doi:10.19562/j.chinasae.qcgc.2024.12.015

SFW-YOLOv8复杂场景视频车辆检测模型*

祝 琴[1,2],韩沈阳[2],曾明如[2],赖平红[3],吴垂茂[2],胡玮轶[2]
(1. 南昌大学公共政策与管理学院,南昌 330036; 2. 南昌大学信息工程学院,南昌 330036;
3. 江西省人民医院,南昌 330038)

[摘要] 针对复杂交通监控场景中视频车辆检测模型难以提取丰富的目标特征的问题,本文从充分利用视频图像时空特征信息的角度,新建时空特征融合模块SF-Module,运用Transformer模型中的多头自注意力机制实现视频车辆图像当前帧和历史帧时空特征信息的提取和融合,丰富目标的特征信息;在此基础上,基于YOLOv8网络,在其颈部网络融合新建的时空特征融合模块SF-Module,挖掘视频图像序列的时空特征信息;同时,引入WIoU损失函数作为预测框回归损失,减少低质量标注框产生的有害梯度,设计SFW-YOLOv8视频车辆检测模型。最后,新建的SFW-YOLOv8复杂场景视频车辆检测模型在UA-DETRAC数据集上进行实验,对数据集中的部分图片进行了模拟雨天和雾天的数据增强,提高车辆检测模型的泛化性。实验结果表明,SFW-YOLOv8视频车辆检测模型的*MAP*50和*MAP*50:5:95值为79.1%和63.6%,较YOLOv8模型分别提高了1.7%和3.3%,推理速度为11 ms/帧,具有较为优秀的检测性能。

关键词:车辆目标检测;时空特征融合;Transformer;YOLOv8;注意力机制

SFW-YOLOv8 Complex Scene Video Vehicle Detection Model

Zhu Qin[1,2], Han Shenyang[2], Zeng Mingru[2], Lai Pinghong[3], Wu Chuimao[2] & Hu Weiyi[2]
1. *School of Public Policy and Management, Nanchang University, Nanchang* 330036;
2. *School of Information Engineering, Nanchang University, Nanchang* 330036;
3. *Jiangxi Provincial People's Hospital, Nanchang* 330038

[**Abstract**] For the problem that it is difficult for video vehicle detection models to extract rich target features in complex traffic monitoring scenarios, in this paper a new spatial-temporal feature fusion module SF-Module is established from the perspective of making full use of spatial-temporal feature information of video images. The multi-head self-attention mechanism in Transformer model is used to extract and fuse the temporal and spatial feature information of current and historical frames of video vehicle images to enrich the feature information of the target. On this basis, based on YOLOv8 network, the newly created spatio-temporal feature fusion module SF-Module is integrated in its neck network to mine spatio-temporal feature information of video image sequences. At the same time, the WIoU loss function is introduced as the prediction frame regression loss to reduce the harmful gradient generated by the low quality label frame, and the SFW-YOLOv8 video vehicle detection model is designed. Finally, the newly established SFW-YOLOv8 complex scene video vehicle detection model is tested on the UA-DETRAC dataset, and some images in the dataset are simulated to enhance the data on rainy and foggy days, so as to improve the generalization of the vehicle detection model. The experimental results show that the values of *mAP*50 and *mAP*50:5:95 of the SFW-YOLOv8 video vehicle detection model are 79.1% and 63.6%, which are 1.7% and 3.3% higher than that of the YOLOv8 model, respectively. The reasoning speed is 11 ms/ frame, which has excellent detection performance.

Keywords: vehicle target detection; spatio-temporal feature fusion; Transformer; YOLOv8; attention mechanism

*国家自然科学基金(72164027)资助。
原稿收到日期为2024年04月26日,修改稿收到日期为2024年06月12日。
通信作者:曾明如,教授,硕士生导师,E-mail:zeng_mr@163.com。

前言

车辆检测算法可以分为两类,分别为传统的车辆检测算法和基于深度学习的车辆检测算法。传统的车辆检测方法已难以满足智慧交通对车辆检测与跟踪的高性能要求,而基于深度学习的车辆检测与跟踪方法在检测和跟踪准确度、实时性上均具有优势。

早期传统的车辆检测算法有帧间差分法[1]、背景差分法[2]、光流法[3]等算法。帧间差分法主要通过对比视频序列相邻帧像素间的差异,建立目标的运动联系,获取运行目标轮廓实现目标检测,算法中时间间隔的设定较难。背景差分法通过当前帧与背景模型对比实现运动物体检测,该方法中的背景模型难建立,复杂环境下的检测效果差。光流法通过图像序列中像素在时间域上的变化计算物体运动信息实现目标检测,该方法受光照等外界环境影响大。研究发现,除了使用运动特性实现车辆检测的方法,机器学习方法也在传统车辆检测任务中广泛应用[4]。机器学习方法是基于区域选择特征实现的,车辆检测流程主要分为3步[5]。传统的车辆检测算法存在易受检测环境影响、需要手工设计提取特征、需要使用计算冗余的滑动窗口定位车辆等问题,导致车辆检测算法的复杂度高、实时性差以及适用性不强。因此,应用深度学习技术研究设计车辆检测算法逐渐成为主流。

基于深度学习的车辆检测算法使用深度神经网络提取车辆的特征并通过神经网络对车辆进行分类和位置回归,由于无须手工设计和提取特征且无须使用滑动窗口定位车辆位置等优点,基于深度学习的车辆检测算法迅速发展且在准确度和实时性上都超过了传统的车辆检测算法。当前,基于深度学习的车辆检测算法根据是否生成候选区域可分为两阶段车辆检测算法和单阶段车辆检测算法。

两阶段目标检测算法须先生成目标候选框,然后通过卷积神经网络实现目标分类和位置回归修正。2014年,Girshick等[6]提出了R-CNN检测算法,R-CNN作为两阶段目标检测算法的开山之作,相较于传统目标检测算法,其检测精度更高,但检测速度较慢。在随后几年,众多学者陆续提出了SPPNet[7]、Fast R-CNN[8]、Faster R-CNN[9]、Mask R-CNN[10]等两阶段算法,这些算法在候选区域选择、候选区域特征提取以及目标分类和回归等方面对R-CNN进行了改进,大大提高了两阶段目标检测算法的检测准确度和实时性。随着两阶段目标检测算法的进步,大量科研团队[11]将其应用在车辆检测领域。这些两阶段的车辆检测算法可以实现较高的检测准确度,但由于需要生成候选区域,检测实时性上不如单阶段车辆检测算法。

单阶段检测算法无须生成目标的候选区域,其通过深度神经网络提取图像特征,然后直接回归待检测目标的类别概率和位置信息。2015年,Redmon等[18]提出了首个单阶段检测算法YOLO(you only look once),YOLO算法将目标检测看作是一个回归问题,在空间上将整个图像分割成固定数量的网格单元,对每个单元格预测该位置是否有对象、边界框坐标和大小以及对象的类别。YOLO单一的网络框架使其在检测速度上比两阶段检测算法具有明显优势,但是在检测多目标、小目标以及检测准确率上和优秀的两阶段检测算法相比还是有差距。此后,SSD[19]、RetinaNet[20]、YOLO系列[21-26]、DETR[27]等一系列单阶段检测算法陆续被提出,单阶段检测算法逐渐在检测准确性和实时性上优于两阶段检测算法。由于单阶段检测算法有更好的综合性能,特别是有着优秀的实时性,很多学者[28-34]将单阶段检测算法应用到了车辆检测领域。单阶段的目标检测算法得益于其端到端的检测网络设计,具有检测速度快、精度高的优点,在车辆检测以及其它领域均成为了主流的检测方法。

