贾仁祥, 单鸿涛. 改进 YOLOv4+DeepSORT 的城市车流量统计[J]. 智能计算机与应用,2025,15(7):11-20. DOI:10.20169/j.issn.2095-2163.250702

改进 YOLOv4+DeepSORT 的城市车流量统计

贾仁祥, 单鸿涛
(上海工程技术大学 电子电气工程学院, 上海 201620)

摘 要: 针对原始 YOLOv4 与 DeepSORT 算法在车流量统计过程中速度较慢的问题,综合考虑实时性与准确率,提出一种改进的 YOLOv4 与 DeepSORT 结合的车流量统计方法。首先,为了提升检测器的速度,采用 Mobilenetv3-CA 替换 YOLOv4 的主干网络,降低模型参数量,提升网络的速度;使用 CDIoU(Control Distance IoU) loss 作为定位损失,使网络预测框与真实框具有更高的重合度;用 Focal loss 改进置信度损失,使模型在训练过程中更好地学习遮挡车辆特征。然后,采用无迹卡尔曼滤波改进 DeepSORT 的运动关联,提升跟踪过程的非线性能力;采用车辆颜色特征替代重识别网络的深度特征作为外观信息,降低了跟踪算法的计算耗时。最后,将改进的 YOLOv4 与 DeepSORT 算法相结合,在视频中设置虚拟检测线进行车流量统计。实验结果表明,改进后的算法在速度上均超过 25 FPS,达到了实时性需求,多个视频的准确率均达到 90%。

关键词: YOLOv4; DeepSORT; 注意力机制; CDIoU; 车流量统计
中图分类号: TP391.4 **文献标志码**: A **文章编号**: 2095-2163(2025)07-0011-10

Improved urban traffic statistics of YOLOv4+DeepSORT

JIA Renxiang, SHAN Hongtao
(School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: Aiming at the slow speed of the original YOLOv4 and DeepSORT algorithms in the process of traffic flow statistics, an improved traffic flow statistics method combining YOLOv4 and DeepSORT was proposed considering the real-time performance and accuracy. Firstly, in order to improve the speed of the detector, MobilenetV3-CA was used to replace the backbone network of YOLOv4 to reduce the number of model parameters and improve the speed of the network. The Control Distance IoU(CDIoU) loss is used as the location loss to make the network prediction box and the real box have higher coincidence degree. Focal loss was used to improve the confidence loss, so that the model could better learn the occluded vehicle features in the training process. Then, unscented Kalman filter is used to improve the motion correlation of DeepSORT to improve the nonlinear capability of the tracking process. The vehicle color feature is used to replace the depth feature of the rerecognition network as the appearance information, which reduces the computation time of the tracking algorithm. Finally, the improved YOLOv4 algorithm is combined with DeepSORT algorithm, and the virtual detection line is set in the video for traffic flow statistics. Experimental results show that the speed of the improved algorithm is more than 25 frames/second, which meets the demand of real-time performance. In terms of accuracy, the accuracy of multiple videos is up to 90%.

Key words: YOLOv4; DeepSORT; attention mechanism; CDIoU; traffic flow statistics

0 引 言

城市道路场景下的车流量统计是交通智能管理与安全监测的关键技术,快速且精准的实现车流量统计,可以合理地分配城市道路资源,提升道路的通行效率,有效降低城市交通拥堵问题。

车流量统计分为车辆目标检测和跟踪两部分,传统的车辆检测方法主要分为提取候选区域、人工选择特征和分类等 3 个阶段。提取候选区域阶段通常采用滑动窗口的方式,从输入图像中获得所有可能包含待检测车辆目标的候选区域;人工选择特征阶段设计合适的特征代替物体,其中典型的代表有 HOG[1]、SIFT[2]、DPM[3];分类阶段能够将提取的特征进行分类,HOG 特征与支持向量机(Support

基金项目: 国家自然科学基金(61673257)。
作者简介: 贾仁祥(1996—),男,硕士,主要研究方向:计算机视觉与机器学习。
通信作者: 单鸿涛(1971—),女,博士,副教授,主要研究方向:计算机视觉。Email:shanhongtao@sues.edu.cn。
收稿日期: 2023-11-13

Vector Machine, SVM)的结合在车辆检测中具有广泛的应用[4-6]。然而,传统方法容易受到背景与天气的干扰,存在鲁棒性差,适用性弱等缺陷。近年来,随着深度学习技术的发展,目标检测算法已从基于人工选择特征的传统算法转向了基于深度神经网络的检测技术。

基于深度学习的目标检测模型可以分为两大类:

1)双阶段检测算法,该算法的典型代表是基于候选框的 R-CNN 系算法,如 R-CNN[7]、Fast R-CNN[8]、Faster R-CNN[9]等,其将检测问题划分为两个阶段。首先产生候选区域,然后对候选区域分类,该类算法虽然有不错的精度,但速度较慢。

2)单阶段检测算法,比较典型的算法如 SSD[10]和 YOLO[11-14]。此类算法不需要产生候选区域,直接产生物体的类别概率和位置坐标值,相较于双阶段算法,在速度上具有较大的提升,但准确率较低。

如果仅使用检测算法对车辆进行统计,会存在前后帧重复车辆多次计算的问题,需要采用跟踪算法确定前后帧车辆的身份统一。SORT[15]算法是一种基于检测的跟踪方式,采用卡尔曼滤波与匈牙利匹配算法对目标进行跟踪,算法具有较高的速度,但是无法解决长期遮挡和目标识别的问题。DeepSORT[16]是 SORT 算法的改进版,增加了重识别网络提取特征,降低了跟踪过程中目标身份切换的问题,提升了跟踪的精度。

