计算机科学
COMPUTER SCIENCE

基于YOLOv3与改进VGGNet的车辆多标签实时识别算法

顾曦龙, 宫宁生, 胡乾生

引用本文

顾曦龙, 宫宁生, 胡乾生.基于YOLOv3与改进VGGNet的车辆多标签实时识别算法[J]. 计算机科学, 2022, 49(11A): 210600142-7.

GU Xi-long, GONG Ning-sheng, HU Qian-sheng. Multi-label Vehicle Real-time Recognition Algorithm Based on YOLOv3 and Improved VGGNet [J]. Computer Science, 2022, 49(11A): 210600142-7.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210600142

基于 YOLOv3 与改进 VGGNet 的车辆多标签实时识别算法

顾曦龙　宫宁生　胡乾生
南京工业大学计算机科学与技术学院　南京 211816
(781537596@qq.com)

摘　要　为了能快速、有效地识别视频中的车辆信息,文中结合 YOLOv3 算法和 CNN 算法的优点,设计了一种能实时识别车辆多标签信息的算法。首先,利用具有较高识别速度和准确率的 YOLOv3 实现对视频流中车辆的实时监测和定位。在获得车辆的位置信息后,再将车辆信息传入经过简化与优化的类 VGGNet 多标签分类网络中,对车辆进行多标签标识。最后将标签信息输出至视频流,得到对视频中车辆的实时多标签识别。文中训练与测试数据集来源为 KITTI 数据集和通过 Bing Image Search API 获取的多标签数据集。实验结果证明,所提方法在 KITTI 数据集上的 mAP 达到了 91.27,多标签平均准确率达到 80%以上,视频帧率达到 35 fps,在保证实时性的基础上取得了较好的车辆识别和多标签分类效果。

关键词:计算机视觉;车辆识别;多标签识别;目标检测;深度学习
中图法分类号　TP183;TP391.4

Multi-label Vehicle Real-time Recognition Algorithm Based on YOLOv3 and Improved VGGNet

GU Xi-long, GONG Ning-sheng and HU Qian-sheng
College of Computer Science and Technology, Nanjing Tech University, Nanjing 211816, China

Abstract　In order to quickly and effectively identify vehicle information in video, this paper combines the advantages of YOLOv3 algorithm and CNN algorithm to design an algorithm that can identify vehicle multi-label information in real time. Firstly, the high recognition speed and accuracy of YOLOv3 are used to realize real-time monitoring and positioning of vehicles in video stream. After obtaining the vehicle location information, the vehicle information is passed into the improved simplified and optimized VGGNet multi-label classification network to identify the vehicle with multiple tags. Finally, the label information is output to the video stream to obtain real-time multi-label recognition of vehicles in video. The training and test data sets in this paper are derived from KITTI data sets and multi-label data sets obtained through Bing Image Search API. Experimental results show that the mAP of the proposed method on KITTI data set reaches 91.27, the average accuracy of multi-label is more than 80%, and the frame rate of video reaches 35fps. It achieves good results in vehicle identification and multi-label classification on the basis of ensuring real-time performance.

Keywords　Computer vision, Vehicle recognition, Multi-label recognition, Target detection, Deep learning

1　引言

近年来,随着交通监控信息系统的日益普及,基于视频的车辆识别技术发展迅猛,已经成为智能交通领域的研究热点[1]。在实际应用中,车辆识别仍然有诸多困难,如光照、噪声等因素的影响,摄像头摆放位置不当,存在大量近似车型。为此,很多学者展开了研究,部分成果已经逐步得到应用。根据所采用的技术不同,车辆识别技术的发展主要可以分为两个阶段:基于浅层学习的阶段和基于深度学习的阶段。基于浅层学习的车辆识别技术一般先通过人工提取特征,再设计分类器进行识别。Bake 等[2]在 HSV 颜色空间中使用 H 和 S 两个分量的颜色直方图构成二维特征向量,解决了车辆颜色特征的表达问题;Li 等[3]用图像中的目标作为特征,采用空间金字塔技术引入空间信息来实现对特征的表达;Buch 等[4]利用 3D 模型提取运动轮廓,并与投影的模型轮廓进行比较来识别车辆的位置和类别,很好地解决了车辆阴影对车辆识别的影响。但基于浅层学习的方法的网络结构比较简单,难以剥离出数据的深层特征,识别效果过分依赖于人为的特征提取。因此,当外界因素影响较大时,识别结果往往不理想。基于深度学习的方法通常建立一个用于数据分析和学习的神经网络,从原始数据中逐层提取特征,来提升分类精度和预测准确度。常见的模型包括卷积神经网络(Convolutional Neural Network,CNN)、受限波尔兹曼机(Restricted Boltzmann Machine,RBM)等。Rachmadi 等[5]考虑到传统单支 CNN

基金项目:国家重点基础研究发展计划(973 计划)(2005CB321901);基于高压缩比技术的移动环境执法视频采集与管理系统(ZX16487470001);软件开发环境国家重点实验室开放课题(BUAA-SKLSDE-09KF-03)
This work was supported by the National Key Basic Research and Development Program(973 Program)(2005CB321901), Mobile Environment Law Enforcement Video Acquisition and Man-agement System Based on High Compression Ratio Technology(ZX16487470001) and Open Project of the State Key Laboratory of Software Development Environment(BUAA-SKLSDE-09KF-03).
通信作者:宫宁生(chinahqs@163.com)

