融合多尺度特征的目标检测模型

刘万军，王凤*，曲海成
辽宁工程技术大学软件学院，辽宁 葫芦岛 125105

摘要 为使 YOLOv2 算法在保证检测速度的同时进一步提高目标检测的精确率，在 YOLOv2 模型的基础上提出 RF-YOLOv2 新模型。该模型先将 KITTI 数据集经过聚类，选出最适合 KITTI 数据集的候选框个数和候选框尺寸；其次在网络结构的训练部分采用残差块结构增加卷积层，提取更符合目标的特征描述；最后在网络结构的检测部分引入特征金字塔网络，将不同尺寸大小的特征图进行融合，使得低层特征图也具有丰富的语义信息。实验结果表明，RF-YOLOv2 模型能获得更深层的特征、能融合更多尺寸的目标信息，改善了目标检测过程中由实际道路场景复杂、目标外形和结构多变等特点导致的检测率不高问题，在保证算法实时性的条件下，提高了对目标检测的精确率，RF-YOLOv2 模型对大目标检测效果更佳。

关键词 图像处理；目标检测；深度学习；卷积神经网络；特征融合；残差网络
中图分类号 TP301.6 **文献标识码** A **doi**：10.3788/LOP56.231007

Object Detection Model Based on Multi-Scale Feature Integration

Liu Wanjun, Wang Feng*, Qu Haicheng
College of Software, Liaoning Technical University, Huludao, Liaoning 125105, *China*

Abstract To ensure detection speed and further improve object detection accuracy, a new model RF-YOLOv2 is proposed on the basis of the YOLOv2 model. In this new model, the KITTI data set is first clustered to select the most suitable number and size of candidate boxes. Next, a residual block structure is used to increase the number of convolutional layers in the training part of the network structure. This increase helps the model to extract more strong features to better describe objects. Finally, a feature pyramid network is introduced in the detection part of the network structure, fusing the feature graphs with different sizes. This network allows even low-level feature graphs to capture rich semantic information. Experimental results show that the RF-YOLOv2 model can gain the deeper information about features and can integrate more object size information. These improvements alleviate significant problems in current models that lead to low detection rates when actual road scenes are complex or when objects vary in shape or structure. The proposed model also improves object detection accuracy in real time detection and achieves better results for large object detection.

Key words image processing; object detection; deep learning; convolutional neural network; feature fusion; residual network
OCIS codes 100.3008; 100.4996; 200.4260

1 引 言

目标检测是计算机视觉领域的研究重点和热点[1-2]，主要任务是给出目标的正确位置和所属类别。其检测图像可以分为多种类型，如红外图像[3]、遥感图像[4]等。目标检测的方法分为两类：一类是传统的目标检测方法；一类是基于卷积神经网络(CNN)的目标检测方法。传统的目标检测方法中具有代表性的算法是 Felzenszwalb 等[5]提出的多尺度形变部件模型(DPM)，该模型把梯度方向直方

收稿日期：2019-05-10；**修回日期**：2019-05-21；**录用日期**：2019-06-03
基金项目：国家自然科学基金青年基金项目(41701479)、辽宁省自然科学基金(20180550529)、第六批生产技术问题创新研究基金(20160092T)

* **E-mail**：838808390@qq.com

图(HOG)[6]特征和支持向量机(SVM)[7]分类器结合,利用两者的优点在图像处理和人脸识别等任务上取得了重要突破。基于 CNN 的目标检测算法分为基于候选区域的目标检测算法和基于回归的目标检测算法。基于候选区域的目标检测算法包括 R-CNN 算法[8]、Fast R-CNN 算法[9]和 Faster R-CNN 算法[10]。其中 R-CNN 算法利用 Selective Search 算法[11]产生类别无关的候选区域,然后利用 CNN 提取目标特征,最后利用 SVM 进行分类;Fast R-CNN 算法在 R-CNN 算法的基础上提出感兴趣区域(ROI)池化层,同时将多个任务的损失函数写在一起,实现单极训练过程;Faster R-CNN 算法在两者基础上抛弃了滑动窗口生成候选区域的策略,提出候选区域网络(RPN)。基于回归方法的目标检测算法有 Redmon 等[12]提出的 YOLO 算法,该算法直接在输出层给出目标正确位置和所属类别,检测速度可以达到每秒 45 帧,有效地实现了端到端的目标检测;在 YOLO 算法之后 Liu 等[13]提出 SSD 算法,该算法使用全图各个位置的多尺度区域特征进行回归,保证速度的同时也提升了精度。同时基于 CCN 的目标识别算法需要依赖大型 GPU 计算平台,计算资源消耗大,难以向嵌入式平台移植,因此目前有许多方法研究如何在嵌入平台上进行目标检测[14]。

与基于回归方法的目标检测算法相比,基于候选区域的目标检测算法在目标位置定位和检测精度方面有比较明显的优势。但通常基于候选区域的算法有 3 个不足:一是这类算法在检测速度远远达不到实时性的需求;二是这类算法的网络模型借鉴了已有的模型如 GoogleNet[15],比 YOLOv2[16]的 Darknet 模型训练的时间长,且参数量大;三是这类算法对硬件设备要求高,而 Darknet 模型实验环境简单而廉价,只需要一个显存大于 4G 的机器就可以进行检测模型训练。目前较为流行的单阶段目标检测算法是 YOLOv2 和 YOLOv3[17],YOLOv3 的优点是对目标检测的精确率较高,但是检测速度不如 YOLOv2。YOLOv2 虽然对目标检测的精确率不如 YOLOv3,但是检测速度能满足实时性的需求。基于此,本文提出融合多尺度的目标检测模型,其主要贡献是在满足实时性的基础上,提高目标检测的精确率。