为了提升车流量统计的速度,综合考虑精度与速度,本文采用 YOLOv4 作为目标检测算法的基本框架,结合 DeepSORT 方法实现车辆的跟踪与统计。针对 YOLOv4 模型改进主要包括 3 部分:首先采用 Mobilenetv3-CA[17-18]替代 YOLOv4 的主干网络,降低模型的复杂度,提升网络的运行速率;其次将 CDIoU[19]损失函数作为定位损失,使边界框回归具有更高的重合度;最后采用 Focal loss[20]改进置信度损失函数,可以更好的检测遮挡车辆,降低模型的漏检率。针对 DeepSORT 算法,采用无迹卡尔曼滤波作为运动关联,增加车辆跟踪的非线性能力,同时使用颜色特征替代重识别网络的深度特征,提升跟踪的速度。

1 检测网络结构

1.1 YOLOv4-Mobilenetv3 网络

YOLOv4-Mobilenetv3 网络将原 YOLOv4 的主干特征提取网络 CSPDarknet53 替换为 Mobilenetv3 网络。YOLOv4-Mobilenetv3 网络结构主要包括输入(Input)、主干特征提取网络(Mobilenetv3)、空间金字塔池化(Space Pyramid Pool, SPP)、路径聚合网络(Path Aggregation Network, PANet)与输出(Head),网络结构如图 1 所示。

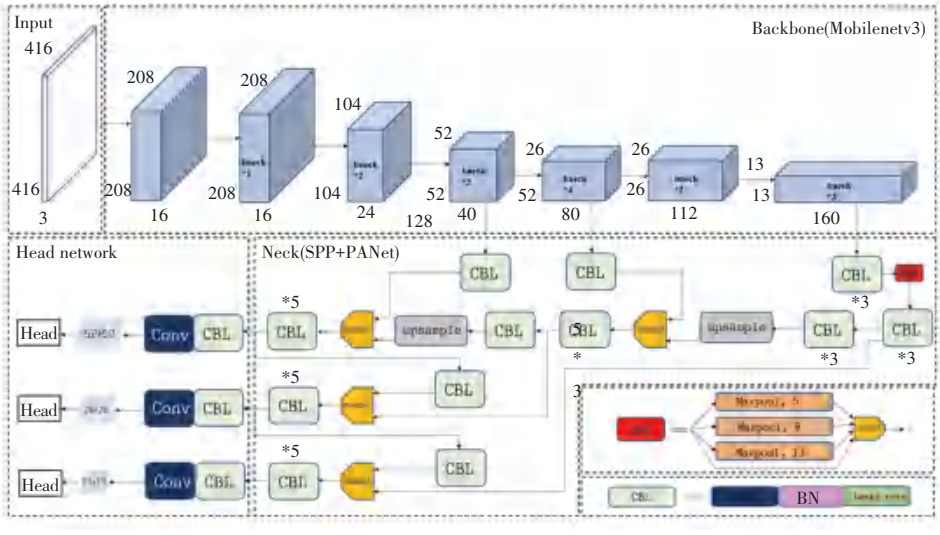


图 1 YOLOv4-Mobilenetv3 网络整体框架

Fig. 1 YOLOv4-MobileNetv3 overall network framework

Mobilenetv3 主要由 bneck 模块构成,bneck 是一种特殊的深度可分离卷积模块,主要由反残差连接结构、深度可分离卷积和(Squeeze and Exciatation, SE)通道注意力机制构成。反残差连接结构通过 1×1 卷积提升通道的维度,降低深度可分离卷积过程中通道数过低带来的特征丢失情况。通道维数提升之后,深度可分离卷积通过将标准卷积拆分为深度卷积与逐点卷积,可以极大的减少网络的参数量,降低了时间

与空间复杂度，提升了网络的运行速率。网络中加入的 SE 通道注意力机制，可以动态的调整通道权重，提高特征的整体表达能力。由于 Mobilenetv3 网络在检测速度和精度上都有较好的表现，因此所提算法选取 Mobilenetv3 作为主干特征提取网络。

特征增强阶段采用了 SPP 模块和 PANet 结构。SPP 网络对特征层进行了 1×1、5×5、9×9 和 13×13 4 种尺度的最大池化（Maxpooling）操作，经过 SPP 后可以有效提升网络的感受野，并提取出显著的上下文特征。PANet 是对特征金字塔网络（Feature Pyramid Network，FPN）的进一步改进，通过将不同特征层的特征经过上采样与下采样进行特征融合，提升了网络的预测能力。

输出阶段对 3 个不同大小的特征层进行结果预测，分别检测小、中、大 3 种目标。与 YOLOv3 原理相同，首先判断不同特征层的先验框是否包含目标与目标类型，然后经过非极大值抑制处理与先验框调整，最后获得相应的预测框。