网络的局限性，采用2条CNN数据提取2组深度特征，经过组合实现并行网络的同步学习，实现了车辆颜色的良好识别；Krause等[6]提出了一种基于协同分割和对齐生成目标局部标注的细粒度车辆识别方法，先将车辆目标分割出来，再对目标构建最小生成树，生成目标的局部关键区域，最后对这些关键区域提取深度特征以训练SVM分类器；Hu等[7]采用3种传统的手工特征作为深度网络的输入，并利用深度玻尔兹曼机方法(Deep Boltzmann Machine，DBM)可以有效融合特征的优点，来表示特征进行车辆识别。相比浅层学习方法，这些基于深度学习的方法不仅节省了研究人员的精力，而且在分类精度和识别准确度上都有了很大的进步。但这些模型对细小物体的检测和识别能力还是不足，往往存在漏检的问题；并且对于挖掘车辆深层信息，如车型、车色、位置坐标等方面的研究还较少。

为了更好地进行车位识别和深度信息挖掘，本文提出使用YOLOv3网络进行车辆识别。首先利用YOLOv3中的深度残差网络提取图像特征，然后引入锚点机制确定车辆的位置信息，最后通过多尺度预测获得不同尺度的车辆，从而解决前述网络难以检测细小物体的问题。在框定车辆后，将其输入改进后的VGGNet网络，实现对车辆的多标签分类，从而获取车辆的更深层信息。

2 目标检测概况

目前，目标检测的主流算法分为两大类：基于候选区域(Region Proposal)的算法，如Faster R-CNN[8]和R-FCN[9]等；以及基于回归的算法，如YOLO[10]和SSD[11]等。

2014年，Girshick等在CNN的基础上提出了R-CNN[12]。R-CNN先通过Selective Search选取多个候选区域，在进行变形、膨胀、加框等操作后将其送入AlexNet中提取特征，最后将特征图送入支持向量机(SVM)分类器得到分类结果。鉴于其良好的性能，R-CNN称为目标检测领域第一个能真正工业化的算法。但在R-CNN算法中需要输入大小统一的图片，因此在剪裁图片时会引起严重变形，为了解决该问题，He等提出了SPP-Net[13]，在卷积层提取特征后添加空间金字塔池化(Spatial Pyramid Pooling，SPP)来代替对原始图片的切割、变形等操作，使得网络不再依赖于输入图片的大小。之后，Fast R-CNN和Faster R-CNN借鉴SPP的思路，对R-CNN做出了改进。Faster R-CNN使用区域推荐网络(Region Proposal Network，RPN)替代了R-CNN中的Selective Search，在图像上提取多个矩形框(Anchor Box)，再判断这些矩形框中是否存在目标，然后对这些目标进行识别，进一步提升网络性能。虽然Faster R-CNN通过RPN实现了共享的全卷积网络，但感兴趣区域(Region of Interesting，ROI)相关的子网络并未完全共享，因此重复计算全连接层是拖累检测速度的一大原因。针对这个问题，R-FCN将全连接层也改为全卷积层，实现了全网络中所有计算的共享，进一步提高了识别速度。总的来说，R-CNN的提出使得目标检测领域的研究迈进了一大步，但由于基于区域推荐的方法通常是两步式(Two-Stage)算法，因此虽然检测精度高，但检测速度还有待提升。

2016年，Redmon等在CVPR上提出了一种全新的目标检测算法YOLO。值得一提的是，在此之前目标检测问题被认为是分类问题，提出的两步式算法如R-CNN系列也是基于分类思想的，即先产生候选区域再进行目标检测，而YOLO创新性地将目标检测问题看作回归问题，直接预测目标的位置信息和分类信息，然后进行目标检测和分类。YOLO对R-CNN系列算法的高精度、低速度的问题进行了折中，在保证一定检测精度的同时，提升了检测速度，使得实时目标检测成为现实。YOLO虽然平衡了检测速度和检测精度，但难以对小物体进行定位，由此Liu等借鉴Faster R-CNN的Anchor思想提出了SSD算法，并引入金字塔网络获得不同尺度的特征图进行目标识别，识别结果相比YOLO有了进一步的提升。此后，YOLOv2[14]、YOLOv3[15]以及各种基于SSD的改良算法[16-17]也相继被提出，对目标检测算法做了进一步的补充和改进。

3 基于YOLOv3和改进VGGNet的车辆识别算法

3.1 基于YOLOv3的车辆检测

3.1.1 利用残差网络提取车辆特征

神经网络的深度对特征提取和识别效果有着重要影响，但事实上并不是网络越深越好。常规的网络堆叠在网络越来越深时，由于梯度消失的现象会更加明显，网络的训练效果也会越来越差。残差网络ResNet[18]提出的残差结构可以有效地缓解深层网络中梯度消失和梯度爆炸的问题。残差网络的基本单元是残差块(res_block，res_unit)，残差网络由多个残差块堆叠而成。

