

融合 HC-FPN 与 ICRU 的机载视觉 滑行道线检测方法

章涛, 付豪, 罗其俊, 王壮, 谢向镗

(中国民航大学 天津市智能信号与图像处理重点实验室, 天津 300300)

摘要: 为通过机载视觉传感器获取飞机滑行方向与滑行道线的相对位置关系, 增强民航飞机滑行阶段机组人员对滑行道线的感知能力, 以实现民航飞机视觉辅助驾驶, 提出了一种融合 HC-FPN 与 ICRU 的机载视觉滑行道线检测方法; 通过 ResNet 进行图像特征提取, 在网络颈部设计了 HC-FPN 融合多级高层信息; 通过实例检测, 并采用行锚检测与动态卷积预测滑行道线形状; 此外设计了 ICRU 满足具有复杂拓扑结构的滑行道线检测; 在损失中引入了 PIoU 损失函数; 经过实验测试, 该网络在自制滑行道线数据集上 F_1 达到 85.81%, 在 CurveLanes 数据集和 CULane 数据集上 F_1 达到 86.17% 和 78.63%, 在 TuSimple 数据集上准确率达到 96.45%, 实验结果证实了该方法的准确率和有效性, 有助于飞机滑行过程中机组人员辅助驾驶。

关键词: 滑行道线检测; 行锚检测; 特征融合; 拓扑结构; 机载视觉

A Taxiway Line Detection Method for Onboard Vision Based on the Fusion of HC-FPN and ICRU

ZHANG Tao, FU Hao, LUO Qijun, WANG Zhuang, XIE Xiangle

(Tianjin Key Laboratory of Intelligent Signal and Image Processing, Civil Aviation University of China, Tianjin 300300, China)

Abstract: To acquire the relative positional relationship between the taxiing direction of an aircraft and taxiway markings through airborne visual sensors, and to enhance the crew's perception of taxiway markings during the taxiing phase of civil aviation aircraft, thereby achieving visual-assisted driving for such aircraft, an airborne vision taxiway line detection method integrating HC-FPN and ICRU is proposed. Extract image features through the ResNet, design the HC-FPN in the network neck to fuse multi-level high-level information, perform instance detection, and predict the shapes of taxiway lines by using row-anchor detection and dynamic convolution. Additionally, design the ICRU to meet the requirements for detecting taxiway lines with complex topological structures, and introduce the PIoU loss function in the loss calculation. Experimental results show that the proposed network achieves an F_1 score of 85.81% on a self-developed taxiway line dataset, 86.17% on the CurveLanes dataset, 78.63% on the CULane dataset, and an accuracy of 96.45% on the TuSimple dataset, which verifies the accuracy and effectiveness of the proposed method, with an assisted driving for crew members during aircraft taxiing.

Keywords: taxiway line detection; anchor-based detection; feature fusion; topological structure; onboard vision

0 引言

在民用航空领域, 飞机在机场滑行阶段的安全至关重要。目前机场已经部署了包括场面监视雷达和多点定位系统等在内的多种场面监视技术, 用于监视机场上的

各种目标。但在滑行阶段, 飞机主要依靠机场地面交通管制员通过语音通信方式为机组提供指令和位置信息。机组人员在接受到指令后根据机场地图以及滑行道标志将飞机滑行到指定位置。然而, 若语音指令过于复杂或沟通出现误解, 可能导致机组人员滑行操作失误, 严重

收稿日期:2024-12-21; 修回日期:2025-02-17。

基金项目:国家重点研发计划(2022YFB3904303)。

作者简介:章涛(1980-),男,博士,教授。

引用格式:章涛,付豪,罗其俊,等.融合 HC-FPN 与 ICRU 的机载视觉滑行道线检测方法[J].计算机测量与控制,2026,34(1):42-50.