无论是两阶段检测算法还是单阶段检测算法,其主要考虑的应用场景是单张图片的检测,即使是视频序列图片,也仅是分解成单张图片进行检测,并没有将视频序列图片关联起来,视频目标检测尤其是复杂场景视频目标检测准确度较低。因此,有学者[35-39]通过提取融合视频序列中的时空信息,丰富目标特征,提高视频检测的准确度。在视频目标检测任务中,提取融合视频序列中的时空信息,可以有效提高检测准确性。

本文从充分利用视频图像时空特征信息的角度,建立时空特征融合模块SF-Module,在此基础上,基于YOLOv8网络,融合所建立的时空特征融合模块SF-Module;同时,引入WIoU损失函数作为预测框回归损失,设计SFW-YOLOv8视频车辆检测模型。车辆检测模型在UA-DETRAC数据集[40]上进行实验,对数据集中的部分图片进行了模拟雨天和雾天的数据增强,提高车辆检测模型的泛化性。

1 数据预处理

1.1 数据集划分

在对SFW-YOLOv8进行训练时,将UA-DETRAC数据集划分为两个数据集,数据集1用于对YOLOv8模型进行初步训练,而数据集2用于对SFW-YOLOv8模型进行训练。SFW-YOLOv8训练时会载入YOLOv8在数据集1中训练好的权重,并且训练时只对SFW-Module模块进行训练,其它层全部冻住。SFW-YOLOv8的训练流程如图1所示。

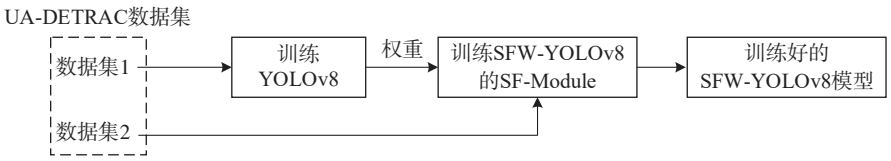


图1 SFW-YOLOv8训练流程图

UA-DETRAC数据集的数据量巨大,如果使用全部数据进行训练会有很大的训练成本,所以本文将对UA-DETRAC进行缩减。对于训练YOLOv8的数据集1,是在UA-DETRAC数据集的每个视频图像上每隔10张选取一张,最终获取的训练集图片为8 209张,测试集图片为5 617张。

SFW-YOLOv8训练时需要连续的图像作为输入,所以数据集2的选取不能间隔获取。因此,训练SFW-YOLOv8的数据集2,将在UA-DETRAC数据集中选取视频场景较为不同的视频图像,训练集选取了30个视频的图像,测试集选取了18个视频的图像。为了同时训练多个视频图像时更加便捷,同时为了减少训练成本,将选取的视频图像再缩减到相同的数量,使用训练集30个视频图像和测试集18个视频图像中的最小视频数694作为统一数量,选取这些视频的前694帧作为训练和测试的数据。最终形成的数据集2,训练集图片为20 820张,测试集图片为12 492张。

UA-DETRAC数据集不仅标注了车辆的位置信息,还标注了道路中的忽略区域,如图2所示,蓝色框为车辆标注框,黑色为忽略区域标注框。由于UA-DETRAC数据集忽略区域中也存在车辆但是却没有对应的车辆标注信息,这会影响车辆检测模型的训练,降低车辆检测模型的准确率。因此,将对忽视区域的图像进行黑化处理,降低该部分图像对检测网络训练的影响。UA-DETECT数据集中标注了忽视区域相对于图像的位置坐标,而本文是根据这些标注信息对忽视区域进行黑化,即划分规则就是数据集中忽视区域的标注信息。黑化处理后的图像如图3所示。忽视区域中的少量标签信息,在黑化过程中也会将该标注信息从对应的标签文件中删除。

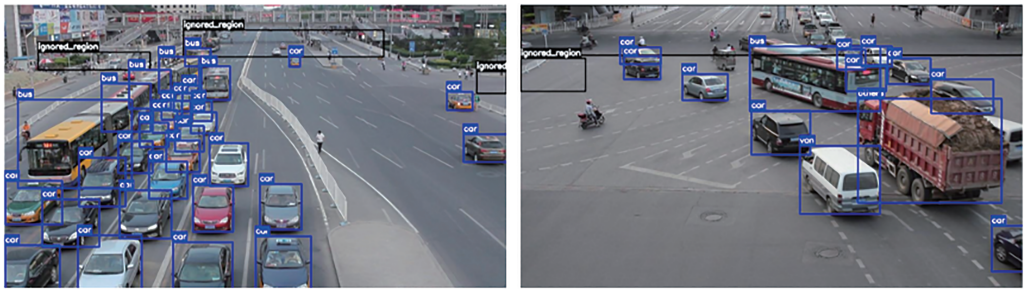


图2 UA-DETRAC数据集中忽略区域的展示

1.2 数据增强

在YOLOv8中,输入到网络的数据会依次经过Mosaic、MixUp、随机HSV色彩变换、随机水平翻转等数据增强方法。部分数据增强方法的效果如图4所示。

YOLOv8的数据增强方法较难提高模型在大雨、大雾这种恶劣天气情况下的检测能力,而UA-DETRAC数据集虽然包含了晴天、小雨、夜晚场景的数据,但缺乏大雾、大雨等恶劣天气场景数据,影响模型的泛化能力。基于此,本文将对数据集1和数

图3　UA-DETRAC数据集忽略区域黑化效果图

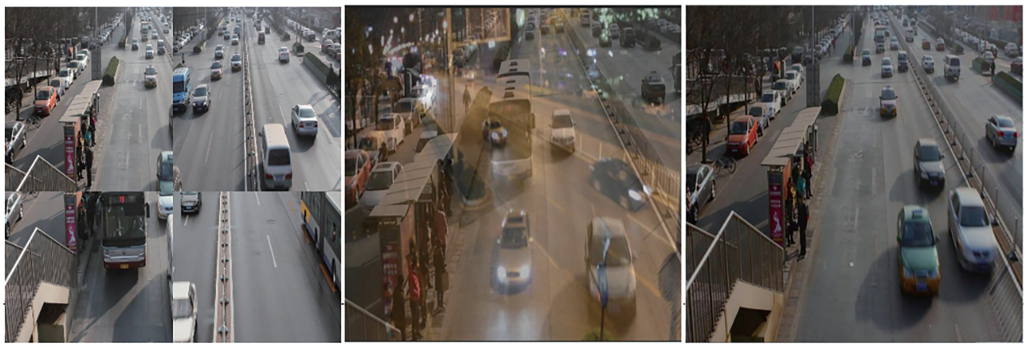


图4　YOLOv8图像增强部分展示

据集2均通过模拟增强大雾天和大雨天的数据,以提高SFW-YOLOv8模型在现实交通场景中的泛化能力。雾天模拟的方法是通过RGB通道合成雾,即通过OpenCV库(cv2)中的addWeighted()函数实现的,其作用是将两张源图片以一定的权重进行混合,本文将源图与纯白图片以一定权重混合。雨天模拟的方法首先是随机生成与源图维度大小一致的噪声,然后通过阈值控制噪声的多少,即雨点的多少,再通过对随机噪声进行拉长、旋转,模拟不同大小和方向的雨滴,最后将原图与雨点噪声图通过RGB通道合成。模拟雾天、雨天数据增强前后的效果如图5所示。

最后,在训练YOLOv8和SFW-YOLOv8时使用的数据增强流程如图6所示。

图5　UA-DETRAC数据集模拟雾天雨天的效果图

图6　YOLOv8和SFW-YOLOv8的数据增强流程

在测试阶段或者推理阶段,SFW-YOLOv8只会对输入的图片通过LetterBox操作进行resize处理,即保持原图的长宽比进行比例缩放,当长边resize到需要的长度时,短边剩下的部分采用灰色填充。

