文章编号：2096-1472(2021)-06-32-05
DOI:10.19644/j.cnki.issn2096-1472.2021.06.008

基于深度学习的交通图像识别的研究与应用

章康明，曹 新

(大连东软信息学院，辽宁 大连 116023)
✉zhangkangming16@dnui.edu.cn; caoxin@neusoft.edu.cn

摘 要：为了克服目标检测算法在交通图像识别领域对数据集利用不充分、对小物体检测不敏感等问题，提出了一种基于SSD算法改进的检测模型。选择自动驾驶领域最重要的测试集作为模型训练的数据集，通过对比实验，选择出训练集、验证集和测试集最合适的划分比例。实验结果显示，合理的数据集划分相较于其他的对照组对于检测目标的准确率提升了13%，检测时间缩短了15%，证明合理的数据集划分能够提升模型泛化能力和检测效率。针对该算法对于小物体检测不敏感这一问题，有针对性地调整了模型的结构及参数，并修改了模型输入图像的尺寸。实验结果表明，在输入相同图片尺寸下，模型对于小物体的检测能力显著提升，整体检测能力提升了14.5%，且保证了较高的检测速率。以上均证明新算法的有效性。

关键词：深度学习；计算机视觉；目标检测；SSD
中图分类号：TP311.5 **文献标识码**：A

Research and Application of Traffic Image Recognition based on Deep Learning

ZHANG Kangming, CAO Xin

(*Dalian Neusoft University of Information, Dalian* 116023, *China*)
✉zhangkangming16@dnui.edu.cn; caoxin@neusoft.edu.cn

Abstract: Aiming at problems of insufficient utilization of datasets and insensitivity to small object detection in the field of traffic image recognition by target detection algorithm, this paper proposes an improved detection model based on SSD (Solid State Disk) algorithm. The most important test set in the field of autonomous driving is selected as the dataset for model training. Through comparative experiments, the most appropriate division ratio of training set, validation set and test set is selected. Experimental results show that compared with other control groups, reasonable dataset division has an increase of 13% in accuracy of detecting targets and a decrease of 15% in detection time, which proves that reasonable dataset division can improve model generalization and detection efficiency. Aiming at the problem that the algorithm is not sensitive to small object detection, structure and parameters of the model is adjusted and size of the input image of the model is modified. The experimental results show that under the same image input size, the detection capability of small objects is significantly improved, the overall detection capability is improved by 14.5%, and a higher detection rate is guaranteed. The above all prove the effectiveness of the new algorithm.

Keywords: deep learning; computer vision; target detection; SSD

1 引言(Introduction)

近年来，随着人工智能的蓬勃发展，深度学习在越多越多场景下应用，尤其是自动驾驶领域。根据IDC最新发布的《全球自动驾驶汽车预测报告(2020—2024)》数据显示，2024年全球L1—L5级自动驾驶汽车出货量预计将达到约5,425 万辆，年均复合增长率达18.3%。在巨大的市场需求推动下，对自动驾驶技术的要求愈加严苛。驾驶环境复杂多变，对驾驶场景中的目标检测算法模型的泛化能力有很高的要求，同时

基金项目：2018年度辽宁省自然科学基金项目《基于数据融合和V2X减速预警模型的研究》.

也需要保证模型的检测速率。

目标检测是在图片中对可变数量的目标进行查找和分类，主要存在目标种类与数量问题、目标尺度问题以及外在环境干扰问题。目标检测算法经过长时间的发展迭代，经过从传统的目标检测方法到深度学习方法的变迁。传统的目标检测算法主要基于传统手工设计特征并结合滑动窗口的方式来进行目标检测和定位，典型的代表有Viola-Jones[1]、HOG[2]、DPM[3]等。传统目标检测算法设计出的特征鲁棒性较差，效率较低，且通过滑动窗口提取特征的方式流程烦琐。因此在2008年DPM算法提出后，传统目标检测算法遇到了较大的瓶颈。