时可能导致飞机发生碰撞事故。因此, 为了提高飞机滑行阶段的安全性, 为机组人员提供直观可靠的辅助决策信息显得尤为重要。通过利用机载视觉传感器, 获取飞机滑行方向与滑行道中线的偏离角度, 这一信息对于机组人员具有极高的参考价值。此外, 滑行道线也是飞机在机场移动期间飞机引导系统的关键技术^[1-2]。

目前, 道线检测方法的研究主要聚焦在公路车道线检测领域, 而针对机场滑行道线检测的研究相对较少。与车道线检测相比较, 滑行道线存在大量的密集和分叉曲线, 其中滑行道边线是两条相邻的平行曲线, 在滑行道出口处滑行道中线以分叉的形式存在, 并且在滑行道中线分叉处的检测至关重要, 该信息有助于机组人员在滑行道出口处引导飞机滑行过程中更轻松做出准确的决策。在机场跑道中线及滑行道线检测领域, 空客公司已开展了基于机器视觉的智能辅助驾驶系统研究与验证工作^[3], 该系统利用机载视觉传感器为机组人员提供了辅助滑行信息。相较于机场滑行道线检测, 国内外学者对于公路车道线的检测进行了较为深入的研究, 这些研究方法主要可以分为传统方法和深度学习方法两类。传统的车道线检测方法通过手工提取不同环境中车道线的颜色、边缘或梯度变化等特征^[4-5], 利用霍夫变换^[6-7]、最小二乘法^[8]等后处理来拟合车道线形状。然而, 传统方法无法在实际场景中保持鲁棒性, 因为手工设计的模型难以适应不同场景中车道线的多样性^[9]。随着深度学习的发展, 一些基于深度学习的方法^[10]在车道检测中展现出显著优势, 能够从图像信息中提取车道线特征, 具有较高的准确性和鲁棒性。基于深度学习的车道线检测方法主要分为基于分割的检测算法^[11-13]、基于参数的检测算法^[14-15]、基于线锚的检测算法^[16-18]和基于行锚的检测算法^[19-20]。

基于分割的检测算法将车道线检测视为像素点的集合并进行分类问题, 将场景图像中的每个像素分为车道线区域或背景区域。相较于一般的语义分割任务, 车道检测需要实例区分。SCNN^[11]在分割模块中使用了特殊的卷积操作, 通过空间卷积处理切片特征并逐个相加, 聚合不同维度的信息。Curvelane-NAS^[12]使用神经网络结构搜索来寻找更好的网络以捕获准确的车道线信息, 以利于检测曲线车道。RESA^[13]在垂直和水平方向上切分特征图, 然后使每个切片接收与特定步长相邻的另一个切片特征。通过循环移动特征图的方式, 最终每个位置都能汇聚整个空间的信息, 从而实现特征图中像素间信息的有效传播。但基于分割的检测方法受限于分割逐像素的检测, 该过程需要大量的时间, 处理速度较慢。

基于参数的检测算法思想将车道线建立一种合适的模型, 然后利用这些参数进行回归来检测车道线。PolyLaneNet^[14]采用多项式回归, 首次提出使用深度网

络回归车道曲线方程, 利用卷积神经网络输出从左到右顺序的固定数量车道线的多项式系数、垂直偏移和置信度分数。LSTR^[15]将 Transformer^[22]引入车道线检测任务中, 通过得到车道线的全局特征, 直接解码图像中每条车道线的车道形状模型参数。但在复杂的车道线场景下, 基于参数的检测方法会导致预测精度低, 并且该方法对预测参数比较敏感, 难以实现很高的检测精度。

基于线锚的检测算法利用预定义的线锚作为参考来回归车道线。Line-CNN^[16]最先采用线锚的方法检测车道线。LaneATT^[17]使用不同位置的直线和不同旋转角度作为锚点, 并提出了一种全新的基于线锚的注意力机制来聚合全局信息。CLRNet^[18]使用均匀间隔的点作为线锚, 并设计了 ROIgather 模块来获取全局上下文信息。虽然预定义线锚可以减少无视觉问题对车道线检测的影响, 但是预定义的线锚限制了车道线预测时的自由度。