2 时空特征融合模块SF-Module

SFW-YOLOv8车辆检测模型的网络结构是基于YOLOv8-L设计,其网络结构主要分为主干网络Backbone、颈部网络Neck和解耦头Decoupled Head 3部分,SFW-YOLOv8车辆检测模型的网络结构如图7所示。

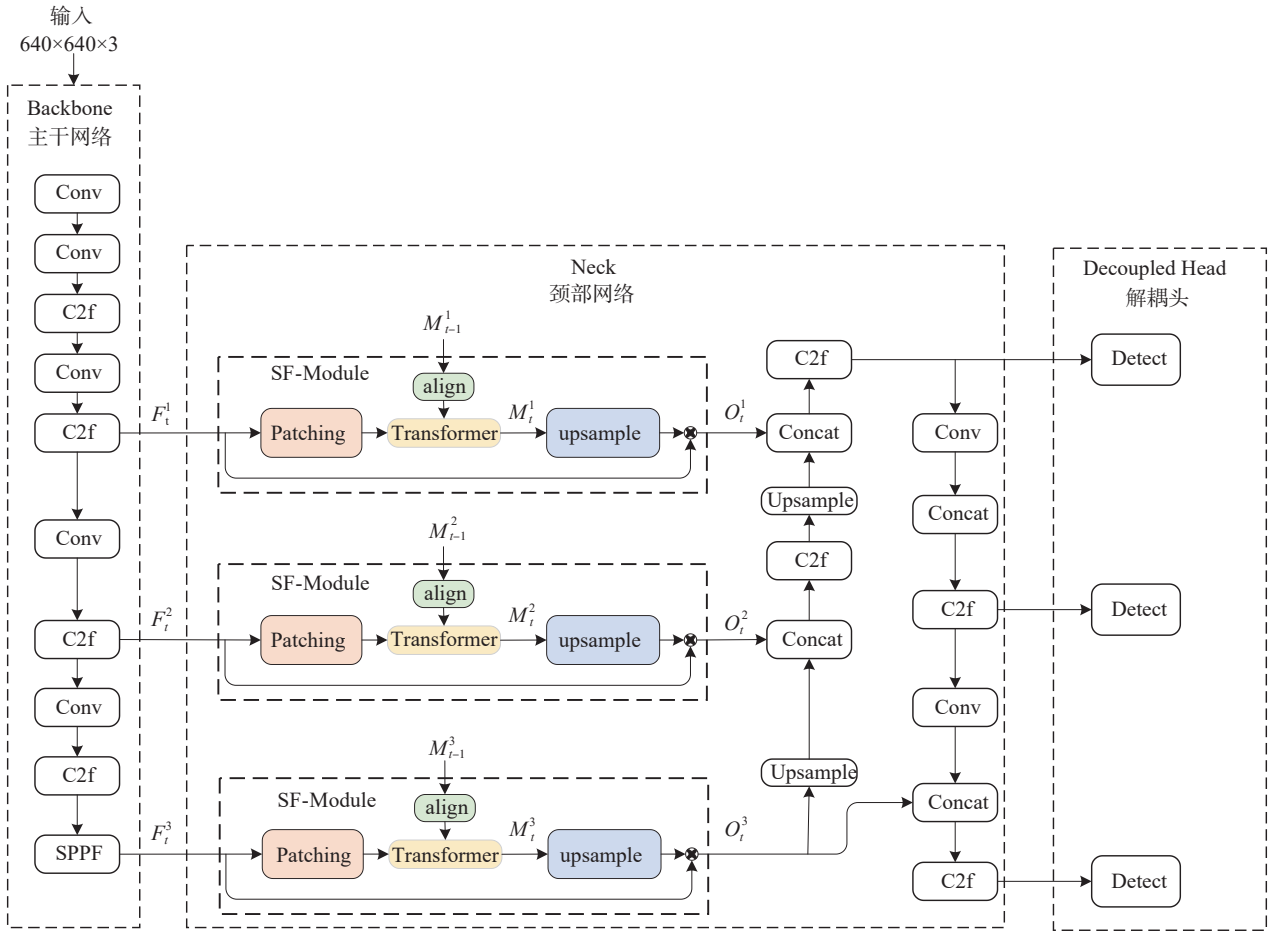


图7 SFW-YOLOv8模型网络结构图

在视频图像目标检测中,连续帧图像之间是包含了大量的时空信息的,关联时空特征可以增强当前帧的特征信息,进而提高检测的准确率。现有的STMN[37]、ConvGRU[39]网络具有捕获视频图像序列间的长时间依赖能力,但不能捕获空间位置上的长距离依赖,缺乏对图像特征的全局理解;Transformer[41]模块可以处理时序数据,具有捕获图像在空间位置上的长距离依赖能力。基于此,本文建立以Transformer网络模块为基础的时空特征融合模块SF-Module(sptio-temporal feature fusion module,SF-Module),SF-Module模块的结构如图8所示。SF-Module包括4个部分:Patching部分、对齐部分、Transformer部分和上采样部分。

(1)Patching部分

Transformer结构能处理特征tocken数据,而SFW-YOLOv8模型主干网络输出的是特征图数据,因此Patching部分实现视频车辆图像特征维度的变化,从特征图数据转化为特征向量数据。

对于给定输入特征图$\boldsymbol{F}_t \in \mathbb{R}^{C\times H\times W}$,首先进行卷积运算(Conv),卷积核大小为$k\times k$,步长为$k$,输出$C\times\frac{H}{k}\times\frac{W}{k}$的特征图;然后将卷积后的特征在第1维和第2维进行展平操作(Flatten),展平后得到$C\times\frac{HW}{k^2}$的特征向量;再将展平后的特征进行转置,最终得到Transformer所能接收的token形式的特征向量$\boldsymbol{X}_t \in \mathbb{R}^{\frac{HW}{k}\times C}$,Patching特征转换公式如式(1)所示。