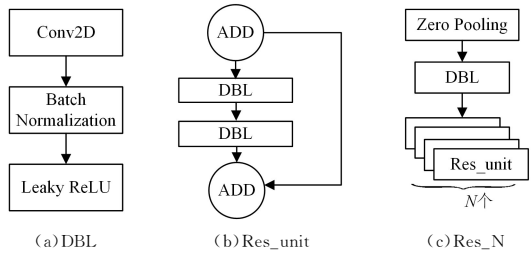


图1 YOLOv3基本组件及残差块结构

Fig. 1 Basic component and residual block structure of YOLOv3

残差块的主要结构分为两个分支：恒等映射和残差分支。残差网络的核心思路是利用旁路分支连接到后面的层，使后面的层可以直接学习残差，减少传统卷积层和全连接层信息损耗或丢失的问题。图1(a)给出了YOLOv3网络中的基本组件，称之为DBL结构，由卷积层、批量标准化(Batch Normalization，BN)和Leaky ReLU函数3部分组成。将该结构与残差块结构组合后得到如图1(b)所示的结构，该残差块包括两个DBL基本组件，输入和输出利用分支直接连接，相邻残差块之间使用ADD操作(Short Connection)连接，具体操作如式(1)所示：

$$y=F(x)+x \tag{1}$$

其中，y为本层输出张量，x为上层输出张量，$F(\cdot)$为本层转化函数，ADD操作将两个张量直接相加，这样保证了残差块的输入和输出张量的维度不变。而由多个残差块就组成了如图1(c)所示的残差块组，该结构先利用一个Zero Padding层改变张量尺寸，再连接一个DBL组件和残差块组(Res_N)。实际上，残差网络首先使用步长为2的卷积层对输入张量进行下采样来缩小图像尺寸，然后经过连续的3×3和1×1卷积层得到输出。

本文叠加了5个残差网络来提取特征，分别用于获取128×128，64×64，32×32，16×16，8×8分辨率下的特征。YOLOv3的深度残差网络结构如表1所列。

表 1　用于提取各分辨率特征的残差网络结构

Table 1　Structure of ResNet for extracting features of each resolution

	类型	滤波器	尺度	输出
	卷积层	32	3×3	256×256
	卷积层	64	3×3/2	128×128
	卷积层	32	1×1	
1×	卷积层	64	3×3	
	残差层			128×128
	卷积层	128	3×3/2	64×64
	卷积层	64	1×1	
2×	卷积层	128	3×3	
	残差层			64×64
	卷积层	256	3×3/2	32×32
	卷积层	128	1×1	
8×	卷积层	256	3×3	
	残差层			32×32
	卷积层	512	3×3/2	16×16
	卷积层	256	1×1	
8×	卷积层	512	3×3	
	残差层			16×16
	卷积层	1024	3×3/2	8×8
	卷积层	512	1×1	
4×	卷积层	1024	3×3	
	残差层			8×8
	平均池化		全局	
	连接数		1000	
	分类器			

3.1.2　使用多尺度的边框回归预测结果

YOLOv3 将图像均分为 $S\times S$ 个网格(Grid)用于预测目标，对于一个目标，若其中心点落在某个网格内，那么这个网格就负责预测这个目标的位置和分类。

YOLO 使用边框回归(Bounding Box Regression)的思路来实现对物体位置的预测。具体来说，就是预先选定先验框(Anchor Box)，然后通过对先验框进行平移与尺度缩放，使之契合物体的实际边框。网格对物体位置的预测就是寻找先验框平移和尺寸缩放的参数。这里涉及到 3 个边框：1)物体的实际边框(Ground Truth)，代表训练值；2)先验框(Anchor Box)，代表预设的固定值；3)预测框(Bounding Box)，代表深度网络训练后得到的预测值。

如图 2 所示，P 代表先验框，G 代表实际边框，$F(P)$代表预测框。衡量预测准确度的一个重要指标是预测边框与实际边框的交并比(Intersection over Union，IoU)，其数学表达式如式(2)所示：

$$IoU_{\text{pred}}^{\text{Truth}}=\frac{G\bigcap F(P)}{G\bigcup F(P)} \tag{2}$$

IoU 的值反映了预测的准确性，因此一般先设定一个 IoU 阈值来判断预测框是否命中目标。因此，边框回归的主要思路就是：对于先验框 P 和实际框 G，找到一个变换 $F(P)$，使得 $F(P)$与 G 的交并比尽量大。

R-CNN 使用简单的平移和尺度缩放来实现先验框到预测框的变换，由于 R-CNN 未对中心坐标做任何约束，使得其可以出现在图像的任意位置，而实际上先验框的设计是为了预测原图的部分区域，不受控制的预测框打破了原有的空间信息，导致训练早期需要很长的时间才能稳定。为了解决这个问题，需要对预测框坐标做一定的约束，使得其始终落在负责预测该物体的网格中。最后采用式(3)计算预测边框的绝对位置。

$$\begin{cases} b_x=\sigma(t_x)+c_x \\ b_y=\sigma(t_y)+c_y \\ b_w=p_w e^{t_w} \\ b_h=p_h e^{t_h} \end{cases} \tag{3}$$