2 RF-YOLOv2 算法原理

2.1 网络结构

YOLOv2 网络包括 19 个卷积层,5 个最大池化层和 1 个全局平均池化层。在网络中多次使用了 3×3 和 1×1 的卷积核,将 1×1 的卷积核放在 3×3 的卷积核之间,用来压缩特征,加深网络深度,每次池化操作后通道数变为原来的 2 倍,最后得到的模型就是 Darknet-19。由于 YOLOv2 网络的卷积层数量较少,因此提取的特征层次较浅,本文为提取更深层次的特征,首先增加更多的卷积层,并在卷积层之间加入残差结构来避免梯度消失的问题。其次为了提高目标检测的准确率,将不同尺度的信息进行融合,并使用多尺度进行检测,得到 RF-YOLOv2 网络结构,见表 1。

表 1 RF-YOLOv2 网络结构

Table 1 RF-YOLOv2 network structure

Layer block	Type	Number of filters	Size / stride	Output
	Convolutional	32	3×3	416×416
	Maxpool		2×2/2	208×208
	Convolutional	64	3×3	208×208
1×	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			208×208
	Maxpool		2×2/2	104×104
	Convolutional	128	3×3	104×104
2×	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			104×104
	Maxpool		2×2/2	52×52
	Convolutional	256	3×3	52×52
4×	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			52×52
	Maxpool		2×2/2	26×26
	Convolutional	512	3×3	26×26
4×	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			26×26
	Maxpool		2×2/2	13×13
	Convolutional	1024	3×3	13×13
4×	Convolutional	512	1×1	
	Convolutional	1024	3×3	
	Residual			13×13
	Avgpool		Global	3
	Softmax			

2.2 检测过程

在训练过程中不断学习输入图像的目标类别和背景上下文信息,将检测图像划分为 $S\times S$ 个网格,每个网格预测 B 个边框,每个边框要预测检测对象的横坐标、纵坐标、宽度、长度(x,y,w,h)和置信度共 5 个预测参数。置信度是判断边界框是否包含目

标以及目标位置是否正确的指标，通过边界框包含物体的概率和图像交并比（I_{OU}）计算。I_{OU}为边界框和真实物体区域的交集与两者并集之比。置信度和交并比的计算公式可表示为

$$C_{\mathrm{condidence}}=P(O_{\mathrm{object}})\times I_{\mathrm{OU\_boundingbox}}^{\mathrm{groundtruth}},\tag{1}$$

$$I_{\mathrm{OU}}=\frac{a_{\mathrm{area}}(B_{\mathrm{pt}}\cap B_{\mathrm{gt}})}{a_{\mathrm{area}}(B_{\mathrm{pt}}\cup B_{\mathrm{gt}})},\tag{2}$$

式中：O_{object}为目标；$P(O_{\mathrm{object}})$为边界框包含目标物体的概率，如果预测框和真实物体区域重叠度为百分之百，则$P(O_{\mathrm{object}})=1$，置信度为1，如果网格中不存在预测框，预测框和真实物体区域没有重叠部分，则$P(O_{\mathrm{object}})=0$，置信度为0；B_{gt}为训练样本标注的目标真实区域；B_{pt}为预测目标物体的边界框；a_{area}为指定图像区域的面积；$I_{\mathrm{OU_boundingbox}}^{\mathrm{groundtruth}}$为边界框和真实物体区域的交集与两者并集之比。每个网格还要预测C个类别，因此输出$S\times S\times(5\times B+C)$的一个张量。RF-YOLOv2检测流程如图1所示。首先调整输入图片的尺寸，并在图像上运行CNN提取图像特征；然后选取尺寸为13×13和26×26的特征图进行检测，目标中心所落在的网格负责预测该目标；最后由模型的置信度对所得到的检测结果进行阈值处理，进而得到最终检测结果。

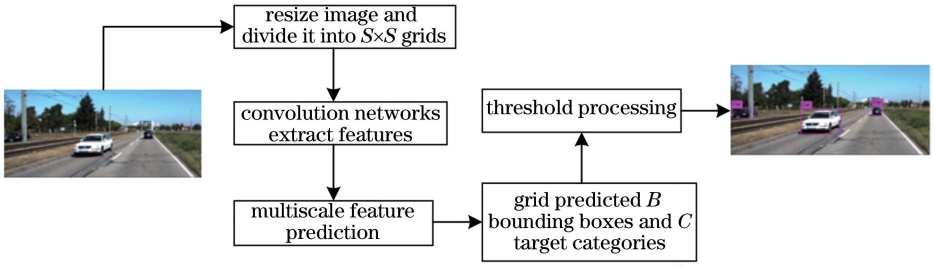

图1 RF-YOLOv2检测流程图

Fig. 1 Flowchart of RF-YOLOv2 detection

2.3 损失函数

损失函数采用均方差函数，公式为

$$L_{\mathrm{loss}}=L_{\mathrm{confidence}}+L_{\mathrm{coord}}+L_{\mathrm{groundtruth}},\tag{3}$$

式中：L_{loss}为网络训练的损失函数，由3部分组成；$L_{\mathrm{confidence}}$为背景的置信度误差；L_{coord}为先验框与预测框的坐标误差；$L_{\mathrm{groundtruth}}$为与每个真实框匹配的预测框各部分误差值，包括坐标误差、置信度误差和分类误差。背景的置信度误差$L_{\mathrm{confidence}}$计算公式为

$$L_{\mathrm{confidence}}=\sum_{i=0}^{W}\sum_{j=0}^{H}\sum_{k=0}^{A}l'\lambda_{\mathrm{noobj}}\cdot(-b_{ijk}^{(o)})^2,\tag{4}$$