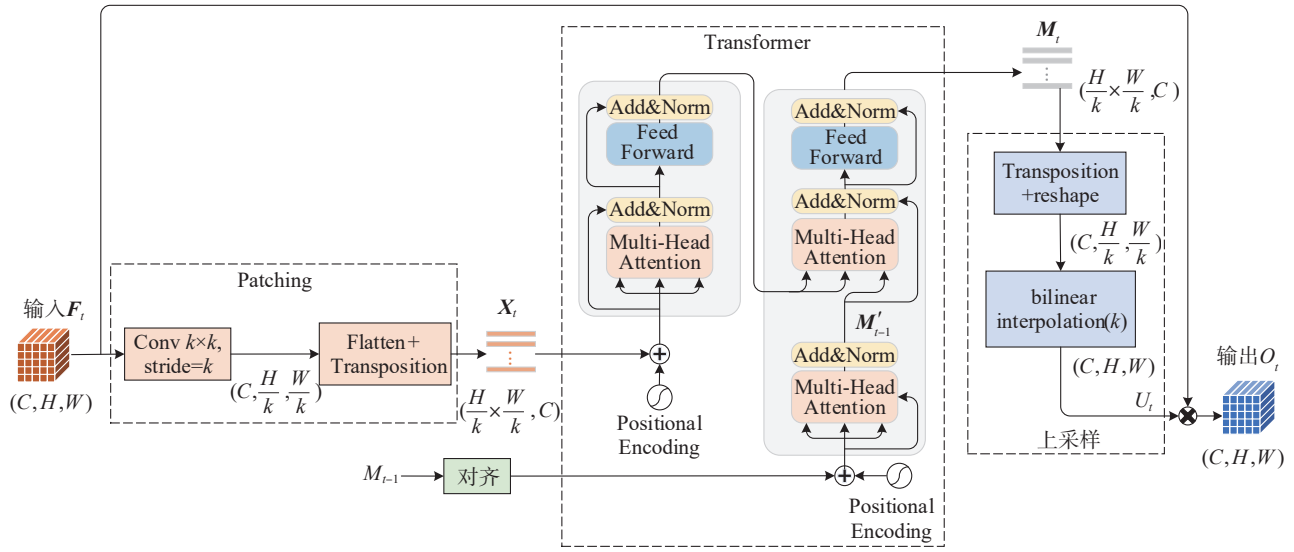


图8　SF-Module时空特征融合模块结构图

$$\boldsymbol{X}_t = Flatten(Conv(\boldsymbol{F}_t))^{\mathrm{T}} \tag{1}$$

(2)对齐部分

视频中车辆目标是运动的,车辆目标在当前帧特征$\boldsymbol{F}_t$上的空间位置和历史帧特征上的空间位置不对齐,最终产生拖尾现象[37]。对齐部分可以实现历史帧特征$\boldsymbol{M}_{t-1}$和当前帧的特征$\boldsymbol{X}_t$在空间位置上对齐,在对齐前需先将$\boldsymbol{X}_t$和$\boldsymbol{M}_{t-1}$恢复成$C\times\frac{H}{k}\times\frac{W}{k}$的维度。对齐过程是通过$\boldsymbol{X}_t$和$\boldsymbol{X}_{t-1}$的位置关系来修正$\boldsymbol{M}_{t-1}$,使其与$\boldsymbol{X}_t$对齐。首先,计算$\boldsymbol{X}_t$在位置$(x,y)$上的特征向量与$\boldsymbol{X}_{t-1}$在位置$(x,y)$周围位置上的特征向量的相似度$C_{x,y}(i,j)$,然后使用该相似度对$\boldsymbol{M}_{t-1}$在位置$(x,y)$上进行加权,最后得到对齐后的$Aligned_\boldsymbol{M}_{t-1}$,相关计算如式(2)和式(3)所示。

$$C_{x,y}(i,j)=\frac{\boldsymbol{F}_t(x,y)\cdot\boldsymbol{F}_{t-1}(x+i,y+j)}{\sum\limits_{i,j\in[-d,d]}\boldsymbol{F}_t(x,y)\cdot\boldsymbol{F}_{t-1}(x+i,y+j)} \tag{2}$$

$$Aligned\_\boldsymbol{M}_{t-1}(x,y)=\sum_{i,j\in[-d,d]}C_{x,y}(x,y)\cdot \boldsymbol{M}_{t-1}(x+i,y+j) \tag{3}$$

式中i、j表示位置(x,y)的附件区域,其限制在范围$[-d,d]$,d越大计算成本也也越大。本文使用的数据集视频帧率为25,相邻帧时间间隔较小,目标偏移不会太大,将d设置为1。

(3)Transformer部分

Transformer部分是SF-Module的核心,其作用就是将当前帧和历史帧的时空特征信息进行多头自注意力计算,进而得到当前帧特征和历史帧特征的相似信息。由图8可知,Transformer部分是由一个TransformerEncoder和一个TransformerDecoder组成。

TransformerEncoder计算过程具体可分为3步,首先,运用可训练的位置编码参数与当前帧特征矩阵$\boldsymbol{X}_t$相加,实现$\boldsymbol{X}_t$的位置编码;其次,经位置编码后的$\boldsymbol{X}_t$再进行多头自注意力计算,具体计算公式如式(4)~式(6)所示;最后,多头注意力计算的结果经残差融合和层归一化输入到全连接层组成前馈网络中,前馈网络的输出再经一次残差融合和层归一化后,最终得到TransformerEncoder的输出。

$$\mathrm{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{4}$$

$$\mathrm{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=Concat(head_1,\cdots,head_h)\boldsymbol{W}^O \tag{5}$$

$$head_i=Attention(\boldsymbol{Q}\boldsymbol{W}_i^Q,\boldsymbol{K}\boldsymbol{W}_i^K,\boldsymbol{V}\boldsymbol{W}_i^V) \tag{6}$$

式中$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$是位置编码后的$\boldsymbol{X}_t$经权重矩阵计算后得到的特征矩阵。多头自注意力就是在自注意力的基础上,将$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$分为多组后再进行自注意力计算(理论上$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$是分别通过不同的$\boldsymbol{W}_i^Q$、$\boldsymbol{W}_i^K$、$\boldsymbol{W}_i^V$线性变换进行分组,实际操作是直接将$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$中每个向量进行均分,本文是分为8组),将得到的结果$head_i$进行拼接,拼接后的特征矩阵再通过一个权重矩阵$\boldsymbol{W}^o$进一步融合。

TransformerDecoder接收历史帧特征信息$\boldsymbol{M}_{t-1}$和TransformerEncoder的输出,计算过程大致可分为

3步。首先,历史帧特征矩阵$\boldsymbol{M}_{t-1}$进行多头注意力计算得到$\boldsymbol{M}'_{t-1}$;其次,将TransformerEncoder的输出经权重矩阵计算后得到$\boldsymbol{Q}$、$\boldsymbol{K}$,而$\boldsymbol{M}'_{t-1}$通过权重矩阵计算后作为$\boldsymbol{V}$,当前帧特征矩阵得到的$\boldsymbol{Q}$和$\boldsymbol{K}$与历史帧特征矩阵得到的$\boldsymbol{V}$进行多头自注意力计算;最后,通过全连接层组成的前馈网络得到整个Transformer部分的最终输出$\boldsymbol{M}_t$,即当前帧特征和历史帧特征之间的时空特征信息。

(4)上采样部分

上采样部分的功能是将Transformer部分输出的时空特征$\boldsymbol{M}_t$由$C\times\frac{HW}{k^2}$的维度恢复成SF-Module输入特征的$C\times H\times W$维度,对后续特征融合时进行特征维度的统一。计算过程可以大致分为两步,首先时空特征$\boldsymbol{M}_t$通过转置和reshape操作输出特征$\boldsymbol{R}_t\in\mathbb{R}^{C\times\frac{HW}{k^2}}$;然后使用双线性插值方法(bilinear interpolation,BI)对特征$\boldsymbol{R}_t$进行k倍上采样,上采样操作后得到的最终输出特征$\boldsymbol{U}_t\in\mathbb{R}^{C\times H\times W}$,计算公式如式(7)所示。

$$\begin{cases}\boldsymbol{R}_t=reshspe(\boldsymbol{M}_t^{\mathrm{T}})\\ \boldsymbol{U}_t=BI(\boldsymbol{R}_t)\end{cases}\tag{7}$$