其中，b_x，b_y，b_w，b_h 为预测的绝对位置信息；c_x，c_y 为当前网格左上角到图像网格左上角的距离；p_w，p_h 为先验框的宽高；t_x，t_y，t_w，t_h 为网络训练需要学习的相对位置信息；$\sigma(\cdot)$表示 Sigmoid 函数，目的是将 t_x，t_y 归一化，使得预测的位置信息落于特定的网络中。其具体关系如图 3 所示。

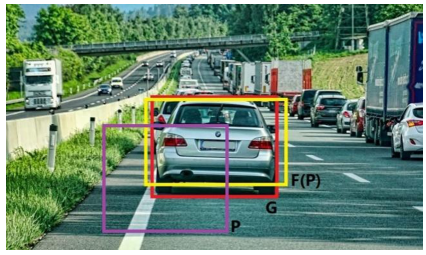


图 2　先验框、预测框与实际边框

Fig. 2　Anchorbox，bounding sox and ground truth

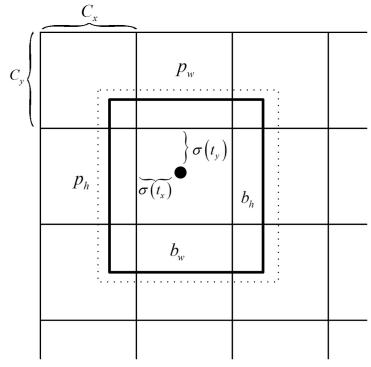


图 3　相对位置预测示意图

Fig. 3　Diagram of relative position prediction

由于预测框是通过某一个先验框变化后得到的，因此先验框的选定能直接影响整个网络的训练速度和效果。而相比手动设置先验框，使用 k-means 聚类算法实现对不同数据集选定不同的先验框能起到更好的效果。与传统 k-means 算法的距离公式不同，YOLOv3 使用 IoU 得分作为判断依据，具体如式(4)所示：

$$d_{\text{centroid}}^{\text{box}}=1-IoU_{\text{centroid}}^{\text{box}} \tag{4}$$

其中，$d_{\text{centroid}}^{\text{box}}$表示预测框到聚类中心点的距离，$IoU_{\text{centroid}}^{\text{box}}$表示预测框与实际边框的 IoU 值。

使用 k-means 聚类能有效地提高先验框的质量，但由于目标的检测是在 $S\times S$ 个网格的基础上进行的，因此在检测细小物体(如远方的车辆)时会有多个物体落在同一个网格中，从而造成漏检。为了弥补这个缺点，可以使用多个尺度的特征图分别检测不同尺寸的物体，降低漏检率。本文在 3 个尺度的特征图下直接回归预测车辆信息，同时在选择先验框时也设置了 3 种不同的尺寸，这意味着一共会产生 9 个先验框，从而避免细小事物被漏检。

3.2　改进的 VGGNet

VGGNet[19]是由牛津大学的计算机视觉组和 Google Deep Mind 公司的研究员一起研发的一种网络结构，其突出的创新点在于提出了多个 3×3 的卷积层叠加能达到单个 5×5 或更大核的卷积层同样的感受野，不仅大幅度地降低了参数数量，而且利用了更多的非线性操作，使得网络深度更大且特征学习能力更强。从网络结构来看，传统的 VGGNet 设计了 5 组卷积，每组卷积包括 2～3 个 3×3 的卷积层，每组卷积层最后都会连接一个最大池化层用于缩小图片尺寸；最后

使用 3 个全连接层和 1 个 softmax 层进行分类。

本文借鉴 VGGNet 的思路，设计了一个简化的 VGGNet 用于对已通过 YOLOv3 网络完成定位的车辆做进一步的多标签识别。本文缩减了 VGGNet 的网络结构，将 5 组卷积缩减至 3 组，将全连接层缩减至 1 层。缩减后的 VGGNet 识别效率得到了大幅提升，而识别率因为输入图像的高质量并不会出现显著降低。

传统 VGGNet 最终的激活函数是 softmax，如式(5)所示，softmax 令输出向量中各个分量的总和为 1，这种形式体现的是各个分类可能性的比重，因此主要应用于单标签识别。本文将最终的激活函数由 softmax 改为 Sigmoid，如式(6)所示。由于 Sigmoid 中输出向量的各个分量没有总和的约束，代表的是各个分类本身的可能性，因此可以应，用于多标签的识别。由于激活函数的更改，损失函数也从分类交叉熵更改为二元交叉熵。

$$\begin{aligned}&\vec{y}_{\text{softmax}}=(t_1,t_2,\cdots,t_{n-1},t_n)\\&\text{s.t.}\sum_i t_i=1,t_i\in(0,1)\end{aligned}\tag{5}$$

$$\vec{y}_{\text{Sigmoid}}=(t_1,t_2,\cdots,t_{n-1},t_n),t_i\in[0,1]\tag{6}$$

最后本文设计图 4 所示的类 VGGNet，用于对车辆进行多标签识别。

如图 4 所示，本文使用的类 VGGNet 结构的输入为 96×96 的 RGB 图像，经过 3 组卷积提取特征。3 组卷积的卷积核大小均为 3×3，卷积个数分别为 1，2，2。每个卷积层后都紧跟着 1 个 ReLU 层和 1 个 BN 层用于规范数据。每组卷积最后都连接 1 个最大池化层和 1 个 Dropout 层[20]，最大池化层用于改变张量尺寸，3 个池化层的步长分别为 3，2 和 2，窗口分别为 3×3，2×2 和 2×2。3 个 Dropout 层用于防止过拟合，其中概率均设置为 0.25。经过 3 组卷积后，使用 1 个 flatten 将输出转为一维张量，然后经过 1 个通道数为 1024 的全连接层后送入最后的 sigmoid 分类器，得到通道数为 N 的最终输出，其中 N 为样本类别数量。