式中：W、H分别为特征图的宽与高；A为每个网格单元对应的先验框数目；i、j分别为当前物体中心所在的行和列；k为当前目标所属的第k类别；$b_{ijk}^{(o)}$为当前网格没有目标，负责预测背景，o为背景；l'为当前预测框的最大交并比，小于设定的阈值；λ_{noobj}为没有目标的权重系数。当$\max(I_{\mathrm{OU}})$小于阈值设定的值时，预测框认为没有目标，这个预测框就标记为背景。默认先将所有的预测框都当作没有目标进行计算，直到某个预测框的最大交并比$\max(I_{\mathrm{OU}})$大于设定阈值时，再将该预测框的目标置信度损失设为0。先验框与预测框的坐标误差公式为

$$L_{\mathrm{coord}}=l\lambda_{\mathrm{prior}}\cdot\sum_{r\in(x,y,w,h)}(p_{\mathrm{rior},k}^{(r)}-b_{ijk}^{(r)})^2,\tag{5}$$

式中：λ_{prior}为先验框的权重系数；$p_{\mathrm{rior},k}^{(r)}$为第$k$类的先验框坐标；$b_{ijk}^{(r)}$为第$k$类预测框的坐标；$r$为先验框和预测框的位置；$l$为$t<12800$时计算先验框与预测框的坐标误差，$t$为图片的数量。因为在开始训练前期，候选框的尺寸是人为设定的几个固定值，不能满足所有待检测目标的先验框形状，所以为了保证训练前期的预测框能够快速学习到先验框的形状，在训练图片数量小于12800张的时候计算先验框与预测框的坐标误差。计算与每个真实框匹配的预测框各部分误差公式为

$$\begin{aligned}L_{\mathrm{groundtruth}}=&l_k^{t_{\mathrm{truth}}}\lambda_{\mathrm{coord}}\cdot\Big[\sum_{r\in(x,y,w,h)}(t_{\mathrm{truth}}^{(r)}-b_{ijk}^{(r)})^2\Big]+\\&\lambda_{\mathrm{obj}}\cdot(I_{\mathrm{OU},t_{\mathrm{truth}}}^{(k)}-b_{ijk}^{(o)})^2+\\&\lambda_{\mathrm{class}}\cdot\Big[\sum_{c=1}^{C}(t_{\mathrm{truth}}^{(c)}-b_{ijk}^{(c)})^2\Big],\end{aligned}\tag{6}$$

式中：等号右边第一项为坐标误差，第二项为置信度误差，第三项为分类误差；c为当前目标所属类别；C为总的类别个数之和；$l_k^{t_{\mathrm{truth}}}$为先验框内存在目标；λ_{coord}为坐标误差的权重系数；λ_{obj}为有目标的权重系数；λ_{class}为类别的权重系数；$t_{\mathrm{truth}}^{(r)}$为真实框坐标；$b_{ijk}^{(r)}$为预测框坐标；$I_{\mathrm{OU},t_{\mathrm{truth}}}^{(k)}$为真实框与当前所属类别的交并比值，$t_{\mathrm{truth}}^{(c)}$为物体的真实类别；$b_{ijk}^{(c)}$为预测框物体所属类别。对于每一个真实目标框，先确定其中心点所在的网格，然后计算这个网格的先验框与真实框的交并比值，计算交并比值时不考虑坐标，只考虑形状，所以先将先验框与真实框的中心点都偏移

到原点，然后计算出对应的交并比值，交并比值最大的先验框与真实框匹配，用来预测这个真实框。

3 RF-YOLOv2 模型

为有效地权衡目标检测的精确率和实时性，本文采用 ResNet 网络[18]和特征金字塔网络[19]中的思想，对 YOLOv2 模型进行改进，提出目标检测新模型 RF-YOLOv2。该模型首先通过聚类方法选取初始候选框个数和大小来提高检测速度和定位精度；然后通过加深网络模型的深度，进行更复杂的特征提取来获得更符合目标的特征；最后通过对不同尺寸特征图进行融合，使获得的特征包含更多的全局信息。

3.1 目标框维度聚类

YOLOv2 算法采用 Faster R-CNN 算法的思想，引入 anchor 机制。anchor boxes 是手工挑选的一组固定尺寸和宽高比的初始候选框。在网络训练过程中如果初始候选框的选择符合检测目标的特点，网络会更容易学到目标正确的预测位置，所以通过聚类方法对 KITTI 数据集[20]中的目标框进行聚类分析，从而选出最优候选框个数和宽高维度。实验采用手肘法确定 KITTI 数据集目标框的个数，手肘法思想是随着聚类数目的不断增加，每个簇的聚合程度就会不断提高，选取误差平方和函数（SSE）作为目标函数，即

$$S_{\mathrm{SSE}}=\sum_{a=1}^{n} w_a\,(y_a-m_a)^2 \tag{7}$$

式中：S_{SSE}为误差平方和函数；n 为所有样本个数之和；w_a 为第 a 个簇；y_a 为 w_a 中的样本点；m_a 为 w_a 中所有样本的均值。(7)式表示每个样本点到其聚类中心距离的平方和，聚类过程中目标函数变化如图 2 所示，其中 K 为候选框个数。

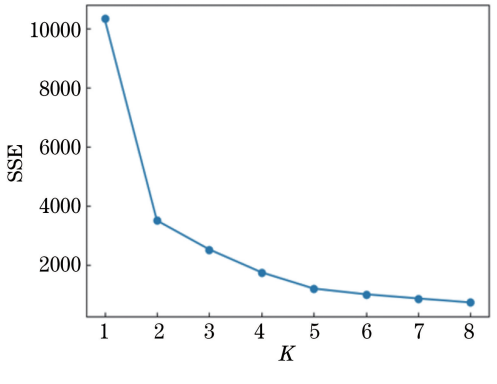

图 2 目标函数变化曲线

Fig. 2 Object function change curve

从图 2 可以看出，第一个拐点是 $K=2$，函数变化趋势减缓，但 $K=2$ 时，函数的损失值太大；第二个拐点是 $K=5$，此时函数损失值较小，取 5 为最优聚类个数。对于候选框的宽高比，实验希望能够通过 anchor boxes 获得更好的交并比分数，并且交并比分数与候选框的尺寸无关，所以选择距离度量函数为

$$d(b_{\mathrm{ox}},c_{\mathrm{centroid}})=1-I_{\mathrm{OU}}(b_{\mathrm{ox}},c_{\mathrm{centroid}}), \tag{8}$$