式中BI表示双线性插值。

双线性插值是一种上采样方法,它是先在横轴上进行两次线性插值计算,然后再在纵轴上进行一次线性插值计算。如图9所示,已知Q_1,Q_2,Q_3,Q_4 4个像素点的坐标$(w_0,h_0),(w_1,h_0),(w_0,h_1),(w_1,h_1)$以及像素值$f(Q_1),f(Q_2),f(Q_3),f(Q_4)$,先通过原图与目标图的比例关系获取需要求的点$P$的坐标$(w,h)$,然后通过$Q_1$、$Q_2$两个点求点$R_0$的像素值$f(R_0)$,再通过$Q_3$、$Q_4$两个点求点$R_1$的像素值$f(R_1)$,最后通过点$R_0$和$R_1$求点$P$的像素值$f(P)$,具体计算过程如式(8)所示。

$$\begin{cases}w=(w\_dst+0.5)\cdot\dfrac{w\_src}{w\_dst}-0.5\\ h=(h\_dst+0.5)\cdot\dfrac{h\_src}{h\_dst}-0.5\\ f(R_0)=\dfrac{w_1-w}{w_1-w_0}\cdot f(Q_1)+\dfrac{w-w_0}{w_1-w_0}\cdot f(Q_2)\\ f(R_1)=\dfrac{w_1-w}{w_1-w_0}\cdot f(Q_3)+\dfrac{w-w_0}{w_1-w_0}\cdot f(Q_4)\\ f(P)=\dfrac{h_1-h}{h_1-h0}\cdot f(R_0)+\dfrac{h-h_0}{h_1-h_0}\cdot f(R_1)\end{cases}\tag{8}$$

式中:h_src、w_src分别表示原图的高和宽;h_dst、w_dst分别表示上采样目标图的高和宽。

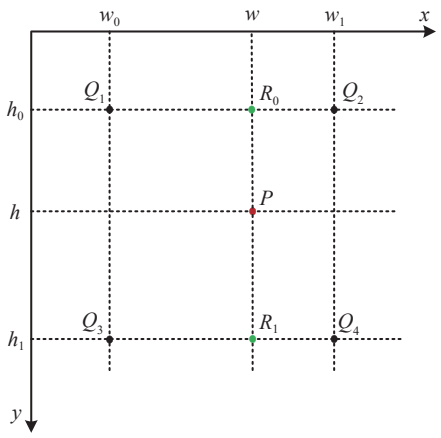


图9 双线性插值示意图

上采样的输出特征$\boldsymbol{U}_t$和TF-Module的输入特征$\boldsymbol{F}_t$进行元素相乘,实现当前帧的特征和历史帧时空特征的进一步融合,最终得到时空融合模块SF-Module的输出$\boldsymbol{O}_t\in\mathbb{R}^{C\times H\times W}$。如图7所示,在SFW-YOLOv8网络中共有3个SF-Module时空特征融合模块,这3个模块分别与主干网络的3条分支输出相连接,从上到下3个SFW-Module模块的Patching部分中的卷积核大小和步长分别对应为4×4, 4; 2×2, 2; 1×1, 1。SFW-Module模块须同时输入当前帧和历史帧的特征信息,即当前帧进行检测时,需要上一帧已经完成检测,所以SFW-YOLOv8在训练时只能设置batch为1,逐帧进行训练。

3 WIoU损失函数

YOLOv8的损失是分类损失和预测框回归损失加权和,其中分类损失使用VFL(varifocal loss)[42]损失函数计算,预测框回归损失使用CIOU损失函数和DFL(distribution focal loss)损失函数计算,总损失$Loss$的具体计算方法如式(9)~式(12)所示。

$$Loss=aVFL+bCIOU+cDFL\tag{9}$$

$$VFL(p,q)=\begin{cases}-q\left(q\log(p)+(1-q)\log(1-p)\right), & q>0\\ -\alpha p^{\gamma}\log(1-p), & q=0\end{cases}\tag{10}$$

$$L_{CIoU}=1-IoU+\frac{\rho^2(b,b^{gt})}{c^2}+\sigma v\tag{11}$$

$$DFL(S_i,S_{i+1})=-((y_{i+1}-y)\log(S_i)+(y-y_i)\log(S_{i+1}))\tag{12}$$

在分类损失函数VFL中:p表示预测的IACS(iou-aware classification score)得分;对于正样本,q为预测框和真实标注框的IoU(交并比),对于负样本,q值为0;γ是一个调制因子,调节该参数抑制负样本的作用;α为可调整的比例因子,用于平衡正负样本之间的损失,防止过度抑制。

在$CIoU$损失函数中,IoU是预测框和真实标注框的交并比,$\rho(b, b^{gt})$表示预测框和真实标注框的中心点之间的欧式距离,c表示能够同时包含预测框和真实标注框的最小外接矩阵的对角线距离,v用来度量预测框和真实标注框相对比例的一致性,σ是一个权重函数。

DFL[43]损失函数是通过对预测框位置进行总体建模来提供更多的信息以及准确的位置预测。

在SFW-YOLOv8的训练数据集中存在着一些未标注或者定位不准的低质量标注框,如图10所示(灰色框为未标注框,红色框为定位不准的标注框,蓝色为正常标注框)。低质量的标注框会导致预测框在训练中盲目拟合,降低模型定位能力。

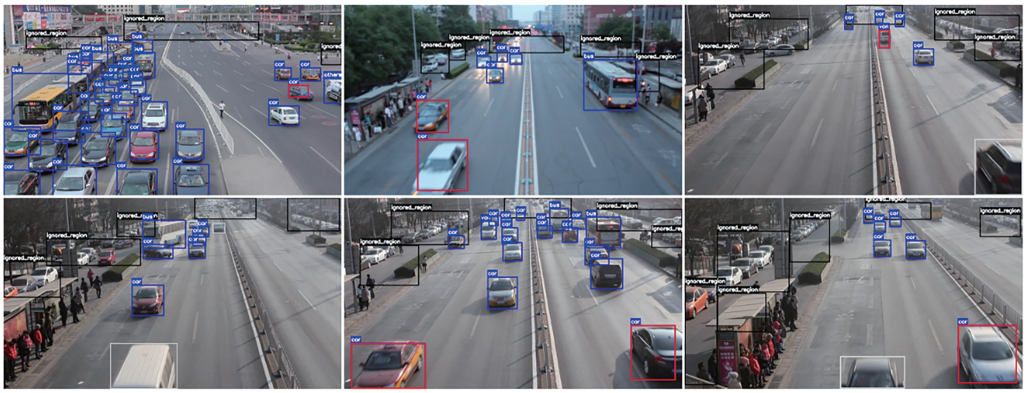


图10　低质量标注框示意图

模型的定位能力受预测框回归损失的直接影响,而预测框回归损失中的CIOU损失会通过距离、长宽比等几何因数加重低质量样本的惩罚,进而无法控制预测框在低质量标注框上的盲目拟合,而WIoU(Wise-IoU)[44]损失可以降低高质量标注框的竞争力,同时也减少低质量示例产生的有害梯度,使模型可以专注于普通质量的标注框,并提高车辆检测模型的整体性能。因此,为了减少低质量标注框对SFW-YOLOv8模型训练的影响,在YOLOv8损失函数的基础上将预测框回归损失中的CIOU损失替换为WIoU损失后作为SFW-YOLOv8的损失函数,WIoU损失的计算公式如式(13)和式(14)所示。

$$L_{\mathrm{WIoU}} = R_{\mathrm{WIoU}} L_{\mathrm{IoU}} \tag{13}$$

$$R_{\mathrm{WIoU}} = \exp\left(\frac{(x - x_{\mathrm{gt}})^2 + (y - y_{\mathrm{gt}})^2}{(W_{\mathrm{g}}^2 + H_{\mathrm{g}}^2)}\right) \tag{14}$$

式中:x和x_{gt}分别表示预测框和真实标注框的横坐标值;y和y_{gt}分别表示预测框和真实标注框的纵坐标值;W_{g}和H_{g}分别表示预测框和真实标注框最小外接矩阵的宽和高;L_{IoU}表示预测框和真实标注框的交并比损失。$R_{\mathrm{WIoU}} \in [1, e)$,将显著放大普通质量标注框的$L_{\mathrm{IoU}}$;$L_{\mathrm{IoU}} \in [0, 1]$,当预测框与标注框重合较好时,将显著减少高质量标注框的R_{WIoU},并减少其对预测框和标注框中心点之间距离的关注。