基于行锚的检测算法充分利用车道线的形状先验信息, 将图像网格划分, 对等间距的每一行网格预测车道线可能的单元格。UFLD^[19]首先提出了一种基于行锚的车道线检测方法, 该方法在效率上取得了显著的成功。CondLaneNet^[20]将车道线的起始位置特征作为卷积核参数, 并将卷积核参数与整个特征图进行卷积以检测车道线。虽然基于行锚的检测算法可以简化模型的输出, 提高检测的实时性, 但同时也会损失一定的准确度。

为解决分叉和密集场景下滑行道线检测困难的问题, 提出了一种滑行道线检测方法。设计高层融合特征金字塔 (HC-FPN, high-level capability-feature fusion pyramid network) 实现多级融合高层信息, 利用 Transformer 编码融合全局信息以及通道注意力模块保留有效数据的同时去掉多余的数据实现特征选取, 并利用双线性插值和通道注意力模块进行特征融合, 以提高模型对形状细长且弯曲的滑行道线检测能力。同时, 设计实例卷积循环单元 (ICRU, instance convolutional recurrent unit) 对每条滑行道线输出状态向量和卷积核参数向量, 分别表示预测的起始点包含多少条曲线和用于对应曲线形状预测的参数, 解决检测密集和分叉滑行道线的困难。最后, 在基于行锚的检测中, 引入点交并比 (PIoU, point intersection over union) 损失函数, 对每条滑行道线终止位置进行约束, 避免密集场景中预测出的一条曲线包含不同实例信息的情况, 有利于提高检测精度。

1 滑行道线检测框架

由于滑行道线与车道线在结构上有着强的相似性, 并且在检测实时性上有一定的实际需求, 因此, 本文在基于行的车道线检测算法思想下展开机载视觉滑行道线检测, 主要可由 3 部分组成: 1) 数据集采集, 构建机

载视觉滑行道线数据集,用于模型训练、测试和实验评估;2)根据滑行道线的特点,在基于行的检测算法的思想上将所提出的模块相结合,得到滑行道线检测算法;3)在自建数据集以及3个主流的车道线数据集上对所提算法与其他经典算法进行对比实验、可行性实验和消融实验,通过实验数据得到实验结论。

根据行锚的车道线检测方法的思想,对于给定的一张图像 $I \in R^{C \times H \times W}$,目的是预测滑行道线 $L = \{l_1, l_2, \dots, l_N\}$ 的集合,其中 N 是滑行道线的总数,并且每一条滑行道线 l_i 由许多有序的坐标表示,如式(1)所示:

$$l_i = [(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \dots, (x_{iN_i}, y_{iN_i})] \quad (1)$$

其中: i 表示第*i*条滑行道线, N_i 表示第*i*滑行道线可以取到的最大点数。

整体结构如图1所示,本文采用ResNet^[23]作为主干网络。为了提供足够的多层全局信息,保证每条滑行道线能够聚合到该实例的所有信息,以HC-FPN作为网络的颈部。在滑行道线检测过程中,首先根据实例分割方法,通过预测每条滑行道线的起始点来检测滑行道线实例。然后根据动态卷积核参数的卷积运算^[24-25],为每个实例输出一组卷积核参数。为了能够检测具有分叉和密集的滑行道线,利用ICRU对每个检测实例循环输出状态向量和卷积核参数向量,直到状态向量不在输出有效值,所以ICRU可以解决具有复杂拓扑结构的滑行道线检测。

最后根据每个实例的卷积核参数与逐行公式预测每条滑行道线的逐行位置、垂直范围和偏移映射来预测每条滑行道线的具体形状。

1.1 高层融合特征金字塔

机场滑行道线具有曲线细长且弯曲的形状特点,并且滑行道边线是非常密集的两条平行的曲线,在滑行道出口处还存在分叉的曲线,这样的曲线特征增加了模型对滑行道线的检测难度。