式中：b_{ox}、c_{centroid}分别为边界框和目标的中心；d 为边界框和目标中心的距离。当 $k=5$ 时 anchors 参数经过计算得出候选框尺寸分别为(0.38947，1.21642)、(0.94106，1.82330)、(0.69790，5.11059)、(1.82293，3.44036)和(3.12285，6.57371)。

3.2 RF-YOLOv2 模型

YOLOv2 模型一共包含 19 个卷积层，卷积层数量较少，使得提取的特征不能很好地描述目标，同时在进行目标检测时，只用尺寸大小为 13×13 特征图进行检测，使得感受野大小有限。这就导致 YOLOv2 模型在处理较远、较小、有遮挡等情况下的目标时，可能会造成漏检或者错检情况。

RF-YOLOv2 模型主要针对上述问题进行改进。首先针对卷积层数量较少问题对 YOLOv2 模型的卷积层数量进行翻倍，并且为了在增加更多的卷积层的同时避免梯度消失问题，引入 ResNet 中的残差块结构，如图 3 所示。

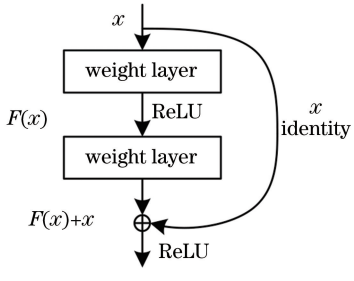

图 3 残差块结构

Fig. 3 Residual block structure

ResNet 提出 identity mapping 和 residual mapping，最后的输出公式为

$$y=F(x)+x, \tag{9}$$

式中：identity mapping 指本身，也就是式中的 x；residual mapping 指残差，也就是式中 $F(x)$部分。从图 3 可以看出 ResNet 能够在深层网络结构上缓解梯度消失问题，主要是因为增加了一个 identity mapping，可以将它看作是恒等映射，能把当前输出跳过本层运算直接传输给下一层网络，而不增加任何参数，同时在向后传播过程中也将下一层的网络梯度直接传递给上一层网络，从而缓解梯度消失

问题。

然后针对检测的特征图感受野大小有限问题，采用特征金字塔网络的思想。特征金字塔网络对不同尺度的目标采用不同尺度的特征进行预测，并采用自底向上和自顶向下的链接方式，每层的特征来源于当前层和更高层的特征融合。网络中相邻的特征图尺寸都是 2 倍关系，先将高层低分辨率特征图进行上采样，然后将上采样图和自下而上图通过元素相加方式合并。为了减少通道维度，可采用 1×1 的卷积核。最后为减少上采样的混叠效应，应在每个合并的图上添加一个 3×3 的卷积核，生成最终的特征图。融合方法如图 4 所示。

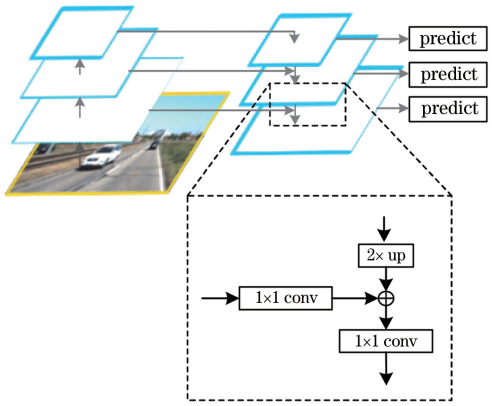

图 4 特征金字塔网络

Fig. 4 Feature pyramid network

在 RF-YOLOv2 中同时使用 13×13 尺寸和经过高层特征和低层特征融合的尺寸为 26×26 的特征图进行检测。特征融合过程是首先将尺寸为 13×13 的特征图经过上采样后与尺寸为 26×26 的特征图进行相加；再通过 3×3 的卷积核减少上采样的混叠效应；最终生成融合后的特征图。图 5 为 RF-YOLOv2 整体流程图，首先对输入的图像进行尺度归一化，将图片调整为需要的尺寸，进行图像预处理；然后将图像划分为$S\times S$个网格，并在图像上运行 CNN 提取图像特征。对尺寸为 13×13 和 26×26 的特征图进行检测，如果待检测的目标中心在某个网格上，那么该网格就负责检测该目标，每个网格预测 B 个边界框和 C 个类别，RF-YOLOv2 模型对尺寸为 13×13 的特征图预测 3 个边界框和 3 个类别，对尺寸为 26×26 的特征图预测 2 个边界框和 3 个类别；最后由模型的置信度对所得到的检测结果进行阈值处理，得到最后的检测结果。

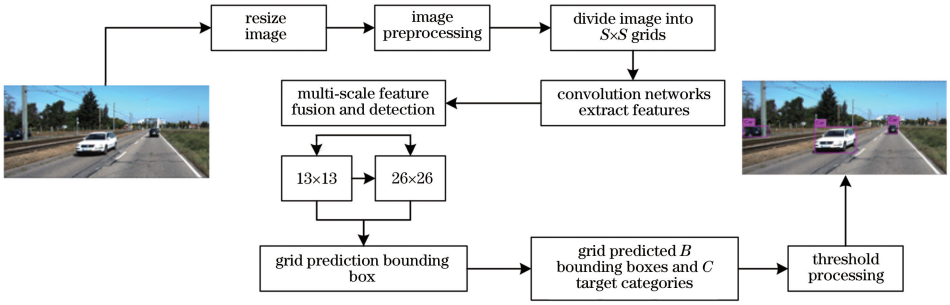

图 5 RF-YOLOv2 流程图

Fig. 5 Flowchart of RF-YOLOv2

4 实验结果及分析

4.1 实验数据及实验参数配置

实验在平台为 i7-6700 处理器、内存为 16 GB、显卡为 NVIDIA GTX1060 的 Ubuntu14.04 操作系统中进行。实验使用的数据集是 KITTI 数据集，该数据集的图像数据来源于市区、乡村和高速公路等场景，每张图像有各种程度的光照和遮挡情况，最多包含 15 辆车和 30 个行人。