1.2　损失函数

YOLOv4 算法的损失函数由定位损失、置信度损失和分类损失 3 部分组成，其中定位损失采用 CIoU（Complete Intersection over Union）损失，置信度损失和分类损失采用交叉熵损失，总损失函数如下：

$$Loss = \sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[1 - IoU + \frac{\rho^2(b,b^{gt})}{c^2} + \alpha\gamma\right] - \sum_{i=0}^{S^2}\sum_{j=0}^{B}\left(I_{ij}^{obj} + l_{noobj}I_{ij}^{noobi}\right)\left[\hat{C}_i^j\log(C_i^j) + (1 - \hat{C}_i^j)\log(1 - C_i^j)\right] - \sum_{i=0}^{S^2} I_{ij}^{obj}\sum_{c\in class}\left[\hat{P}_i^j\log(P_i^j) + (1 - \hat{P}_i^j)\log(1 - P_i^j)\right] \quad (1)$$

$$\alpha = \frac{\gamma}{(1 - IoU) + \gamma} \quad (2)$$

$$\gamma = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \quad (3)$$

式（1）中：α 为权重参数，γ 用于度量横纵比的相似性，计算式分别为式（2）、（3），S^2 为网格的数量，B 为单个网格中先验框的数量，IoU、$\rho^2(b,b^{gt})$ 分别表示预测框与真实框的交并比和中心点的欧式距离，I_{ij}^{obj}、I_{ij}^{noobi} 分别表示预测网格中是否有目标，c 为预测框和真实框的最小外接矩形框的对角线距离，l_{noobj} 用于平衡正负样本的权重参数，$\hat{C}_i^j$、C_i^j 和 $\hat{P}_i^j$、P_i^j 分别为真实框和预测框的置信度和类别概率，w、h 和 b 分别表示预测框的宽、高和中心点坐标，w^{gt}、h^{gt} 和 b^{gt} 表示真实框的宽、高和中心点坐标。

1.3　CA 注意力机制

Mobilenetv3 网络中采用 SE 通道注意力，只是在通道上增加了权重，而忽略了位置信息。CA（Coordinate Attention）注意力模块通过在通道注意力中嵌入位置信息，增大了网络模型的感受野的同时避免了大量计算开销，而位置信息对于视觉任务中捕获空间结构具有重要作用。CA 注意力机制如图 2 所示。

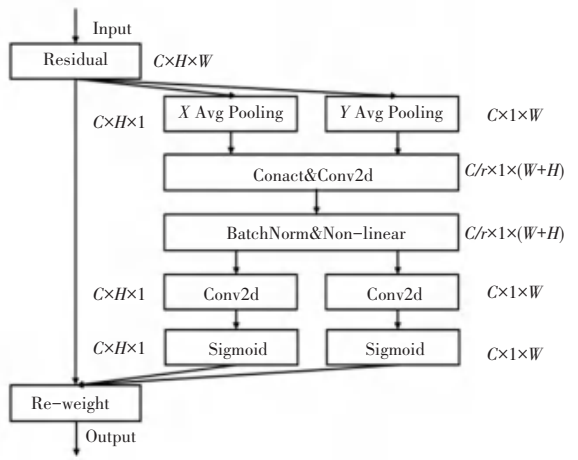


图 2　CA 注意力机制

Fig. 2　CA attention mechanism

SE 模块对输入的每个特征通道先进行一个全局的平均池化，再经过两个全连接层后，使用 Sigmoid 函数来生成通道权重。与 SE 模块中将特征张量转化为单个特征向量不同，CA 通过 X、Y 两个方向的全局平均池化，分别沿着水平和垂直两个方向聚合特征，得到一对方向感知的特征图。两个注意力图中的每个元素都反映了感兴趣的对象是否存在于相应的行和列中，从而可以更准确地定位感兴趣对象的位置信息，更好地识别整个模型。

1.4　定位损失改进

原始 YOLOv4 算法中，定位损失采用的是 $CIoU$ 损失函数，公式如下：

$$L_{loc} = CIoU = 1 - IoU + \frac{\rho^2(b,b^{gt})}{c^2} + \alpha\gamma \quad (4)$$

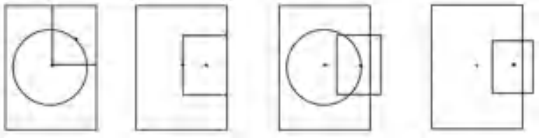
$CIoU$ 虽然同时考虑了交并比、中心点距离和横纵比一致性，能够很好的评价两框的重合与相似程度，但是 $CIoU$ 函数本身还是存在着如下问题[21-22]：

1）由于 γ 反应的是目标框之间横纵比的差异，并不是直接反应了预测框与真实框的大小关系。当出现 $w = kw^{gt}$、$h = kh^{gt}$ 时 $\gamma = 0$，这与现实不符。

2）由式（3）对 w 和 h 分别求偏导，得出 $\frac{\partial\gamma}{\partial w} =$

$-\frac{h}{w}\frac{\partial\gamma}{\partial h}$，可以发现两个式子符号相反。所以，当其中一个变量增加时，另一个变量会相应减少。

3）如图 3 所示，在某些特殊情况下，*CIoU* 无法准确衡量预测框与真实框的重合程度。



（a）*CIoU* = 0.81 （b）*CIoU* = 0.8 （c）*CIoU* = 0.98 （d）*CIoU* = 0.81

图 3 相对位置不同对 *CIoU* 的影响

Fig. 3 Influence of different relative positions on *CIoU*

上述图中，圆点为各矩形的中心点，较大的为真实框，较小的为预测框，4 幅图中的预测框与真实框均相等，可以求解出 *CIoU*（b）< *CIoU*（a）< *CIoU*（c）。如（d）所示，*CIoU*（c）中的预测框一直向真实框的中心点移动，存在一点使得 *CIoU*（a）与 *CIoU*（d）相等。然而（d）的 *IoU* 值要小于（a），因此 *CIoU* 的惩罚项无法有效的反应预测框与真实框之间的重合程度。

为了解决这些问题，采用了一种新的高效的损失函数，直接采用预测框与真实框的顶点距离与外接矩形框的对角线长度的比值作为惩罚项（如图 4），具体的损失函数如下：

$$CDIoU = 1 - IoU + \frac{\| RP - GT \|_2}{4 \times c} = 1 - IoU + \frac{AE + BF + CG + DH}{4WY} \quad (5)$$