自2012年卷积神经网络的兴起，基于深度学习的目标检测方法发展并成熟，在检测效率和精度上有了巨大的飞跃，逐渐取代传统机器视觉方法，成为目标检测领域的主流算法。目前，基于深度学习的目标检测算法分为One-Stage[4]和Two-Stage[5]。以R-CNN[6]、Faster R-CNN[7]为代表的Two-Stage检测算法具有良好的检测精度，但检测速度相对较慢，无法满足自动驾驶领域实时性需求。One-Stage采用直接回归目标位置的方法，以YOLO[8]、SSD[9]为代表，在保证检测精度的同时，提高了检测速度，但不同模型也有各自的缺陷。

本文提出了一种基于深度学习的交通图像识别算法，以SSD为目标检测模型，使用合适的数据集划分对模型进行训练，并针对模型对小目标检测性能的不足，修改模型相应的输入尺寸，在保证多数目标检测能力的同时，大大改善了模型对小目标的检测能力，辅以快速的图片检测能力，能够提升汽车行驶过程中的安全性，对于深度学习在交通图片识别以及自动驾驶应用上具有参考意义。

2 SSD模型(SSD model)

2.1 模型概述

SSD是一种One-Stage的目标检测模型，移除了region proposals[10]步骤以及后续的像素采样的步骤；借鉴了YOLO的回归思想以及Faster R-CNN的anchor机制，精度可以和Faster R-CNN匹敌，速度上远远快于Faster R-CNN。回归思想的引入降低了模型复杂度，提高了算法的检测速度；anchor机制能够检测不同尺度的目标，提高算法检测精度。

SSD网络模型如图1所示，由主干网络和多尺度feature map预测两部分组成。主干网络由VGG-16组成，舍弃了FC6和FC7两个全连接层，用于特征提取。同时在网络后面添加八个卷积层作为多尺度feature map预测，这些卷积层在尺寸上逐渐减小，在多个尺度上对检测结果进行预测。用于预测检测的卷积模型对于每个特征层都是不同的。基于前馈卷积网络，产生固定大小的边界框集合，并对这些边界框中存在的目标类别实例进行评分，通过非极大抑制来控制噪声，确保网络保留最有效地几个预测，并产生最终的检测结果。

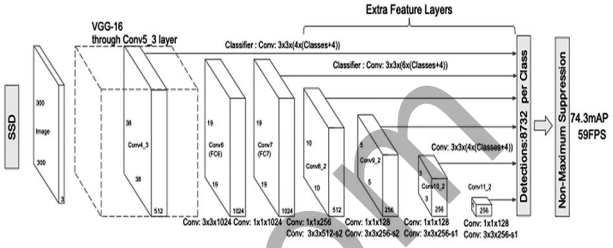

图1 SSD模型

Fig.1 SSD models

2.2 基本原理

2.2.1 用于检测的多尺度映射

多尺度特征映射主要为了提高检测精度。基础网络后添加的一些卷积层主要用于检测和分类。不同的特征层(Feature Layer)预测的边界框(Bounding Box)是不一样的，因此不同的特征层上的卷积模型也不一样。多尺度的特征映射可以检测不同尺度和大小的目标。

2.2.2 卷积预测器

每一个添加的特征层用于一组卷积滤波产生固定的检测预测集合。卷积预测器对这些默认框(Default Box)进行关于类别和位置的回归，然后得出一个类别的得分及坐标偏移量。

2.2.3 默认边界框与长宽比

本文一系列默认的边界框和各个不同的特征层联系在一起，即每个被选中预测的特征层，在每个位置都关联 k 个默认框。在每个特征层的映射中，对于给定 k 个边界框中的每一个，本文计算 c 个类别分数以及相对原始默认边框的四个偏移量，并允许不同的默认边界框，有效地离散出可能的输出框。

2.2.4 NMS(非最大值抑制)

为了避免重复预测，过滤掉背景和得分不是很高的框，从而得到最终预测。

2.2.5 匹配策略

在训练过程中需要确定默认框和真值边界框(Ground Truth Box)之间的联系后训练网络。对于每一个真值边界

框，本文计算其与默认边界框之间的杰卡德系数(Jaccard Overlap)，也就是IOU，默认两者之间的阈值大于0.5，即为正样本。因此本文简化了学习过程，允许网络为多个重叠的默认边框预测高分，而不只是挑选一个边界框。