实验选取此数据集中目标检测部分进行实验，KITTI 目标检测数据包括 7481 个训练图像和 7518 个测试图像，共有 80256 个标记目标。由于测试图像没有给定标注信息，因此在进行训练和测试时，实验将 7481 张图片，按照 8∶1∶1 的比例，用 5984 张图片作为训练集，784 张图片作为验证集，713 张图片作为测试集。数据标签细分为 car、van、truck、pedestrian、person_sitting、cyclist、tram 以及 misc。不同类别在 KITTI 数据集上出现的数量如图 6 所示。可以看出 car 类别占数据集的比例最大，其次是 pedestrian 类别、van 类别和 cyclist 类别。实验中将 8 个标签类别合并为 3 个标签类别，分别是 car、pedestrian 和 cyclist。为有效利用数据，本实验将数据标签为 van、truck、tram 的目标合并到 car 类别中，将数据标签为 person_sitting 的目标合并到 pedestrian 类别中，将数据标签为 misc 的目标忽略。

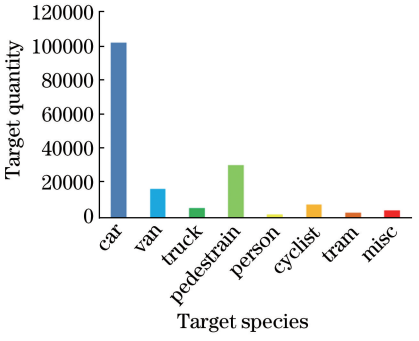

图 6 各类别在 KITTI 数据集上出现的数量

Fig. 6 Number of categories appearing on KITTI data set

在 KITTI 数据集上进行训练时，为了只验证本文模型的有效性，参数的选择与原始模型 YOLOv2 保持一致。初始学习率设定为 0.001，学习策略为 steps，迭代 40000 次后，学习率下降为0.0001，继续迭代到 50000 次。动量为 0.9，权重衰减系数为 0.0005，饱和度和曝光变化大小为 1 到 1.5 倍，色调变化范围为−0.1～0.1。

4.2 实验评价指标

实验分别使用每秒传输帧数、精确率、召回率、交并比、损失值和 Precision-Recall 曲线图来对比分析模型的性能。每秒传输帧数可以度量目标检测的处理速度。交并比可以评估目标边界框定位的准确性，通常交并比的阈值设为 0.5，为了保证实验结果更加准确，实验采用交并比的阈值为 0.6。损失值表示预测值与真实值的相近程度，损失值越小，说明模型的稳健性越好。精确率和召回率可以评估网络模型性能，精确率表示被预测为某类别中真正属于该类别的目标比例，召回率表示所有被正确识别的目标占所有应该被识别目标的比例。精确率和召回率的公式分别为

$$P_{\text{precision}}=\frac{T_{\text{P}}}{T_{\text{P}}+F_{\text{P}}},\tag{10}$$

$$R_{\text{recall}}=\frac{T_{\text{P}}}{T_{\text{P}}+F_{\text{N}}},\tag{11}$$

式中：T_{P} 为正确检测到的样本数量；F_{P} 为未被检测到的正确样本数量；F_{N} 为被错误检测到的样本数量。

KITTI 数据集使用 PASCAL VOC 数据集[21]中定义的方法来评估单类目标检测模型的结果，用 Precision-Recall 曲线进行定性分析，用平均精确率（A_{P}）定量分析模型的精度。平均精确率是 Precision-Recall 曲线的积分值，KITTI 数据集将召回率划分为 41 个等间距的阈值，分类器的平均精确率为阈值点上召回率的精确值的平均值。首先设定召回率在[0,1]范围内取 41 个等间距阈值，然后可以在每一个阈值区间得到一个最大精确值[21]。

4.3 实验结果与分析

实验对本文模型 RF-YOLOv2、YOLOv2 模型和 YOLOv3 模型采用设置好的参数各自训练 50000 次，识别单类目标的精确率和检测速度见表 2。

表 2 精确率和检测速度对比

Table 2 Comparison of accuracy and detection speed

Model	Accuracy of car /%	Accuracy of pedestrian / %	Accuracy of cyclist / %	Detection speed / (frame·s^{-1})
YOLOv2	68.56	44.26	55.95	**46.4**
RF-YOLOv2	87.88	52.91	74.05	30.3
YOLOv3	**89.34**	**60.93**	**83.94**	23.1

对比 YOLOv2 模型，RF-YOLOv2 模型对标签为 car 的目标识别率提高了 19.3 个百分点，对标签为 pedestrian 的目标识别率提高了 8.6 个百分点，对标签为 cyclist 的目标识别率提高了大约 18.1 个百分点；尽管 RF-YOLOv2 模型在速度上下降了 16.1 frame/s，但依然能满足目标检测实时性的需求。对比于 YOLOv3 模型，RF-YOLOv2 模型在检测精确率上有一定差距，这是由于 RF-YOLOv2 模型网络结构更加简化，网络参数减少造成精度损失。从实验数据发现 RF-YOLOv2 模型对于标签为 car 的目标识别率和 YOLOv3 模型相差只有1.4%左右，但 RF-YOLOv2 模型的检测速度要快于 YOLOv3 模型，整体速度上增加了 7.2 frame/s。