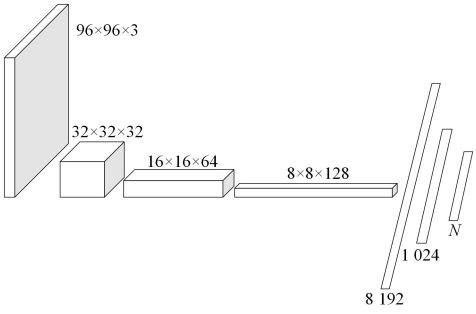


图 4 用于多标签识别的类 VGGNet 结构图

Fig. 4 Diagram of VGGNet-like sructure for multi-label recognition

3.3 帧率优化算法

在实际检测中，考虑到一段视频中的车辆是连续运动的，本文设计了相应算法，用于减少对同一车辆的重复识别，如算法 1 所示。该算法首先保存上一帧中所有的目标信息，包括位置信息、分类信息以及上次多标签分类时的位置信息。对于当前帧的每个目标，首先寻找上一帧中与该目标最接近(IoU 的值最大)的目标，若 IoU 值大于 0.7，则判断为同一目标。对于同一个目标，计算当前帧位置上一次多标签分类时位置信息的 IoU 值，若大于 0.1，则判断该目标尚未进行长距离移动，从而不做重复的多标签分类，沿用上一次的分类结果，减少计算量。若该目标是一个新出现的目标，或距离上次分类经过了长距离的移动，则送入多标签分类网络进行重新分类，并更新信息。

算法 1 Improved Classification Method

Input: the list of location, A; the information of last frame, F_1

Output: the information of current frame, F_2

1. for each i in A do
2. Finding the object O which is nearest i in F1;
3. Getting the location j and last classified location k of O;
4. If IOU(i,j)>0.7 and IOU(i,k)>0.1 then
5. Assigning k to current classified location p;
6. else
7. Multi-classifying i again and assign it to p;
8. end if
9. Adding car information I to F_2;
10. end for
11. Return F_2.

算法 1 中，对于两个 IoU 阈值的设定需要根据实际情况选定，一般来说，对于目标运动速度较快的情况，阈值需要适当减小。

3.4 算法流程

总的算法流程如图 5 所示。首先从输入视频流中获取图像帧，图像帧经过 YOLOv3 中的残差网络提取特征、多尺度特征图检测目标后得到车辆目标的位置信息。这些位置信息用于分割原图像帧，获得对应的目标图像。然后遍历这些目标，通过与上一帧的信息判断这个目标是否在之前短时间内已经进行过多标签分类，若是，则该目标沿用上次的分类结果以减少计算量，否则重新对该目标进行多标签分类。得到该帧所有目标的多标签分类信息和位置信息后，保存该信息并结合原数据帧生成输出帧，最后得到输出视频流。

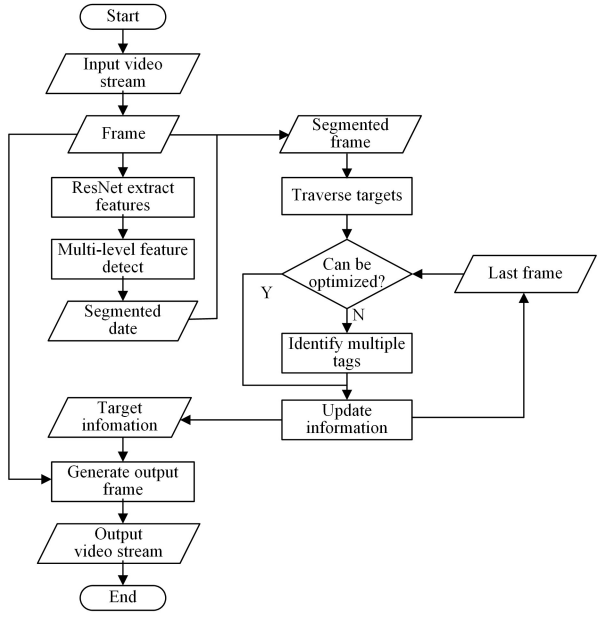


图 5 多标签车辆实时识别算法总流程图

Fig. 5 Flow chart of multi-label vehicle real-time recognition algorithm

4 实验与分析

本文使用公开数据集 KITTI[21] 作为 YOLOv3 的训练集，用于识别车辆并提取视频帧中的车辆位置；然后对 KITTI 中的部分数据进行了人工多标签标识，获得车辆的多标签