2.2.6 损失函数

总体损失函数(Confident Loss)是定位损失和置信度损失(Localization Loss)的加权和，公式如式(1)所示：

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

其中，N 是匹配的默认边界框数量，如果 $N=0$，则 $Loss=0$；a 为定位损失与分类损失之间的比重，通过交叉验证方式设置为1。

定位损失是预测框与真实框参数之间的$\mathrm{Smooth}_{\mathrm{L1}}$损失，与Faster R-CNN类似，回归默认边界框 d 的中心偏移量 (cx,cy) 与其宽度 w、高度 h 的偏移量，公式如式(2)所示：

$$\begin{aligned}
&L_{loc}=\sum_{i\in Pos,m\in\{cs,cy,w,h\}}^{N}\sum x_{ij}^{k}\mathrm{Smooth}_{\mathrm{L1}}(l_i^m-g_j^m)\\
&g_j^{cx}=\left(g_j^{cx}-d_j^{cx}\right)/d_i^w\\
&g_j^{cy}=\left(g_j^{cy}-d_j^{cy}\right)/d_i^h\\
&g_j^w=\log\left(\frac{g_j^w}{g_i^w}\right)\\
&g_j^h=\log\left(\frac{g_j^h}{g_i^h}\right)
\end{aligned} \tag{2}$$

其中，l 代表预测边界框与默认框之间的变换关系，g 代表真值边界框与默认边界框之间的变换关系。

置信损失是在多类别置信度上的softmax损失，公式如式(3)所示：

$$L_{conf}(x,c)=-\sum_{i\in Pos}^{N}x_{ij}^{p}\log(c_i^p)-\sum_{i\in Neg}\log(c_i^0) \tag{3}$$

其中，$x_{ij}^p=\{0,1\}$ 代表第 i 个默认边界框与类别 p 的第 j 个真值边界框是否匹配，匹配为1，否则为0。

2.3 网络结构

如表1所示，VGG作为模型的基础网络，用于模型的特征检测。将VGG-16中的FC7改为卷积层Conv7，同时增加了Conv8、Conv9、Conv10、Conv11几个特征层，用于在多个尺度上进行目标检测，以提高检测精度。低层特征代表局部细节特征，高层特征代表全局特征结构，相关结构如表2所示，从多个角度检测目标，提升检测效果。特征层从低到高，感受野由小到大，能够更好地检测原图中不同大小的目标。

表1 VGG-16部分结构

Tab.1 VGG-16 partical structure

类型	过滤器	尺寸	输出尺寸
Convlutional+Relu	64	3×3	224×224
Convlutional+Relu	64	3×3	224×224
Max-Pool	64	2×2	112×112
Convlutional+Relu	128	3×3	224×224
Convlutional+Relu	128	3×3	224×224
Max-Pool	128	2×2	56×56
Convlutional+Relu	256	3×3	56×56
Convlutional+Relu	256	3×3	56×56
Convlutional+Relu	256	3×3	56×56
Max-Pool	256	2×2	28×28
Convlutional+Relu	512	3×3	56×56
Convlutional+Relu	512	3×3	56×56
Convlutional+Relu	512	3×3	56×56
Max-Pool	512	2×2	14×14
Convlutional+Relu	512	3×3	14×14
Convlutional+Relu	512	3×3	14×14
Convlutional+Relu	512	3×3	14×14
Max-Pool	512	2×2	7×7

表2 SSD 300×300添加的特征层

Tab.2 SSD 300×300 added feature layer

层	卷积接收尺寸	输出尺寸
Conv4_3	92×92	38×38
Conv7	260×260	19×19
Conv8_2	292×292	10×10
Conv9_2	356×356	5×5
Conv10_2	485×485	3×3
Conv11_2	612×612	1×1

3 数据准备及算法概述(Dataset preparation and algorithms overview)