实验统计 YOLOv2 模型和 RF-YOLOv2 模型在训练次数每隔 10000 次时召回率和交并比的变化，两种模型召回率和交并比的变化过程见表 3。可以看出随着训练次数的增加，两种模型的召回率和交并比值都在逐步上升，并且本文模型在训练 30000 次时的召回率就超过了 YOLOv2 模型 50000 次的召回率。在结束训练时本文模型相比 YOLOv2 模型召回率提高了 7.9 个百分点，交并比提高了 4.6 个百分点，说明本文模型 RF-YOLOv2 找到的正确目标更多，产生的边界框更加精确。

模型刚开始训练时，模型参数随机初始化，所以模型前期的损失值较大，实验对比 YOLOv2 模型和 RF-YOLOv2 模型从 10000 次到 50000 次的平均损失值，并将其绘制成图，如图 7 所示。可以看出停止训练时，两种模型的损失值结果仍在减小。刚开始训练时YOLOv2模型更为简单，收敛速度较快，但

表 3 召回率和交并比的变化过程

Table 3 Change process of recall rate and I_{OU}

Number of training	RF-YOLOv2 model		YOLOv2 model	
	Recall rate /%	I_{OU} /%	Recall rate /%	I_{OU} /%
10000	50.36	43.29	48.18	43.42
20000	55.45	46.34	53.11	45.98
30000	61.47	50.65	55.83	47.79
40000	64.92	52.56	54.13	46.72
50000	**65.87**	**53.63**	57.98	49.04

是随着训练次数的不断增加，本文模型在 40000 次左右时损失值已经低于 YOLOv2 模型的损失值，说明本文模型的稳健性好于 YOLOv2 模型。

KITTI 数据集根据目标的遮挡程度和目标样本高度的最小像素值，将目标分为简单样本、中等样本和困难样本 3 类。实验对比了 YOLOv2 模型、Faster-rcnn 模型和 RF-YOLOv2 模型对 3 种类别标签的 3 种样本检测精确率。

表 4 是 3 种模型对 car 类别三种样本的检测结果。可以看出 YOLOv2 模型和 Faster-rcnn 模型的精确率总体低于 RF-YOLOv2 模型。在 car 标签的 3 种样本中，RF-YOLOv2 模型的简单样本的精确率比 YOLOv2 模型提高了 20.4 个百分比，中等样本的精确率提高了 23.9 个百分比，困难样本的精确率提高了 21.9 个百分比。RF-YOLOv2 模型的简单样本的精确率比 Faster-rcnn 模型提高了 3.1 个百分点，中等样本的精确率提高了 2.1 个百分点，困难样本的精确率提高了 2.22 个百分点。

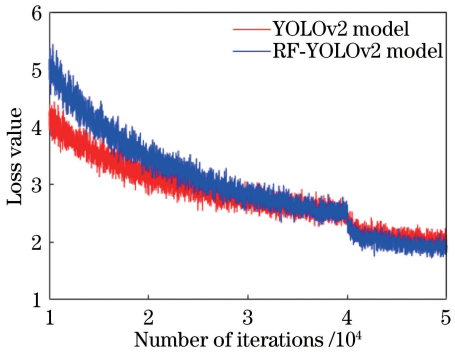

图 7 两种模型的损失图

Fig. 7 Loss graph for two models

表 4 car 类别三种样本检测结果

Table 4 Three sample detection results of car category

Model	Accuracy of easy sample /%	Accuracy of moderate sample /%	Accuracy of hard sample /%
YOLOv2	70.56	57.32	50.44
Faster-rcnn	87.90	79.11	70.19
RF-YOLOv2	**91.01**	**81.26**	**72.41**

表 5 是 3 种模型对 pedestrian 类别三种样本检测结果。可以看出 RF-YOLOv2 模型对目标检测的精确率高于 YOLOv2 模型，但逊于 Faster-rcnn 模型。在 pedestrian 标签的 3 种样本中，RF-YOLOv2 模型的简单样本的精确率比 YOLOv2 模型提高了 4.3 个百分点，中等样本的精确率提高了 7.9 个百分点，困难样本的精确率提高了 9.0 个百分点。RF-YOLOv2 模型的简单样本的精确率比 Faster-rcnn 模型降低了 14 个百分点，中等样本的精确率降低了 8.9 个百分点，困难样本的精确率降低了 7.2 个百分点。

表 5 pedestrian 类别三种样本检测结果

Table 5 Three sample detection results of pedestrian category

Model	Accuracy of easy sample /%	Accuracy of moderate sample /%	Accuracy of hard sample /%
YOLOv2	59.97	49.05	44.91
Faster-rcnn	**78.35**	**65.91**	**61.19**
RF-YOLOv2	64.35	57.02	53.94

表 6 是 3 种模型对 cyclist 类别三种样本检测结果。可以看出 YOLOv2 模型和 Faster-rcnn 模型的精确率总体低于 RF-YOLOv2 模型。在 cyclist 标签的 3 种样本中，RF-YOLOv2 模型的简单样本的精确率比 YOLOv2 模型提高了 23.2 个百分比，中等样本的精确率提高了18.0个百分比，困

表 6 cyclist 类别三种样本检测结果

Table 6 Three sample detection results of cyclist category

Model	Accuracy of easy sample /%	Accuracy of moderate sample /%	Accuracy of hard sample /%
YOLOv2	56.47	56.68	53.02
Faster-rcnn	71.41	62.81	55.44
RF-YOLOv2	79.76	74.68	72.41

难样本的精确率提高了 19.3 个百分比。RF-YOLOv2 模型的简单样本的精确率比 Faster-rcnn 模型提高了 8.3 个百分比，中等样本的精确率提高了 11.8 个百分比，困难样本的精确率提高了 16.9 个百分比。