其中，*RP* 表示预测框 EFGH；*GT* 表示真实框 ABCD；*c* 表示最小外接矩形框对角线距离 *WY*。

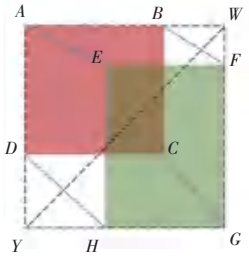


图 4 *CDIoU* 示意图

Fig. 4 *CDIoU* schematic

改进的定位损失函数由于直接采用了顶点距离作为惩罚项，在回归过程中可以使预测框更加迅速的向着真实框的方向进行回归。相对于 *CIoU* 计算反三角函数，*CDIoU* 直接计算顶点距离，可以有效的减小计算量。

1.5 置信度损失改进

在城市道路场景中，汽车之间会发生频繁的遮挡现象，导致检测过程中出现漏检与错检的问题。针对该问题，采用 Focal loss 改进置信度损失。在网络训练过程中，Focal loss 通过调整交叉熵损失（Cross-Entropy Loss）增加权重，解决了正负样本不平衡以及简单样本与困难样本不平衡的问题。YOLOv4 算法中，当预测框与真实框的 *IoU* 大于 0.5 时，判定为正样本，小于 0.5 判定为负样本。困难样本指的是分类不明确的预测框，处在前景与背景的过渡区域。简单样本指的是与前景没有重叠区域的负样本，或者与前景具有很高重合度的正样本。正负样本与难易样本如图 5 所示。

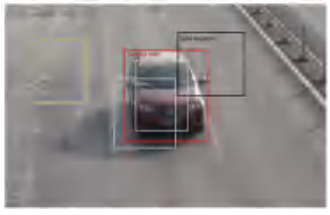


图 5 正负样本与难易样本示意图

Fig. 5 Schematic diagram of positive and negative samples and difficult and easy samples

Focal loss 是在交叉熵损失的基础上进行改进的，交叉熵损失函数如下：

$$CE = y\log y' - (1-y)\log(1-y') = \begin{cases} -\log y', & y = 1 \\ -\log(1-y'), & y = 0 \end{cases} \quad (6)$$

其中，$y' \in [0,1]$ 为目标的置信度，y 为类别标签，取值为 0 或 1。为了解决样本不平衡问题，使损失函数学习更多的有用信息，Focal loss 针对原始交叉熵损失函数进行了改进，公式如下：

$$CE = \begin{cases} -\beta(1-y')^{\lambda}\log y', & y = 1 \\ -(1-\beta)y'^{\lambda}\log(1-y'), & y = 0 \end{cases} \quad (7)$$

其中，β 是权重系数，通常取 0.25，用来平衡正负样本数量；λ 是调节参数，值设为 2，用来降低简单样本的损失，使得模型训练过程中更关注于困难样本[20]。

YOLOv4 的损失函数中本身包括正负样本的平衡权重，所以本文改进时只采用控制难易样本的超参数进行改进，改进后的置信度损失如下：

$$L_{conf} = -\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[(1-C_i^j)^{\lambda}\hat{C}_i^j\log(C_i^j) + (C_i^j)^{\lambda}(1-\hat{C}_i^j)\log(1-C_i^j)\right] - \sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{noobj} I_{ij}^{noobi}\left[(1-C_i^j)^{\lambda}\hat{C}_i^j\log(C_i^j) + (C_i^j)^{\lambda}(1-\hat{C}_i^j)\log(1-C_i^j)\right] \quad (8)$$

2　DeepSORT 车辆跟踪

SORT 算法是在检测结果的基础上利用卡尔曼滤波和匈牙利匹配算法实现目标跟踪。DeepSORT 算法是在 SORT 算法的基础上进行改进，额外引入了目标的外观信息进行匹配计算，使用了级联匹配对更加频繁出现的目标赋予优先匹配权，进一步提升了整体跟踪性能。

2.1　DeepSORT 算法原理

2.1.1　状态估计与跟踪处理

在 DeepSORT 算法中，先通过检测器实现目标检测，并采用了一个八维向量 $X=(u,v,r,h,\dot{x},\dot{y},\dot{r},\dot{h})$ 来描述目标的运动状态。其中，(u,v) 表示目标框的中心点的横坐标和纵坐标，r 表示目标框的横纵比，h 表示目标框的高，$(\dot{x},\dot{y},\dot{r},\dot{h})$ 表示在图像坐标系中所对应的相对速度。利用基于线性观测模型和等速运动模型的卡尔曼滤波器，对当前帧的目标状态进行预测与更新，预测结果表示为 (u,v,r,h)；如果检测框没有与之相关联的轨迹，则简单的预测其状态，不进行状态更新。

在跟踪轨迹的处理方面，每条跟踪轨迹称为一个 tracker，轨迹产生的过程是检测的结果与已经存在的跟踪器没有关联匹配成功的，则认为有可能产生了新的轨迹，此时标为不确定态。当连续三帧内新轨迹的预测结果都能与检测结果正确关联，方可认定新轨迹的出现，此时标为确定态。轨迹的消除是当检测的目标和跟踪的目标连续匹配的帧数超过设定的阈值时，则认为此目标的跟踪已经结束，此轨迹消除标为删除态。