数据集训练类 VGGNet 分类网络，用于对提取到的车辆图片进行进一步的多标签识别。实验基于 Keras(后端使用 Tensorflow-GPU)框架编程实现，PC 机的 CPU 为 Intel Core i97900x，GPU 为 NVDIA RTX2080Ti，内存为双通道 16GB×2 DDR4 3000MHz，系统平台为 Windows10 专业版。

4.1 数据集

4.1.1 KITTI 数据集

KITTI 是目前在自动驾驶场景下最大的计算机视觉算法评测数据集。该数据集于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，采用 2 个灰度摄像机、2 个彩色摄像机、1 个 Velodyne 3D 激光雷达、4 个光学镜头以及 1 个 GPS 导航系统对市区、乡村和高速公路等场景进行数据采集。数据范围包括立体图像(stereo)、光流(optical flow)、视觉测距(visual odometry)、3D 物体检测(object detection)和 3D 跟踪(tracking)等多个方面，用于评测计算机视觉技术在车载情况下的性能。整个数据集由 389 对立体图像和光流图，39.2km 视觉测距序列以及超过 200k 3D 标注物体的图像组成，以 10Hz 的频率采样及同步。数据集中每张图片包括最多 15 辆车和 30 个行人，这些目标还存在各种程度的遮挡和截断。数据集的原始数据包括 8 个分类标签，分别为 car，van，truck，pedestrian，pedestrian(sitting)，cyclist，tram 以及 misc。本文保留了车辆识别需要的 3 个标签，即 car，van 和 truck，共 6970 张图像作为 YOLOv3 的训练与测试数据集。其中按 7∶3 的比例将图片集随机分割成训练集和测试集，最后得到每个标签下的实际目标数量，如表 2 所列。

表 2 Car，Van，Truck 标签表示的实际目标数量

Table 2 Actual target number of Car，Van and Truck

Label	Car	Van	Truck
Training set	19923	2098	347
Test set	8819	816	164
Total	28742	2914	511

本文从上述数据集中，人工对 3500 个目标进行了多标签标识，分为 5 种颜色(Black，White，Orange，Blue，Red)、2 种视图(Front，Rear)、3 种车型(Car，Van，Truck)共计 30 种类别，并将目标从原 KITTI 数据集中分割出来，形成新数据集用于训练和测试类 VGGNet 多标签识别网络。

4.1.2 Bing Image Search API

Bing Image Search API 是微软认知服务(Cognitive Services)中的一个接口，微软认知服务主要用于帮助用户在视觉、语言、文本等 AI 应用中提供数据。该接口帮助用户快速获得符合条件的图片集数据，包括图片列表的缩略图、完整的图像 URL、发布网站的信息和图像元数据等。

为了弥补 KITTI 中人工再标识的样本数量不足，本文利用 Bing Image Search API 批量获取各类车辆图片并通过存放于不同文件夹来实现对图片集的快速标识。我们利用脚本获取了 5 种颜色、2 种视图、3 种车型共计 30 种类别，每种类别获取了 300 张，共计 9000 张图像。经过人工筛选掉不符合要求的图片后，共获得了 5328 张图像。图像数据集的最终数量如表 3 所列。表 3 中每项都由两个数字组成，前者代表该车型下 Front 标签的图片数量，后者代表 Rear 标签的图片数量。需要说明的是，由于 Van 标签有着较多的搜索歧义，为了保障最终数据集各个分类的规模接近，因此带有该标签的分类均额外获取了 500 张图片进行筛选。

表 3 用于多标签识别的车辆数据集

Table 3 Vehicle dataset for multi-label recognition

	Car	Van	Truck
Black	178/169	177/176	192/179
White	182/176	173/162	185/171
Orange	192/177	189/170	184/181
Red	170/182	185/174	173/182
Blue	186/171	175/172	178/167

4.2 实验数据与分析

4.2.1 YOLOv3 网络在 KITTI 数据集上的性能

本文首先将筛选得到的 KITTI 训练集中的图片缩放至 416×416 后送入 YOLOv3 网络中，经过 5 组残差网络构成的骨干网络提取特征。然后通过 3 种不同尺寸的特征图预测图像中车辆的位置信息和初步的分类信息(Car，Van，Truck)。这里使用 k-means 聚类方法计算 KITTI 数据集的预选框，具体为：(10，13)，(15，30)，(31，22)，(30，55)，(61，44)，(60，117)，(120，92)，(150，192)，(368，332)。因为本文使用了 KITTI 中的 3 个标签用于 YOLOv3 的训练，并修改网络结构的输出张量尺度为 13×13×24，26×26×24，52×52×24。

针对 KITTI 数据集进行网络结构及参数调整后，本文测试了 YOLOv3 网络和其他常用网络在 KITTI 数据集上的表现，主要参考指标为均值平均精度(Mean Average Precision，mAP)与帧率(fps)，具体结果如表 4 所列。