通过对比不同标签样本精确率的值，RF-YOLOv2 模型总体好于 YOLOv2 模型，但与 Faster-rcnn 相比只是在 pedestrian 标签上精确率有所降低，这是因为 pedestrian 标签的目标尺寸较小，但不能忽视的是 Faster-rcnn 的速度远低于 RF-YOLOv2 模型，每秒检测帧数为 0.5 frame/s。实验发现 RF-YOLOv2 模型对 car 标签和 cyclist 标签的 3 种样本提升更为明显，对 pedestrian 标签的样本提升较少，考虑到 KITTI 数据集本身的特点，car 标签和 cyclist 标签的 3 种样本目标尺寸要大于 pedestrian 标签的目标尺寸，说明 RF-YOLOv2 模型对稍大目标提升效果更好。

精确率和召回率互相影响，理论上追求两者都高，但是实际上两者相互制约。通过 Precision-Recall 曲线图能更直观地得到两者关系。图 8 比较了 YOLOv2 模型和本文模型 3 种标签样本的 Precision-Recall 曲线图。其中图 8 (a)、(b)分别是 YOLOv2 模型和本文模型对 car 标签的 Precision-Recall 曲线图，图 8 (c)、(d)分别是 YOLOv2 模型和本文模型对 pedestrian 标签的 Precision-Recall 曲线图，图 8(e)、(f)分别是 YOLOv2 模型和本文模型对 cyclist 标签的 Precision-Recall 曲线图。通过对比发现在相同精确率时，本文模型对目标的召回率更高，说明本文模型在保证精确率的同时提高了召回率。

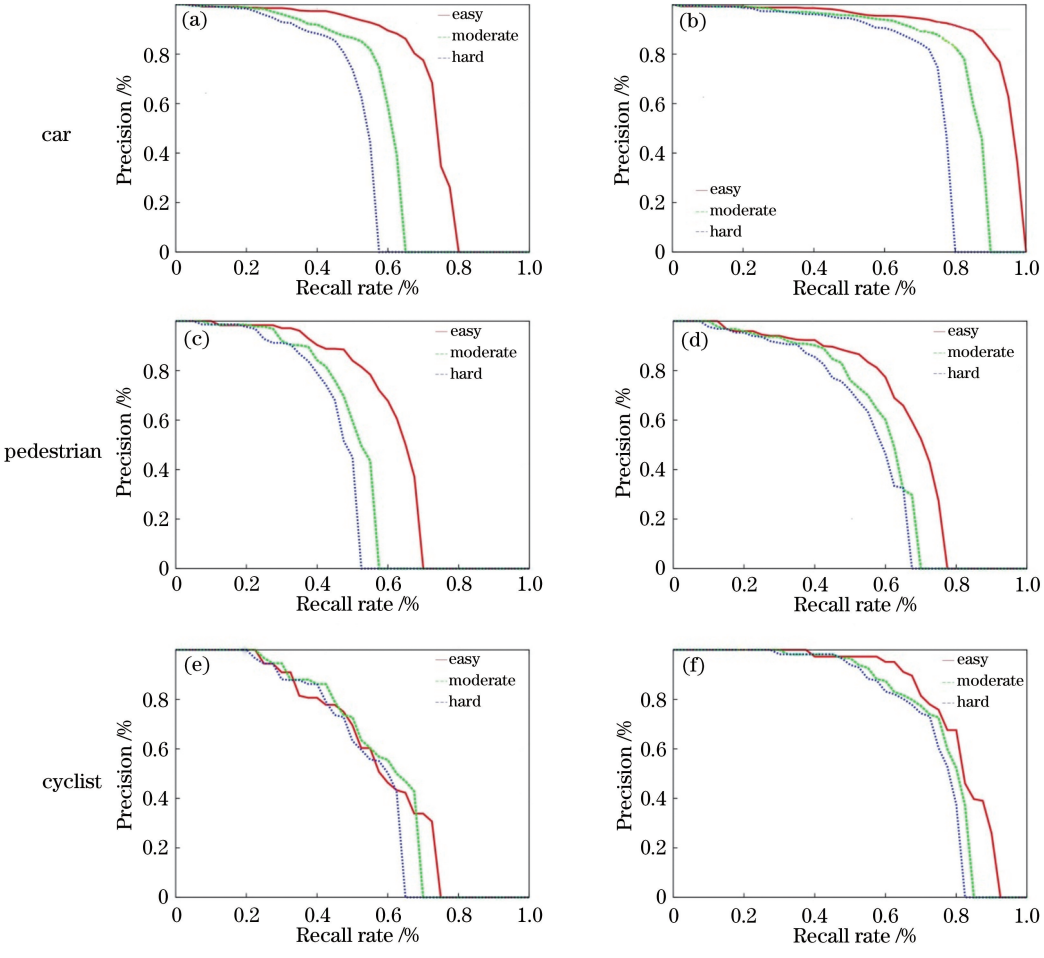

图 8 两种模型的 Precision-Recall 曲线图。(a)(c)(e) YOLOv2 模型；(b)(d)(f) RF-YOLOv2 模型

Fig. 8 Precision-Recall curves of two models. (a)(c)(e) YOLOv2 model; (b)(d)(f) RF-YOLOv2 model

对测试数据集中的样本进行测试实验，结果如图 9 所示。其中图 9 (a)、(c)、(e)、(g)、(i)是 YOLOv2 模型的检测结果，图 9 (b)、(d)、(f)、(h)、(j)是 RF-YOLOv2 模型的检测结果。对比检测结果，可以验证使用聚类算法得到候选框个数和尺寸的有效性，如对比图 9(a)、(b)可以看出 RF-YOLOv2 模型的候选框对较远的白色车辆定位更加精准，说明 RF-YOLOv2 模型具有更精确的定位效果；对比图 9(c)、(d)可以看

图 9 检测结果图。(a)(c)(e)(g)(i) YOLOv2 模型检测结果;(b)(d)(f)(h)(j) RF-YOLOv2 模型检测结果

Fig. 9 Detection results. (a)(c)(e)(g)(i) Detection results of YOLOv2 model; (b)(d)(f)(h)(j) detection results of RF-YOLOv2 model