2.1.2　运动关联与外观关联

DeepSORT 算法使用目标的运动信息和外观信息实现检测结果和跟踪轨迹之间的匹配。目标的运动信息是利用跟踪器中运动目标状态的卡尔曼预测结果和当前帧的检测结果求取马氏距离进行关联，马氏距离又称为协方差距离，是一种有效计算两个未知样本集相似度的方法，在此用于度量轨迹和检测框的匹配程度，公式如下：

$$d^{(1)}(i,j)=(d_j-y_i)^{\mathrm{T}}S_i^{-1}(d_j-y_i) \tag{9}$$

式中：d_j 表示第 j 个检测框的位置，y_i 表示为第 i 个跟踪器对目标的预测位置，S_i^{-1} 表示检测位置和平均跟踪位置之间的协方差矩阵。如果某次关联的马氏距离小于指定的阈值 $t^{(1)}$，则设置运动状态的关联成功，公式如下：

$$b_{i,j}^{(1)}=1[d^{(1)}(i,j)\leqslant t^{(1)}] \tag{10}$$

当运动不确定性很高时，仅仅使用马氏距离进行关联度量经常会失效，在图像空间中使用卡尔曼滤波进行运动状态估计只是一个比较粗糙的预测，容易造成目标 ID 频繁切换的情况。DeepSORT 算法采用外观信息作为第二种关联方法，对每个检测框通过重识别(Re-Identification，ReID)网络计算对应的 128 维特征向量，并保存计算出来的特征向量。计算跟踪器最近的 100 个成功关联的检测框和当前帧检测结果的特征向量的最小余弦距离，公式如下：

$$d^{(2)}(i,j)=\min\{1-\boldsymbol{r}_j^{\mathrm{T}}\boldsymbol{r}_k^{(i)}\mid\boldsymbol{r}_k^{(i)}\in R_i\} \tag{11}$$

式中：$\boldsymbol{r}_j$ 表示检测框的特征向量，$\boldsymbol{r}_k^i$ 表示跟踪器保存的第 i 特征向量。同样的，外观信息的余弦距离小于阈值 $t^{(2)}$，则认为匹配成功，公式如下：

$$b_{i,j}^{(2)}=1[d^{(2)}(i,j)\leqslant t^{(2)}] \tag{12}$$

最终的综合匹配度：

$$c_{i,j}=\lambda d^{(1)}(i,j)+(1-\lambda)d^{(2)}(i,j) \tag{13}$$

式中：λ 是一个超参数，在相机运动幅度较大时为 0，此时忽略目标的运动信息，偏向于使用目标的外观信息进行关联。

2.2　无迹卡尔曼滤波

DeepSORT 算法中采用标准卡尔曼滤波对目标检测框的位置进行预测与更新，但标准卡尔曼滤波器本身是基于等速运动和线性观测模型。在实际的城市道路场景中，道路交通较为复杂，汽车通常会有较大的车速变化，尤其在路口，车辆停止与启动的车速变化迅速。车速的剧烈变化会导致卡尔曼滤波预测的不准确度增加，对此本文采用无迹卡尔曼滤波(Unscented Kalman Filter，UKF)算法，该算法对非线性具有更强的鲁棒性。

给定非线性离散系统：

$$\begin{cases}\boldsymbol{X}_k=f(X_{k-1})+w_{k-1}\\ \boldsymbol{Z}_k=\boldsymbol{H}_kX_k+v_k\end{cases} \tag{14}$$

式中：$\boldsymbol{X}_k\in R^n$，$\boldsymbol{Z}_k\in R^m$ 分别表示 k 时刻的状态向量和观测向量，$f(\cdot)$ 为非线性函数，$\boldsymbol{H}_k$ 为观测矩阵；w_{k-1} 表示 $k-1$ 时刻的系统噪声，v_k 为 k 时刻的观测噪声，两种噪声为相互独立的高斯白噪声。

UKF 可以分为初始化、无迹变换、时间更新和测量更新等步骤：

1)初始化，求状态变量的均值和方差，公式如下：

$$\begin{cases}k=0\\ \hat{X}_0=E(X_0)\\ P_0=E[(X_0-\hat{X}_0)(X_0-\hat{X}_0)^{\mathrm{T}}]\end{cases} \tag{15}$$

2)无迹变换。$(2n+1)$ 个 Sigma 点，均值权值

ω_m^i 和协方差权值 ω_c^i 的计算公式如下：

$$\begin{cases}\chi_0 = \bar{x}, & i = 0\\ \chi_i = \bar{x} + [\sqrt{(n+\lambda)P_x}]_i, & i = 1, 2, \cdots, n\\ \chi_i = \bar{x} - [\sqrt{(n+\lambda)P_x}]_{i-n}, & i = n+1, n+2, \cdots, 2n\\ \omega_m^o = \dfrac{\lambda}{n+\lambda}\\ \omega_c^0 = \dfrac{\lambda}{n+\lambda} + 1 - a + \beta\\ \omega_m^i = \omega_c^i = \dfrac{1}{2(n+\lambda)}\\ \lambda = a^2(n+\kappa) - n\end{cases} \tag{16}$$

式中：n 表示采样点的个数；λ 表示比例参数，可以调整采样点到均值的整体距离大小，从而用来调整预测误差；a 的大小关系到采样点在原函数周围的分布，通常取值是一个很小的正数；κ 表示待选参数，一般选为 0 或 $3-n$，且要求保证 $(n+\lambda)P$ 是半正定矩阵。