4　实验结果与分析

4.1　实验环境与车辆检测数据集

本文具体实验环境如表1所示。

表1　实验环境准备

实验环境	环境配置
操作系统	Ubuntu
GPU	NVIDIA GeForce RTX 2080 Ti
显存	12 GB
编程语言	Python3. 7
深度学习框架	PyTorch1. 8. 0

实验使用的车辆检测数据集为UA-DETRAC,数据集中包含使用佳能EOS 550D相机在中国北京和天津的24个不同地点拍摄的10 h视频。视频以每秒25帧(fps)的速度录制,分辨率为960×540像素;数据集中车辆种类分为轿车、公共汽车、客车和其他,训练集为60个视频的82 085张图片,测试集为40个视频的56 167张图片。

4.2 评价指标

算法的评价指标是衡量算法性能优劣的量化指标,通过评价指标可以了解算法是否存在较强的竞争力,同时为算法的进一步优化提供方向。因此,本文选用mAP(mean average precision,mAP)作为评价指标,全面、准确地评估SFW-YOLOv8模型的有效性。其中mAP计算公式如式(15)~式(17)所示。

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + FN} \tag{16}$$

$$mAP = \frac{1}{classes}\sum_{i=1}^{classes}\int_0^1 P(R)\mathrm{d}(R) \tag{17}$$

式(15)和式(16)中,TP(true positive)表示预测结果是真,真实结果也是真的样本,FP(false positive)表示预测结果为真,真实结果为假的样本,FN(false negative)表示预测结果为假,真实结果为真的样本。式(15)中的P表示检测准确率,反映模型识别是否准确,式(16)中的R表示召回率,反映模型是否识别全面。式(17)中,$\int_0^1 P(R)\mathrm{d}(R)$表示以$P$为纵坐标,$R$为横坐标绘制出的$PR$曲线与$R$坐标轴围成区域的面积,该区域面积越大,表明模型性能越好。mAP指标也指所有类别AP的平均值,因此,mAP指标考虑全面,能够全面评估模型在全部类别上的综合性能。本文分别使用$mAP50$和$mAP50:5:95$作为评价指标,其中$mAP50$指预测框和真实框匹配的IoU阈值设为50%所得mAP值;$mAP50:5:95$是IoU阈值取50%~95%,步长为5%,再计算这些IoU下的均值。模型的推理速度用ms/帧表示,即检测每帧图片所需的时间,其值越小说明推理速度越快。

4.3 结果分析

4.3.1 模型训练

本文所有实验模型均在统一的软硬件环境进行训练和测试,SFW-YOLOv8训练前会载入训练后的YOLOv8的权重参数,训练时只训练SF-Module模块的参数,其它参数冻结。YOLOv8训练过程中批处理大小为16,而SFW-YOLOv8为1,训练周期都为100。YOLOv8和SFW-YOLOv8两模型训练时的$mAP50$和$mAP50:5:95$值变换曲线如图11所示。

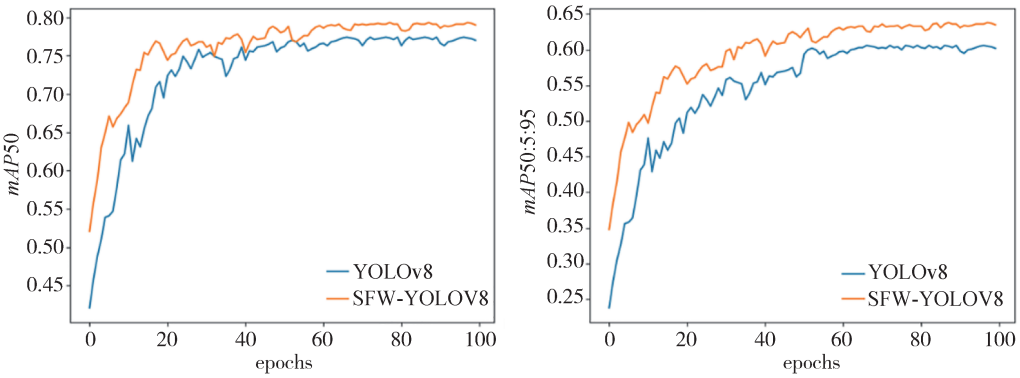


图11 YOLOv8和SFW-YOLOv8训练时mAP值曲线图

如图11所示,蓝色两条曲线为YOLOv8训练时$mAP50$和$mAP50:5:95$值的变换曲线,训练时载入了YOLOv8的权重参数,所以mAP的初始值较大。黄色两条曲线分别为SFW-YOLOv8训练时$mAP50$和$mAP50:5:95$值的变换曲线,由于训练时载入了ImageNet上的预训练权重,所以mAP会有一个不错的初始值,而不是从0附近的值开始上升;由黄蓝曲线对比可知,SFW-YOLOv8在整个训练过程中的mAP值均优于YOLOv8。

4.3.2 消融实验

本文设计了多组消融实验进行测试,验证SFW-YOLOv8中时空特征融合模块SF-Module以及Wise-IoU损失的有效性。

(1)SF-Module模块的有效性实验

将原始YOLOv8和使用convGRU模块的YOLOv8以及SFW-YOLOv8进行对比,验证SF-Module时空融合模块的有效性,对比实验结果如表2所示。

表2 SF-Module的有效性对比

方法	$mAP50$/%	$mAP50:5:95$/%	推理速度/(ms·帧$^{-1}$)
YOLOv8	77.4	60.3	9.3
+convGRU[39]	77.8	61.3	9.8
+SF-Module	**78.2**	**62.4**	**10.9**

如表2所示,在视频图像检测场景下,与原始YOLOv8相比,新增SF-Module模块的YOLOv8虽然*mAP*50值仅提高了0.8%,但*mAP*50:5:95值提高了2.2%,*mAP*50:5:95值的较大提升表明时空特征模块SF-Module可以较好地提高小目标车辆检测的准确率;SF-Module模块与convGRU模块相比,SF-Module模块在融合时空特征信息上的能力更强,在推理时间上的少量增加。

(2)各模块的有效性实验

为了验证SF-Module模块和WIoU损失函数对SFW-YOLOv8性能的影响,本文设计了多组实验。如表3所示,YOLOv8为基准模型,模型A为仅新增SF-Module时空特征融合模块的YOLOv8,模型B为仅使用WIoU损失函数的YOLOv8,模型C为同时新增SF-Module时空特征融合模块并使用WIoU损失函数的YOLOv8。

表3　各模块的有效性实验

方法	SF-Module	WIoU	*mAP*50/%	*mAP*50:5:95/%	推理速度/(ms·帧$^{-1}$)
YOLOv8			77.4	60.3	9.3
A	√		78.2	62.5	10.9
B		√	78.4	61.8	9.4
C	√	√	79.5	63.6	11.2

如表3所示,仅在YOLOv8中使用SF-Module时空特征融合模块的模型A与YOLOv8的对比前文已论述,本处不在赘述;与YOLOv8相比,仅使用WIoU损失函数的模型B在*mAP*50值上提升了1%,在*mAP*50:5:95值上提升了1.5%,推理时间增加了0.1 ms;同时使用SF-Module时空特征融合模块和WIoU损失函数的模型C即SFW-YOLOv8在*mAP*50值上提高了2.1%,在*mAP*50:5:95值上提高了3.3%,推理时间提升了1.9 ms。

由以上模型对比可知,WIoU损失函数可以在训练时减少数据集中低质量标注框的影响,提高车辆检测的准确率并且不改变推理时间(误差范围内),而SF-Module时空特征融合模块的使用,可以丰富图像的特性信息,进一步提高SFW-YOLOv8车辆检测的准确率,尤其是小目标车辆检测的准确率且推理时间只有少量增加。