到 YOLOv2 模型只检测出少量目标,而 RF-YOLOv2 模型在有大面积阴影的情况下依然检测出全部目标,说明 RF-YOLOv2 模型对处于阴影下的车辆识别效果更好;对比图 9(e)、(f)可以看到 YOLOv2 模型在大量密集目标且有遮挡的情况下,识别效果并不理想,而 RF-YOLOv2 模型在密集且有严重遮挡的情况下依然能检测出大部分目标;对比图 9(g)和图 9(h)、图 9(i)和图 9(j)发现 YOLOv2 模型对远处小目标有漏检情况,而 RF-YOLOv2 模型对此有所改进,降低了小目标的漏检率。可以得出结论:RF-YOLOv2 模型对有物体遮挡以及在较强光照或阴影下目标检测的精确率要高于YOLOv2 模型。

5 结 论

在 YOLOv2 模型的基础上进行改进,针对目标检测中由道路场景复杂、目标外形多变等特点导致的检测率不高和定位不准确问题,设计一种新模型 RF-YOLOv2。并将该模型在 KITTI 数据集上进行实验,通过不同评价指标对比本文模型和 YOLOv2 模型,从多角度证明了本文 RF-YOLOv2 模型的有效性。1)在每秒传输帧数上,本文模型稍逊于 YOLOv2 模型,但仍能满足实时性的需求;2)在交并比上,本文模型交并比值更大,对目标定位更加精确;3)在损失值上,从损失值的走向对比可以看出本文模型的稳健性更好;4)在召回率、精确率和 Precision-Recall 曲线图上,本文模型对目标识别的精确率和召回率有更好的平衡能力,在保证精确率时,能拥有更大的召回率。

实验中发现本文模型对目标为 car 类别和 cyclist 类别的精确率提高较多,对目标为 pedestrian 类别的精确率提高较小,是因为目标为 pedestrian 类别相对于 car 类别和 cyclist 类别目标尺寸较小,因此下一步研究重点将放在提高模型对小目标检测的精确率上。

参 考 文 献

[1] Zhang H, Wang K F, Wang F Y. Advances and perspectives on applications of deep learning in visual object detection[J]. Acta Automatica Sinica, 2017, 43(8): 1289-1305.
张慧, 王坤峰, 王飞跃. 深度学习在目标视觉检测中的应用进展与展望[J]. 自动化学报, 2017, 43(8): 1289-1305.

[2] Zhou F Y, Jin L P, Dong J. Review of convolutional neural network[J]. Chinese Journal of Computers, 2017, 40(6): 1229-1251.
周飞燕, 金林鹏, 董军. 卷积神经网络研究综述[J]. 计算机学报, 2017, 40(6): 1229-1251.

[3] Wang H X, Dong H, Zhou Z Q. Review on dim small target detection technologies in infrared single frame images[J]. Laser & Optoelectronics Progress, 2019, 56(8): 080001.
王好贤, 董衡, 周志权. 红外单帧图像弱小目标检测技术综述[J]. 激光与光电子学进展, 2019, 56(8): 080001.

[4] Ou P, Zhang Z, Lu K, *et al*. Object detection in of remote sensing images based on convolutional neural networks[J]. Laser & Optoelectronics Progress, 2019, 56(5): 051002.
欧攀, 张正, 路奎, 等. 基于卷积神经网络的遥感图像目标检测[J]. 激光与光电子学进展, 2019, 56(5): 051002.

[5] Felzenszwalb P, Girshick R, McAllester D, *et al*. Visual object detection with deformable part models[J]. Communications of the ACM, 2013, 56(9): 97-105.

[6] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]// 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), June 20-25, 2005, San Diego, CA, USA. New York: IEEE, 2005: 8588935.

[7] Lin C F, Wang S D. Fuzzy support vector machines[J]. IEEE Transactions on Neural Networks, 2002, 13(2): 464-471.

[8] Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// 2014 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH, USA. New York: IEEE, 2014: 580-587.

[9] Girshick R. Fast R-CNN[C]// 2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE, 2015: 1440-1448.

[10] Ren S Q, He K M, Girshick R, *et al*. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[11] Uijlings J R R, van de Sande K E A, Gevers T, *et al*. Selective search for object recognition[J]. International Journal of Computer Vision, 2013, 104(2): 154-171.

[12] Redmon J, Divvala S, Girshick R, *et al*. You only look once: unified, real-time object detection[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 779-788.

[13] Liu W, Anguelov D, Erhan D, *et al*. SSD: single shot MultiBox detector[M]// Leibe B, Matas J, Sebe N, *et al*. Computer vision-ECCV 2016. Lecture notes in computer science. Cham: Springer, 2016, 9905: 21-37.

[14] Wang X Q, Wang X J. Real-time target detection method applied to embedded graphic processing unit[J]. Acta Optica Sinica, 2019, 39(3): 0315005.
王晓青, 王向军. 应用于嵌入式图形处理器的实时目标检测方法[J]. 光学学报, 2019, 39(3): 0315005.

[15] Szegedy C, Liu W, Jia Y Q, *et al*. Going deeper with convolutions[C]// 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 7-12, 2015, Boston, MA, USA. New York: IEEE, 2015: 15523970.

[16] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 6517-6525.

[17] Redmon J, Farhadi A. YOLOv3: an incremental improvement[J/OL]. (2018-04-08)[2019-05-09]. https://arxiv.org/abs/1804.02767.

[18] He K M, Zhang X Y, Ren S Q, *et al*. Deep residual learning for image recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 770-778.

[19] Lin T Y, Dollár P, Girshick R, *et al*. Feature pyramid networks for object detection[C]// 2017

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 936-944.

[20] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite[C] // 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 16-21, 2012, Providence, RI, USA. New York: IEEE, 2012: 3354-3361.

[21] Everingham M, Eslami S M A, van Gool L, *et al*. The Pascal visual object classes challenge: a retrospective[J]. International Journal of Computer Vision, 2015, 111(1): 98-136.