3.1 数据集选取

训练选用KITTI[11]数据集，由德国卡尔斯鲁厄理工学院和丰田美国技术研究院联合创办，是目前自动驾驶领域最重要的测试集之一。KITTI数据集针对不同的用途，实验选取Object类型中“2D Object Detection”的数据集，主要是为了验证无人驾驶中有关目标检测算法而设置的数据集。该数据

集一共包含7,841 张训练图和7,518 张测试图，包含80,256 个目标Label。所有图像均为彩色并保存为png格式。

3.2 数据集内容

KITTI数据集为摄像机视野内运动物体提供一个3D边框标注(使用激光雷达的坐标系)。该数据集的标注一共分为八个类别：Car、Van、Truck、Pedestrian、Person、Cyclist、Tram和Misc或Don't Care。其中Don't Care标签表示该区域没有被标注，在训练中可以被忽略。

3.3 算法概述

3.3.1 合适的数据集划分

在数据集有限的条件下，高效利用训练数据集能够训练出一个更优越的算法。将数据集按照训练集、验证集以及测试集进行划分，比例分别为30∶20∶50、40∶10∶50、45∶5∶50三种；并使用相同的SSD网络进行训练，初始学习率为0.001，动量为0.9，权重衰减为0.0005，批处理数据大小为8，训练次数均为80,000 次。对训练结果进行评估，选择最佳的数据集划分。

3.3.2 优化模型

对模型进行评估，发现模型对于较小目标检测性能较差。因为KITTI原始数据集大小为375×1242，而原始模型输入大小为300×300，对输入图片的压缩导致检测效果下降，因此将模型输入的尺寸大小修改为384×1280，以提升模型的检测精度。

4 实验与结果分析(Experiments and results analysis)

4.1 实验环境

本文实验所用的环境如下：

硬件环境：Intel(R) Core(TM) i5-6300HQ CPU @ 2.30 GHz、NVIDIA GTM965M。

软件环境：Ubuntu 18.04 LTS、CUDA 10.0 cuDNN 7.4.2、TensorFlow 1.14.0-gpu、Python 3.6.9、conda 4.9.2。

4.2 评价标准

4.2.1 FPS

*FPS*表示模型每秒检测的图片数量。采用*FPS*作为评价该模型检测速度的唯一指标。*FPS*计算公式如式(4)所示：

$$FPS=\frac{N}{T} \tag{4}$$

其中，N表示检测图片总数量，T为检测所耗的总时间。

4.2.2 准确率和召回率

TP：正确划分为正例的个数，即实际为正例且被分类器划分为正例的实例数。

FN：错误划分为正例的个数，即实际为负例但被分类器划分为正例的实例数。

FP：被错误划分为负例的个数，即实际为正例但被分类器划分为负例的实例数。

TN：被正确划分为负例的个数，即实际为负例且被分类器划分为负例的实例数。

TP、*FP*、*FN*、*TN*可用来计算表示目标检测模型精度的多个指标。

准确率(*Precision*)表示检测到的正确目标占所有检测到的目标的比例。准确率用于评价模型的检测精度，计算公式如式(5)所示：

$$Precision=\frac{TP}{TP+FP} \tag{5}$$

召回率(*Recall*)表示检测到的正确目标占所有真实目标的比例。召回率用于评价模型的检测精度，计算公式如式(6)所示。

$$Recall=\frac{TP}{TP+FN} \tag{6}$$

4.2.3 AP

AP(Average Precision)用于评价模型的平均精准度，以召回率为横轴、准确率为纵轴可以画出一条P-R曲线，*AP*即对P-R曲线求平均值，计算公式如式(7)所示。

$$AP=\int_0^1 p(r)\mathrm{d}r \tag{7}$$

4.2.4 mAP

*AP*的计算正对某一特定类别。*mAP*(mean Average Precision)表示各类别*AP*的算数平均值，计算公式如式(8)所示。

$$mAP=\frac{\sum_{i=1}^{K}AP_i}{K} \tag{8}$$

其中，K表示类别数量。

本文中的实验使用*mAP*评价模型的精度。

4.3 实验结果

4.3.1 改进数据集的划分比例

按照3.3.1中的算法描述对模型进行训练，初始学习率为0.001，动量为0.9，权重衰减为0.0005，每次批处理数据大小值设置为8，训练次数均为80,000 次。训练过程中部分loss