表 4 不同网络在 KITTI 上的实验结果

Table 4 Results of different networks in KITTI

Network structure	Enter	Frame rate	mAP	Car	Van	Truck
FasterR-CNN (VGG16)	600×—	14.32	76.61	78.34	72.38	79.11
SSD300	300×300	81.10	82.01	82.86	75.12	88.05
SSD512	512×512	50.87	81.42	81.99	74.71	87.56
YOLOv2	416×416	157.95	65.97	66.94	60.87	70.10
YOLOv3	416×416	73.91	91.27	92.12	87.02	94.68
YOLOv3-Tiny	416×416	298.12	64.60	64.40	62.01	67.39

表 5 多种数据集训练的分类网络的性能对比

Table 5 Performance comparison of classification network trained by multiple datasets

Data set	Scale	KITTI accuracy rate/%			Street accuracy/%		
		Colour	View	Model	Colour	View	Model
Image Search API	5328	80.8	70.1	75.4	79.2	71.3	73.5
Part of the KITTI data set	3500	87.1	77.3	82.4	83.4	73.6	76.6
Fusion data set	8828	86.3	77.9	81.0	84.5	76.9	80.1

从实验数据中可以看出，YOLOv3 在 KITTI 数据集上有着很好的识别率和较高的识别速度，相比 Faster R-CNN 和 SSD 算法都更具优势。YOLOv2 和 YOLOv3-Tiny 的骨干网络比 YOLOv3 简单很多，它们的识别速度比 YOLOv3 要快很多，但识别率相对较差。由于 YOLOv3 已经能够在实验环境中满足实时性的要求，因此本文使用 YOLOv3 作为检测车辆

位置的算法。如果在生产环境中没有足够性能的 GPU 支持，那么使用 YOLOv3-Tiny 会是一个值得考虑的选择。

4.2.2 类 VGGNet 在车辆多标签识别上的性能

本文尝试了多种方法来获得多标签标识的数据集，用于训练多标签分类网络并测试其在 KITTI 数据集以及附近道路实拍视频上的性能，包括使用 Bing Image Search API 来获取并制作多标签标识车辆数据集、人工对 KITTI 数据集中的部分数据进行多标签标识，以及将两者相结合构成新数据集，如表 5 所列。

从表中可以看到，在多标签标识中，颜色是相对容易正确分类的，其次是车型，错误率最高的则是车辆视图。此外，由部分 KITTI 数据集训练的分类器在 KITTI 数据集上的表现最佳，在实际街道视频上的表现也要略好于 Image Search API 获取的数据集。这主要是因为 Image Search API 获取的图片有很大一部分属于车辆的特写图片，与街道的实际图片有一定的差距。融合数据集在 KITTI 数据集上的表现也较为不错，在街道实拍视频中的表现则比单 KITTI 数据集要更好，这主要获益于样本数量的提升。

4.2.3 帧率优化算法的性能测试

本文测试了 YOLOv3 检测目标的速度与类 VGGNet 多标签分类的速度。从表 4 可以看出，YOLOv3 在实验环境中基本能满足实时识别视频的要求，而我们对多标签分类网络进行了测试，包括图像缩放等操作在内，识别速度为 11.3fps，显然不能满足实时识别的要求。本文利用视频内车辆的位置变化是连续不突变的特性，通过保存检测目标上一帧的位置信息和上次多标签分类信息来减少重复计算量。帧率优化算法应用前后的对比实验结果如表 6 所列。

表 6 帧率优化前后的整体算法表现

Table 6 Algorithm performance before and after frame rate optimization

	Average classification target per frame	Average frame rate
Before optimization	3.12	4.10
Optimized	0.32	39.98

如表 6 所列，优化前，每帧需要对所有目标进行多标签分类，严重拖累了整体算法的识别速度。使用帧率优化算法后，视频的识别过程中减少了约 90%的多标签分类计算（减少量取决于视频中车辆在窗口中相对位置的移动速度），使得整体算法的帧率维持在 35 帧以上，基本满足了实时识别的要求。

4.2.4 实验效果图

采用 YOLOv3 和改进后的类 VGGNet 结构进行车辆的多标签识别，得到的效果图如图 6 所示。

(a)前一帧识别结果 (b)后一帧识别结果

图 6 总体识别效果图

Fig. 6 Overall recognition results

图 6 为连续的两帧，图中框出的目标为网络识别出来的目标，每个框的左上角标注有识别结果。识别框分为两种，深色框代表该目标在当前帧进行了多标签识别，浅色框则代表该目标沿用了上一帧的分类信息，图像左上角则标注有当前帧进行多标签分类的目标数量。可以看到，经过帧率优化后，每一帧中大部分目标都是不需要进行重复分类的，总体网络速度得到了较大提升。

5 总结与展望

本文利用 YOLOv3 能够端对端快速检出目标位置信息的能力，完成了对视频中车辆的检测，然后设计了一种简化的 VGGNet 网络结构，通过更改最后的激活函数和网络的损失函数使其能够对 YOLOv3 输出的车辆分割图像进行多标签分类。通过对比实验，本文发现 YOLOv3 在 KITTI 数据集上有着优秀的性能，但多标签分类网络的识别速度较慢。针对这一缺陷，本文设计了一种优化算法，减少了网络中 77%～95%的重复多标签分类计算，使整个识别算法的帧率维持在 35fps 以上。在保证基本满足实时性的要求上，做到了 91.27 的 mAP，以及平均 80%以上的多标签分类准确率。