3）时间更新

$$\begin{cases}\hat{X}_{k|k-1} = \sum_0^{2n} \omega_m^i \chi_{k|k-1}^i\\ P_{k|k-1} = \sum_0^{2n} \omega_c^i (\chi_{k|k-1}^i - \hat{X}_{k|k-1})(\chi_{k|k-1}^i - \hat{X}_{k|k-1})^{\mathrm{T}} + Q_k\\ \chi_{k|k-1}^z = [\hat{X}_{k|k-1} \hat{X}_{k|k-1} + \sqrt{(n+\lambda)P_x}\hat{X}_{k|k-1} + \sqrt{(n+\lambda)P_x}]\\ Z_{k|k-1} = H_k \chi_{k|k-1}^z\\ \hat{Z}_{k|k-1} = \sum_{i=0}^{2n} \omega_m^i Z_{k|k-1}^i\end{cases} \tag{17}$$

4）测量更新

$$\begin{cases}P_{\hat{X}_{k|k-1}\hat{Z}_{k|k-1}} = \sum_{i=0}^{2n} \omega_c^i (\chi_{k|k-1}^i - \hat{X}_{k|k-1})(Z_{k|k-1}^i - \hat{Z}_{k|k-1})^{\mathrm{T}}\\ P_{\hat{Z}_{k|k-1}} = \sum_{i=0}^{2n} \omega_c^i (Z_{k|k-1}^i - \hat{Z}_{k|k-1})(Z_{k|k-1}^i - \hat{Z}_{k|k-1})^{\mathrm{T}} + R_k\\ K_k = P_{\hat{X}_{k|k-1}\hat{Z}_{k|k-1}} P_{\hat{Z}_{k|k-1}}^{-1}\\ \hat{X}_k = \hat{X}_{k|k-1} + K_k(Z_k - \hat{Z}_{k|k-1})\\ P_k = P_{k|k-1} - K_k P_{\hat{Z}_{k|k-1}} K_k^{\mathrm{T}}\end{cases} \tag{18}$$

2.3 颜色直方图

DeepSORT 算法中采用的重识别网络借用了行人重识别领域的网络模型，该数据集主要针对行人重识别任务。行人的外观与车辆外观存在着较大差异，就颜色而言，相同车型的车辆颜色差异要小于行人的颜色差异；在形状上，车辆属于刚性物体，行人属于非刚性物体；在监控视角下，车辆的形变要小于行人的形变。基于以上考虑，本文采用颜色直方图（Color Histogram，CH）作为目标的外观信息，取代了重识别网络提取出的外观信息，降低了网络的运行复杂度，更适用于监控场景下的车辆跟踪。

采用 RGB 颜色空间进行计算，先将 3 通道的像素进行归一化处理后，计算目标区域的颜色直方图。颜色直方图定义如下：

$$\begin{cases}H_C(k) = \sum_{X=0}^{M-1}\sum_{y=0}^{N-1} h(C(x,y)), & k = 0,1,\cdots,K\\ h(C(x,y)) = \begin{cases}1, & \text{if } C(x,y) \text{ 在变换空间量化后等于 } k\\ 0, & \text{other}\end{cases}\end{cases} \tag{19}$$

式中：$C(x,y)$ 为 RGB 空间的彩色图像，M、N 为图像垂直和水平上的像素数目，K 为变换空间的颜色数。

在得到目标区域的颜色直方图后，将直方图映射为向量形式，采用 DeepSORT 算法中的最小余弦距离，计算检测器与跟踪轨迹的外观相似度，如式（11）所示。同样，如式（12），外观信息的余弦距离小于阈值，则认为匹配成功。

3 实验结果与分析

实验平台配置：Windows 10 操作系统，Intel（R）Core i7 10700F CPU，NVIDIA Geforce RTX 3070 显卡，软件环境为 CUDA11. 0，Cudnn8. 0，采用 Pytorch 深度学习框架。

3.1 数据集及预处理

实验采用 UA-DETRAC 公开数据集，该数据集主要拍摄于北京与天津的 24 个不同道路过街天桥。测试集包含 40 个视频序列，训练集包含 60 个视频序列，其中手动标注了 8 250 辆车和 121 万目标对象边界框。因原始数据集的数据格式无法在 YOLO 算法中直接使用，则通过 python 脚本将数据格式转换成 PASCAL VOC 格式，并将车辆标签统一为 car。由于原始数据属于视频数据，相邻帧间的信息差异很小，所以采用每隔十帧选取一张图片的方式对数据集进行划分，可以相应减少训练与测试时间。

3.2 评价指标

为衡量模型性能，采用目标检测中常用平均精度均值（mean Average Precision，*mAP*）与每秒检测帧数（Frames Per Second，FPS）作为评价指标。*mAP*

的计算如下：

$$mAP = \frac{1}{n}\sum_{1}^{n} AP(i) \tag{20}$$

式中：n 为类别数，AP 为平均精度（Average Precision），定义为不同召回率下精确率的均值，用于评价样本中某一类的检测精度，表达如下：

$$AP = \int_{0}^{1} P(R)\,dR \tag{21}$$

式中：P 表示精准率，指所有检测出的目标中检测正确的概率；R 表示召回率，指所有正样本中正确检测的概率，计算如下：

$$P = \frac{TP}{TP + FP} \tag{22}$$

$$R = \frac{TP}{TP + FN} \tag{23}$$

式中：TP 指预测正确的正样本数，FP 指预测为正样本但实际为负样本的数量，FN 指预测为负样本但实际为正样本的数量。

3.3　结果与分析

3.3.1　车辆检测实验

为测试改进的 YOLOv4 网络在车辆检测中的表现，将改进算法与 Faster－RCNN、SSD、YOLOv3、YOLOv4 等经典目标检测网络模型，以及 YOLOv4－Mobilenetv3、YOLOv4－tiny 等轻量级检测网络模型进行训练。在 UA－DETRAC 数据集中，对比各算法在 *mAP*、*FPS* 和参数量等不同评价指标，对比结果见表 1。