图像得到结果如表3所示。

表3 不同数据集划分比例的训练结果

Tab.3 Training results of division ratios for different dataset

数据类型	数据集划分比例 45∶5∶50	数据集划分比例 40∶10∶50	数据集划分比例 30∶20∶50
Car[1]	0.6611	0.6761	0.6598
Van[2]	0.5111	0.5626	0.5607
Truck[3]	0.3473	0.4950	0.5409
Cyclist[4]	0.1365	0.2195	0.3061
Pedestrian[5]	0.1180	0.0915	0.3747
Person_sitting[6]	0.0911	0.2338	0.1485
Tram[7]	0.6227	0.6012	0.3726
Misc[8]	0.4126	0.3896	0.3079
mAP_VOC07	0.3626	0.4086	0.4015
FPS(fps)	78.43	71.43	72.07

实验结果表明，按照40∶10∶50划分的数据集在目标检测上拥有更高的检测精度。后续对模型改进的实验中，所有数据集划分按照40∶10∶50的标准进行。

4.3.2 模型改进前后检测精度和检测速度的实验对比

将输入模型的图像大小修改为384×1280后，按照40∶10∶50的数据集划分，其余参数均保持一致，得到结果如表4所示。

表4 改进后模型训练结果

Tab.4 Training results of the improved model

数据类型	图像输入尺寸 300×300	图像输入尺寸 384×1280
Car[1]	0.6754	0.6942
Van[2]	0.5698	0.5779
Truck[3]	0.4872	0.5496
Cyclist[4]	0.2311	0.4179
Pedestrian[5]	0.0909	0.3709
Person_sitting[6]	0.1515	0.4881
Tram[7]	0.6216	0.3410
Misc[8]	0.4890	0.3687
mAP_VOC07	0.4145	0.4760
FPS(fps)	78.43	26.76

4.4 结果分析

根据4.3.1的实验结果所示，充分利用数据集对模型进行训练，能够显著提升模型的泛化能力。实验结果显示，合理的数据集划分，相对于其他对照组对于检测目标的*mAP*提升了13%，检测时间缩短10%，证明本文提出的数据集划分能够有效提升模型的泛化能力和检测速率，且在几个重要的检测目标上具有较高的*AP*，能够胜任一般交通场景的目标检测需求。

本文使用的KITTI数据集分辨率较高，而SSD本身对于小物体识别的准确率并不高，为了解决模型对小物体检测不敏感这一问题，本文对模型进行修改，将模型的输入尺寸提升为384×1280。根据4.3.2的实验结果所示，Cyclist类别的*AP*从0.2311提升到了0.4179，Pedestrian类别的*AP*从0.0909提升到了0.3709，Car类别的*AP*从0.6754提升到0.6942等，整体的*mAP*提升了接近14%，证明使用本文提出的算法对小物体的检测能力提升显著，同时还微弱提升了其余物体的检测能力，均证明了本文提出新算法的有效性。模型的部分检测过程和检测结果如图2和图3所示，本文实现的算法模型通过使用分辨率较大的KITTI数据集更真实地模拟现实场景，同时通过对模型的优化，在提升对小物体检测精度的情况下，依旧保持较佳的检测速率，大大提升了模型的整体检测效率。

图2 模型检测结果1

Fig.2 Model testing result 1

图3 模型检测结果2

Fig.3 Model testing result 2

5 结论(Conclusion)

为了进一步模拟真实驾驶场景下目标检测的性能，本文提出一种基于SSD的深度学习交通检测识别算法。实验结果显示，利用本文提出的算法检测准确率和检测效率明显优于未经过本文算法优化的检测算法。在KITTI数据集上的*mAP*达到0.47，*FPS*为26.76 fps，不影响其余检测类型的前提下，提升了模型对小物体的检测精度，同时保证了模型的

(下转第27页)