本次实验也存在许多值得改进的地方。首先是数据集的选择，本文使用的 KITTI 数据集虽然是非常优秀的车辆识别数据集，但缺少多标签分类，使得本次实验在进行后续工作时遇到较大困难，后续工作一方面可以继续完成对 KITTI 数据集的多标签标记，另一方面应该积极构建本地街道的多标签数据集。此外，使用类 VGGNet 进行多标签分类的效率还是偏低，今后应该从改进 YOLOv3 网络着手，实现端对端的多标签分类网络结构。

结束语 本文结合 YOLOv3 算法和 CNN 算法的优点，设计了一种能实时识别车辆多标签信息的算法。利用具有较高识别速度和准确率的 YOLOv3 实现对视频流中车辆的实时监测和定位。本文方法在 KITTI 数据集上的 mAP 达到了 91.27，多标签平均准确率达到 80%以上，视频帧率达到 35fps，在保证实时性的基础上取得了较好的车辆识别和多标签分类效果。

参 考 文 献

[1] ZHANG Q, LI J F, ZHUO L. Review of Vehicle Recognition Technology[J]. Journal of Beijing University of Technology, 2018, 44(3): 382-392.

[2] BAEK N, PARK S M, KIM K J, et al. Vehicle color classification based on the support vector machine method[C]// International Conference on Intelligent Computing (ICIC). 2007: 1133-1139.

[3] LI L J, SU H, XING E P, et al. Object bank: a high-level image representation for scene classification & semantic feature sparsification[C]// Advances in Neural Information Processing Systems (ANIPS). 2010: 1378-1386.

[4] BUCH N, ORWELL J, VELASTIN S A. Detection and classification of vehicles for urban traffic scenes[C]// Visual Information Engineering (VIE). 2008: 182-187.

[5] RACHMADI R F, PURNAMA I. Vehicle color recognition using convolutional neural network [J]. arXiv, 2015: 1510.07391.

[6] KRAUSE J, JIN H, YANG J, et al. Fine-grained recognition without part annotations[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015: 5546-5555.

[7] HU A, LI H, ZHANG F, et al. Deep Boltzmann machines based

vehicle recognition[C]// The 26th Chinese Control and Decision Conference(CCDC). 2014:3033-3038.

[8] REN S,HE K,GIRSHICK R,et al. Faster R-CNN:towards real-time object detection with region proposal networks[J]. International Conference on Neural Information Processing Systems,2017,37(6):1137-1149.

[9] DAI J,LI Y,HE K,et al. R-FCN:Object Detection via Region-based Fully Convolutional Networks[C]// Proceedings of the 30th International Conference on Neural Information Processing System(NIPS). 2016:379-387.

[10] REDMON J,DIVVALA S,GIRSHICK R,et al. You Only Look Once:Unified,Real-Time Object Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition(CVPR). 2016:779-788.

[11] LIU W,ANGUELOV D,ERHAN D,et al. SSD:Single Shot MultiBox Detector[C]// European Conference on Computer Vision(ECCV). 2016:21-38.

[12] GIRSHICK R,DONAHUE J,DARRELL T,et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]// IEEE Conference on Computer Vision and Pattern Recognition(CVPR). 2014:580-587.

[13] HE K,ZHANG X,REN S,et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,2014,37(9):1904-16.

[14] REDMON J,FARHADIA. YOLO9000:Better,Faster,Stronger[C]// IEEE Conference on Computer Vision and Pattern Recognition(CVPR). 2017:6517-6525.

[15] REDMON J,FARHADI A. YOLOv3:An Incremental Improvement[J]. arXiv:1804.02767,2018.

[16] ZHANG Z,QIAO S,XIE C,et al. Single-Shot Object Detection with Enriched Semantics[C]// IEEE Conference on Computer Vision and Pattern Recognition(CVPR). 2018:5813-5821.

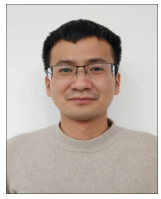
[17] ZHANG S,WEN L,BIAN X,et al. Single-Shot Refinement Neural Network for Object Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition(CVPR). 2018,4203-4212.

[18] HE K,ZHANG X,REN S,et al. Deep Residual Learning for Image Recognition[C]// IEEE Conference on Computer Vision and Pattern Recognition(CVPR). 2016,770-778.

[19] SIMONYAN K,ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. arXiv: 1409.1556,2014.

[20] SRIVASTAVA N,HINTON G,KRIZHEVSKYA,et al. Dropout:A Simple Way to Prevent Neural Networks from Overfitting[J]. Journal of Machine Learning Research,2014,15(1):1929-1958.

[21] GEIGER A,LENZ P,STILLER C,et al. Vision meets robotics:The KITTI dataset[J]. The International Journal of Robotics Research,2013,32(11):1231-1237.



GU Xi-long, born in 1993, postgraduate. His main research interests include deep learning and target detection.

GONG Ning-sheng, born in 1958, Ph.D, professor. His main research interests include mathematical logic, BP neural network, image processing, pattern recognition, data mining and so on.