表 1　不同检测算法的性能对比

Table 1　Performance comparison of different detection algorithms

模型	*mAP*/ %	FPS	模型容量
Faster－RCNN	74.59	18.21	110.00
SSD	71.50	75.20	92.70
YOLOv3	76.20	43.10	242.20
YOLOv4	79.15	45.20	250.20
YOLOv4－tiny	69.70	85.00	23.16
YOLOv4－Ghostnet	70.50	69.50	43.70
YOLOv4－Mobilenetv3	74.78	56.40	55.30
Improved YOLOv4	78.64	51.40	50.30

由表 1 可以看出，在 *mAP* 值指标的比较上，原始 YOLOv4 检测算法的 *mAP* 值最高为 79.15%，相比改进的算法仅高出了 0.51%，但是改进的算法比 YOLOv4 的网络模型减小到了原来的 20.1%，速度快了 13.7%。改进的 YOLOv4 算法相较于 Faster－RCNN、YOLOv3 等经典算法来看，在各项评价指标上都要优于经典算法。与 SSD、YOLOv4－tiny、YOLOv4－ Ghostnet 和 YOLOv4－Mobilenetv3 等算法相比，虽然在速度上有所下降，但在精度上分别提升了 7.14%、8.94%、8.14%和 3.86%。

为了更直观的体现不同改进策略对模型的影响，本文进行了消融实验，实验结果见表 2。

表 2　不同改进策略的性能比较

Table 2　Performance comparison of different improvement strategies

YOLOv4 Mobilenetv3	Focal loss	*CDIoU*	CA	*mAP*/ %	FPS	模型容量
√				74.78	56.4	55.3
√	√			76.78	56.1	55.3
√	√	√		77.44	56.8	55.3
√	√	√	√	78.64	51.4	50.3

由表 2 中可以看出，Focal loss 与 CDIoU 属于训练阶段的损失函数，在预测阶段对于模型的大小与速度几乎没有影响，同时可以将 *mAP* 提升 2.66%。在增加 CA 注意力机制之后，FPS 降低了，但是在精度上提升了 1.2%，模型容量下降了 5 M。

相较于传统的 YOLOv4 算法，本文改进的算法在 *mAP* 降低 0.51% 的前提下，检测速度提升了 13.7%，模型大小减少了 79.9%。综合考虑，本文改进后的模型兼顾了实时性与准确性，更符合城市道路场景下的需求，对于嵌入式设备也具有一定的兼容性。

改进后不同场景下的检测效果对比如图 6 所示。图 6（a）、（c）、（e）、（g）为 YOLOv4 采用 Mobilenetv3 网络改进后的结果，其中包含了多视角、多种光照强度下的照片，并且车辆都存在不同程度的遮挡；图 6（b）、（d）、（f）、（h）为包含 3 种改进策略后的结果，为了方便对比，本文将改进前后不同的检测结果采用白色椭圆虚线框标出。由图 6（a）~（d）可以看出，在两种光照条件下，改进后的算法对远处遮挡的小目标具有更好的检测效果；在图 6（e）、（f）中，两辆出租车具有相似的结构特征，原始算法只检测出了几个矩形框，改进后的算法可以分别检测出两辆车；图 6（g）、6（h）中，改进前的算法的检测框只包含了部分目标，而改进后的算法在目标矩形框的回归过程中更贴合目标的大小，检测出了更大的目标范围。

通过对比可以看出，改进后的算法可以有效提升对遮挡车辆的检测性能，同时提升了边界框与目标的重合度。

(a) 改进前白天场景 (b) 改进后白天场景

(c) 改进前黑夜场景 (d) 改进后黑夜场景

(e) 改进前遮挡场景 (f) 改进后遮挡场景

(g) 改进前预测框重合度 (h) 改进后预测框重合度

图 6 车辆检测可视化结果对比图

Fig. 6 Comparison of visual results of vehicle detection

3.3.2 改进 DeepSORT 算法实验

实验中将无迹卡尔曼滤波融入 DeepSORT 算法,在 UA-DETRAC 测试集中选取视频进行测试,结果见表 3。

表 3 车辆跟踪实验对比

Table 3 Vehicle tracking experiment comparison

模型	IDs	FPS
SORT[15]	95	45.0
DeepSORT[16]	53	21.8
DeepSORT+CH	57	29.4
DeepSORT+UKF	41	20.5
DeepSORT+CH+UKF	43	28.5

由于 SORT 算法仅使用了目标的运动信息进行关联,在上述数据的车辆跟踪中发生了 95 次身份切换,DeepSORT 相比 SORT 算法引入了外观信息,身份切换降低了 44%。在引入了 CH 和 UKF 等改进措施后,相较于原始 DeepSORT 算法进一步降低了 IDs,速度也提升了 30.7%,在本地测试平台上的速度基本满足了实时性的需求。可视化结果如图 7 所示,ID 为 33 的车辆在第 282 帧图像检测跟踪到了目标,但是在 285 帧受到了大型车辆的遮挡导致目标丢失,在 307 帧有继续跟踪到了 ID 为 33 的车辆。目标跟踪在遮挡过程中容易丢失目标,但是当目标再次出现时,可以再次找回丢失的目标,从而降低了目标 ID 变换的问题。