4.3.3　不同跟踪算法对比实验

为了验证本文SFW-YOLOv8车辆检测算法的性能,将SFW-YOLOv8算法与Faster R-CNN、SSD、RetinaNet、YOLOv5、YOLOX、YOLOv7、YOLOv8这些优秀的检测算法进行实验对比,实验对比结果如表4所示。

表4　不同跟踪算法实验对比结果

方法	*mAP*50/%	*mAP*50:5:95/%	推理速度/(ms·帧$^{-1}$)
Faster R-CNN[8]	67.1	50.1	88.5
SSD[19]	70.2	51.8	53.8
RetinaNet[20]	72.4	53.1	50.4
YOLOv5	76.3	58.1	11.3
YOLOX[24]	75.9	56.9	15.5
YOLOv7[26]	76.7	58.4	10.4
YOLOv8	77.4	60.3	9.3
SFW-YOLOv8	**79.5**	**63.6**	**11.2**

由表4可知,本文新建的SFW-YOLOv8车辆检测模型,其*mAP*50值为79.5%,*mAP*50:5:95值为63.6%,均高于表中其它的检测算法;推理速度为11 ms每帧,推理速度远快于Faster R-CNN、SSD以及RetinaNet,略快于YOLOv5、YOLOX,略慢于YOLOv7和YOLOv8。实验结果表明,与经典的检测算法如Faster R-CNN、SSD等及YOLO系列的检测算法相比,新建的SFW-YOLOv8模型的检测性能具有明显的优势。

4.3.4　真实雨雾场景对比实验

本文在BDD100K数据集中选取多个雨天、雾天的视频组成真实雨雾场景数据集,使用YOLOv8和SFW-YOLOV8在该数据集中进行测试对比,验证在真实雨雾场景中SFW-YOLOV8模型的性能优势,实验对比结果见表5。

表5　BDD100K数据集真实雨雾场景实验对比结果

方法	*mAP*50/%	*mAP*50:5:95/%	推理速度/(ms·帧$^{-1}$)
YOLOv8	76.5	59.2	10.3
SFW-YOLOv8	77.6	61.1	11.9

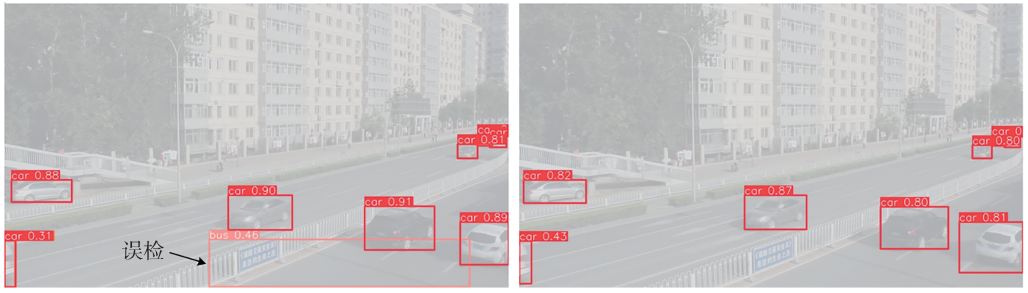
由于BDD100K数据集和UA-DETECT数据集拍摄视角不同,BDD100K数据集视频是行车视角,而UA-DETECT数据集视频是交通监控视角,所以在BDD100K数据集训练的YOLOv8和SFW-YOLOv8模型在准确率上会有所下降(与表4对比),但是在BDD100K数据集真实雨雾场景中本文新建的SFW-YOLOv8车辆检测模型,其*mAP*50值为77.6%,*mAP*50:5:95值为61.1%,均高于YOLOv8检测模型;实验结果表明,与YOLOv8检测算法相比,

本文新建的SFW-YOLOv8模型在真实雨雾场景的检测性能具有明显的优势。

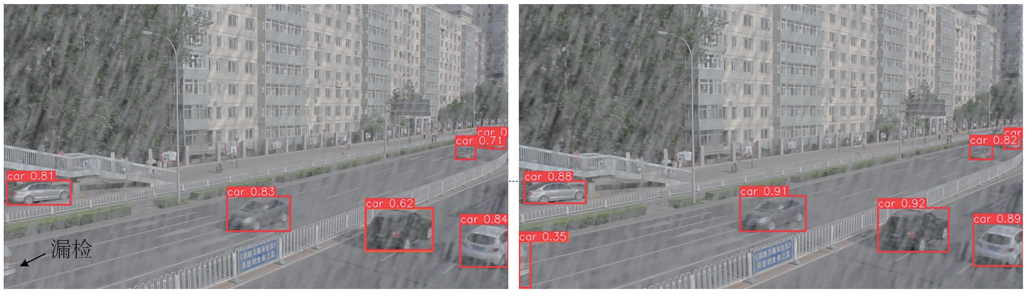
4.3.5 可视化对比

为了直观地展示新建的视频图像车辆检测模型SFW-YOLOv8在恶劣天气情况下的性能,本文将YOLOv8模型和SFW-YOLOv8模型分别在模拟雾天、雨天的视频图像场景中进行可视化对比,结果如图12所示。

(a) 原图



(b) 雾天



(c) 雨天

图12 不同恶劣天气场景可视化结果对比图

图12(a)表示视频图像原图,图12(b)和图12(c)的左侧为YOLOv8模型的检测结果,图12(b)和图12(c)的右侧为SFW-YOLOv8模型的检测结果。SFW-YOLOv8模型在雾天和雨天情况下均正确识别了图像中的车辆,而YOLOv8模型在雾天场景将背景误检为公交车;在雨天场景则有漏检情况。实验结果表明,SFW-YOLOv8模型在雾天、雨天等复杂场景中,相较于YOLOv8模型具有更好的检测性能。

5 结论

(1)本文利用晴天数据,根据模拟算法生成雨天和雾天的数据,提高了雨雾天气视频车辆检测鲁棒性。

(2)本文建立了时空特征融合模块SF-Module,提取了并融合视频车辆图像当前帧和历史帧时空特征信息,并运用Transformer模型中的多头自注意力机制实现了视频车辆图像当前帧和历史帧时空特征信息的提取和融合,丰富了目标的特征信息;另一方面,基于YOLOv8网络,在其颈部网络融合所建立的时空特征融合模块SF-Module,进一步挖掘了视频图像序列的时空特征信息。

(3)鉴于低质量的标注框易导致数据集训练盲目拟合及检测模型的定位能力弱的问题,在

YOLOv8基础上引入WIoU损失函数,减少了低质量标注框产生的有害梯度,提高了SFW-YOLOv8视频车辆检测模型准确度。

参考文献

[1] TSAI D, LAI S. Independent component analysis-based background subtraction for indoor surveillance[J].IEEE Transactions on Image Processing, 2009, 18(1): 158-167.

[2] LEE D S.Effective gaussian mixture learning for video background subtraction[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5): 827-832.

[3] HORN B K P, SCHUNCK B G. Determining optical flow[J].Artificial Intelligence, 1981, 17(1/3): 185-203.

[4] XU Z H, HUANG W Q, WANG W. Multi-category vehicle detection in surveillance video based on deep learning[J]. Journal of Computer Applications, 2019, 39(3): 700-705.

[5] 江岫. 基于改进YOLOv5的车辆检测及跟踪方法研究[D]. 重庆:重庆交通大学,2023.
JIANG Shen. Research on vehicle detection and tracking methods based on improved YOLOv5 [D]. Chongqing: Chongqing Jiaotong University,2023.

