://arxiv.org/abs/1712.00960>.
- [19] JEONG J, PARK H, KWAK N. Enhancement of SSD by concatenating feature maps for object detection [EB/OL]. (2017-05-26) [2020-07-10]. <https://arxiv.org/abs/1705.09587>.
- [20] 谭红臣,李淑华,刘彬,等. 特征增强的SSD算法及其在目标检测中的应用[J]. 计算机辅助设计与图形学学报, 2019, 31(4): 573-579.
- [21] TAN H C, LI S H, LIU B, et al. Feature enhanced SSD algorithm and its application in target detection [J]. Journal of Computer Aided Design and Graphics, 2019, 31(4): 573-579. (in Chinese)
- [22] LIU W, RABINOVICH A, BERG A C. ParseNet: looking wider to see better [EB/OL]. (2015-11-19) [2020-07-10]. <https://arxiv.org/pdf/1506.04579.pdf>.
- [23] 孙皓泽,常天庆,张雷,等. 基于轻量级网络的装甲目标快速检测[J]. 计算机辅助设计与图形学学报, 2019, 31(7): 1010-1121.
- [24] SUN H Z, CHANG T Q, ZHANG L, et al. Fast detection of armored target based on lightweight network [J]. Journal of Computer Aided Design and Graphics, 2019, 31(7): 1010-1121. (in Chinese)