（a）跟踪视频第 282 帧图像　　（b）跟踪视频第 285 帧图像　　（c）跟踪视频第 307 帧图像

图 7　跟踪效果展示图

Fig. 7　Tracking effect display diagram

3.3.3　车流量统计实验

本文数据是从 UA-DETRAC 数据集中挑选的视频图像序列，并未参与目标检测网络的训练，用这些数据进行车流量检测的测试。此外，增加了部分自制数据集作为补充。自制数据集包括在天桥拍摄视频序列与在网络上寻找的相关视频构成，共计 7 段视频序列作为测试数据，且数据覆盖了晴天、阴天、白天和黑夜等多种场景，相对来说具有一定代表性。为了对车流量统计的准确性进行比较，首先需要人工统计视频中车辆的实际数目，将其作为真实的数量，用算法统计的结果与真实数量进行比较。本文对改进前后的算法加载相同的测试数据测试并对比，实验数据相关信息与算法改进前后的实验对比见表 4。

表 4　车流量统计结果

Table 4　Traffic flow detection results

视频序号	实际车流量	算法改进前			算法改进后		
		算法统计量	准确率/%	FPS	算法统计量	准确率/%	FPS
1	54	50	92.50	21.5	52	96.20	28.4
2	75	71	94.60	21.3	72	96.00	28.5
3	33	27	81.80	21.8	30	90.90	29.4
4	220	192	87.27	20.5	203	92.20	27.5
5	110	91	82.70	20.9	102	92.70	27.9
6	80	70	87.50	21.3	74	92.50	28.2
7	21	18	85.70	21.8	20	95.20	29.1

表中算法统计的车流量分别包括上行与下行的数量，并进行加和，准确率是算法统计的车流量与实际车流量的比值。

分析表数据可知，改进后的算法在精度上均达到了 90% 以上，准确率较高，在所测试数据的精度上要优于改进前的算法。虽然改进后算法在检测阶段的精度略低于改进前的算法，但在跟踪阶段改进了目标外观信息和运动信息的代价矩阵，降低了目标身份切换的次数，从而提升了车流量统计的准确性。改进后的算法在速度上相较于改进前的算法提升了 30%左右，网络视频的帧率通常在 25~30 FPS 之间，因此改进后的算法可以满足实时检测。

本文对部分视频进行了可视化处理，其效果如图 8 所示。

图 8　车流量检测可视化结果

Fig. 8　Visualization results of traffic flow detection

4　结束语

针对城市道路车流量统计的实时性与轻量化的需求，采用 YOLOv4 作为检测器，结合 DeepSORT 算法进行车流量统计。采用 Mobilenetv3-CA 改进 YOLOv4 的主干网络，提升了网络的检测效率；为了

降低精度的损失,采用 Focal loss 与 CDIoU 对损失函数进行改进;将 DeepSORT 算法与无迹卡尔曼滤波相结合,降低车辆非线性运动导致的漏跟踪;颜色特征的引入降低了 DeepSORT 算法的计算复杂度。实验结果表明,本文算法在所选测试视频中均达到了 90%的准确率,平均速度超过 25 FPS,满足了实时性的需求。

参考文献

[1] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection [C]//Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Piscataway,NJ:IEEE, 2005: 886-893.

[2] LOWE D G. Object recognition from local scale - invariant features. Int[J]. Journal of Computer Vision, 2004, 60(2): 91-110.

[3] 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.

[4] 李星, 郭晓松, 郭君斌. 基于 HOG 特征和 SVM 的前向车辆识别方法[J]. 计算机科学, 2013, 40(S2):329-332.

[5] 王全, 王长元, 穆静, 等. 车辆行车实时目标区域特征提取及分类训练[J]. 西安工业大学学报, 2015, 35(11):888-892.

[6] 凌永国, 胡维平. 复杂背景下的车辆检测[J]. 计算机工程与设计, 2016, 37(6):1573-1578.

[7] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway,NJ:IEEE,2014: 580-587.

[8] GIRSHICK R. Fast R - CNN [C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway, NJ: 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 and Machine Intelligence, 2016, 39(6): 1137-1149.

[10] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]// Proceedings of European Conference on Computer Vision. 2016: 21-37.

[11] 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. Piscataway, NJ:IEEE, 2016: 779-788.

[12] REDMON J, FARHADI A. YOLO9000: Better, faster, stronger [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 7263-7271.

[13] REDMON J, FARHADI A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv,1804.02767, 2018.

[14] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv, 2004.10934, 2020.

[15] BEWLEY A, GE Z, OTT L, et al. Simple online and realtime tracking [C]// Proceedings of 2016 IEEE International Conference on Image Processing (ICIP). Piscataway,NJ:IEEE, 2016: 3464-3468.

[16] WOJKE N, BEWLEY A, PAULUS D. Simple online and realtime tracking with a deep association metric [C]// Proceedings of 2017 IEEE International Conference on Image Processing (ICIP). Piscataway,NJ:IEEE, 2017: 3645-3649.

[17] HOWARD A, SANDLER M, CHU G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ: IEEE, 2019: 1314-1324.

[18] HOU Q, ZHOU D, FENG J. Coordinate attention for efficient mobile network design [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway,NJ: IEEE, 2021: 13713-13722.

[19] CHEN D, MIAO D. Control distance IoU and control distance iou loss function for better bounding box regression[J]. arXiv preprint arXiv,2103.11696, 2021.

[20] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020 , 42(2): 318-327.

[21] ZHANG Y F, REN W, ZHANG Z, et al. Focal and efficient IOU loss for accurate bounding box regression [J]. arXiv preprint arXiv,2101.08158, 2021.

[22] 高明华, 杨璨. 基于改进卷积神经网络的交通目标检测方法[J]. 吉林大学学报(工学版), 2022, 52(6): 1353-1361.