[6] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. Computer Vision and Pattern Recognition. USA: IEEE, 2014: 580-587.

[7] HE K, ZHANG X, REN S, 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.

[8] GIRSHICK R. Fast R-CNN[C]. Proceedings of IEEE International Conference on Computer Vision, USA: IEEE, 2015: 1440-1448.

[9] 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.

[10] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 2961-2969.

[11] FAN Q, BROWN L, SMITH J. A closer look at faster R-CNN for vehicle detection[C]. 2016 IEEE intelligent vehicles symposium (IV). IEEE, 2016: 124-129.

[12] XU Y, YU G, WANG Y, et al. Car detection from low-altitude UAV imagery with the faster R-CNN[J]. Journal of Advanced Transportation, 2017, 2017.

[13] 朱茂桃,张鸿翔,方瑞华. 基于RCNN的车辆检测方法研究[J]. 机电工程,2018,35(8):880-885.
ZHU Maotao, ZHANG Hongxiang, FANG Ruihua. Research on vehicle detection method based on RCNN [J]. Mechanical and Electrical Engineering,2018,35(8):880-885.

[14] HSU S C, HUANG C L, CHUANG C H. Vehicle detection using simplified fast R-CNN[C]. 2018 International Workshop on Advanced Image Technology (IWAIT). IEEE, 2018: 1-3.

[15] 陈玉敏,李淼,房晓丽. 基于时空融合加速的Fast RCNN运动车辆检测算法[J]. 电子测量技术,2020,43(3):139-145.
CHEN Yumin, LI Miao, FANG Xiaoli. Fast RCNN moving vehicle detection algorithm based on spatiotemporal fusion acceleration [J]. Electronic Measurement Technology, 2019, 43(3): 139-145.

[16] 李松江,吴宁,王鹏,等. 基于改进Cascade RCNN的车辆目标检测方法[J]. 计算机工程与应用,2021,57(5):123-130.
LI Songjiang, WU Ning, WANG Peng, et al. Vehicle target detection method based on improved cascade RCNN [J]. Computer Engineering and Applications, 2019,57(5):123-130.

[17] 柳杰,金积德,郑庆祥. 基于改进Mask RCNN的夜间车辆检测方法[J]. 交通信息与安全,2023,41(2):59-66.
LIU Jie, JIN Jide, ZHENG Qingxiang. Vehicle detection method at night based on improved mask RCNN [J]. Traffic Information and Safety, 2019,41(2):59-66.

[18] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779-788.

[19] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector[C]. European Conference on Computer Vision. Springer, Cham, 2016: 21-37.

[20] LIN T Y , GOYAL P , GIRSHICK R, et al. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, PP(99): 2999-3007.

[21] REDMON J, FARHADI A. YOLO9000: better, faster, stronger [C]. IEEE Conference on Computer Vision & Pattern Recognition. IEEE, 2017: 6517-6525.

[22] REDMON J, FARHADI A. YOLOv3: an incremental improvement[J]. arXiv e-prints, 2018.

[23] BOCHKOVSKIY A, WANG C Y, LIAO H. YOLOV4: optimal speed and accuracy of object detection[J]. arXiv preprint arXiv: 2004.10934, 2020.

[24] GE Z, LIU S, WANG F, et al. YOLOX: exceeding YOLO series in 2021[J]. arXiv preprint arXiv: 2107.08430, 2021.

[25] LI C, LI L, JIANG H, et al. YOLOv6: a single-stage object detection framework for industrial applications[J]. arXiv preprint arXiv:2209.02976, 2022.

[26] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 7464-7475.

[27] CARION N, MASSA F, SYNNAEVE G, et al. End-to-end object detection with transformers[C]. European Conference on Computer Vision, 2020: 213-229.

[28] 李珣,刘瑶,李鹏飞,等. 基于Darknet框架下YOLO v2算法的车辆多目标检测方法[J]. 交通运输工程学报,2018,18(6): 142-158.
LI Xun, LIU Yao, LI Pengfei, et al. Vehicle multi-target detec-

tion method based on YOLOv2 algorithm in darknet framework [J]. Journal of Traffic and Transportation Engineering, 2018, 18(6): 142-158.

[29] CHEN S, LIN W. Embedded system real-time vehicle detection based on improved YOLO network[C]. 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). IEEE, 2019: 1400-1403.

[30] 徐浩,杨德刚,蒋倩倩,等. 基于SSD的轻量级车辆检测网络改进[J]. 计算机工程与应用,2022,58(12):209-217.
XU Hao, YANG Degang, JIANG Qianqian, et al. Improvement of lightweight vehicle detection network based on SSD [J]. Computer Engineering and Applications, 2022, 58(12): 209-217.

[31] ZHANG Y, GUO Z, WU J, et al. Real-time vehicle detection based on improved YOLOv5[J]. Sustainability, 2022, 14(19): 12274.

[32] 蔡刘畅,杨培峰,张秋仪. 基于YOLOv7的道路监控车辆检测方法[J]. 陕西科技大学学报,2023,41(6):155-161,175.
WANG C Y, BOCHKOVSKIY A, LIAO H Y M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 7464-7475.

[33] 许晓阳,高重阳. 改进YOLOv7-tiny的轻量级红外车辆目标检测算法[J]. 计算机工程与应用,2024,60(1):74-83.
XU Xiaoyang, GAO Chongyang. Improved lightweight infrared vehicle target detection algorithm based on YOLOv7-tiny [J]. Computer Engineering and Applications, 2024, 60(1): 74-83.

[34] 周飞,郭杜杜,王洋,等. 基于改进YOLOv8的交通监控车辆检测算法[J]. 计算机工程与应用,2024,60(6):110-120.
ZHOU Fei, GUO Dudu, WANG Yang, et al. Vehicle detection algorithm for traffic monitoring based on improved YOLOv8 [J]. Computer Engineering and Applications, 2024, 60(6): 110-120.

[35] KANG K, OUYANG W L, LI H S, et al. Object detection from video tubelets with convolutional neural networks[C]. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016: 817-825.

[36] FEICHTENHOFER C, PINZ A, ZISSERMAN A. Detect to track and track to detect[C]. Proceedings of the 2017 IEEE International Conference on Computer Vision, 2017: 3057-3065.

[37] XIAO F Y, LEE Y J. Video object detection with an aligned spatial-temporal memory[C]. Proceedings of the 15th European Conference on Computer Vision, 2018: 494-510.

[38] GONG T, CHEN K, WANG X, et al. Temporal RoI align for video object recognition[J]. arXiv:2109.03495, 2021.

[39] 程稳,陈忠碧,李庆庆,等. 时空特征对齐的多目标跟踪算法[J]. 光电工程,2023,50(6):66-79.
CHENG Wen, CHEN Zhongbi, LI Qingqing, et al. Multi-target tracking algorithm with spatiotemporal feature alignment [J]. Opto-electronic Engineering, 2023, 50(6): 66-79.

[40] WEN L, DU D, CAI Z, et al. UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking[J]. Computer Vision and Image Understanding, 2020, 193: 102907.

[41] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [J]. Advances in Neural Information Processing Systems, 2017, 30.

[42] ZHANG H, WANG Y, DAYOUB F, et al. Varifocalnet: an iou-aware dense object detector[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021: 8514-8523.

[43] LI X, WANG W, WU L, et al. Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection [J]. Advances in Neural Information Processing Systems, 2020, 33: 21002-21012.

[44] TONG Z, CHEN Y, XU Z, et al. Wise-IoU: bounding box regression loss with dynamic focusing mechanism [J]. arXiv preprint arXiv:2301.10051, 2023.