为了应对滑行道线细长弯曲的形状特征,提高在实例检测和形状预测的精度,提出了HC-FPN实现多层信息融合。HC-FPN的结构如图2所示,主要由特征选取模块和特征融合模块两部分组成。首先Transformer编码器对主干网络输出的最后一层特征进行全局信息融合,其他不同尺度的特征利用通道注意力(CA, channel attention)模块进行信息筛选。然后利用选择特征聚合(SFF, selective feature fusion)模块,将特征图中的高层和低层信息进行有效的融合。通过这种方式使每条滑行道线能够融合该实例的全局特征信息,相较于传统的FPN^[26]仅通过自顶向下和横向连接将低分辨率、语义强的特征与高分辨率、语义弱的特征结合在一起相比,更有利于模型的检测能力。

1.1.1 特征选取模块

由于滑行道线在整个图像中仅占小部分区域,大部

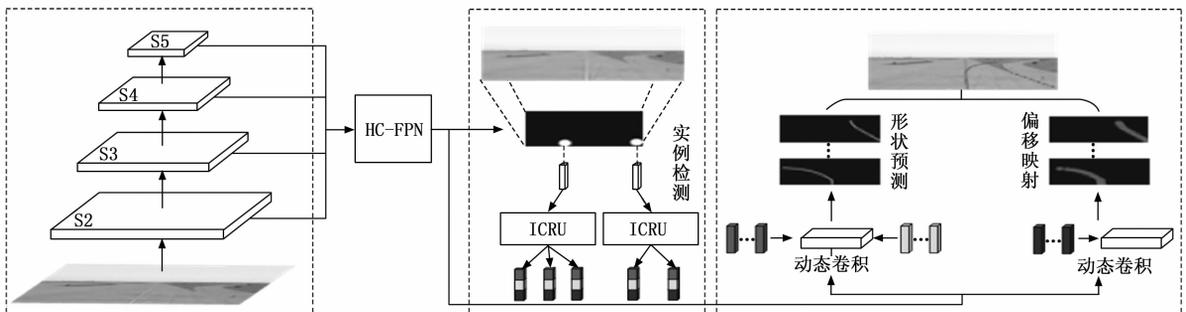


图1 网络整体结构

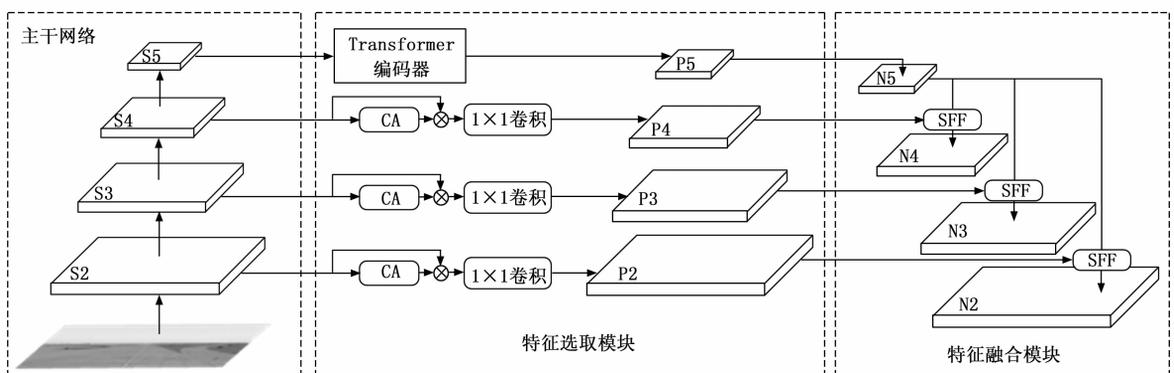


图2 高层融合特征金字塔网络

分区域为无关信息, 因此对于滑行道线无关冗余信息的剔除非常有必要, 利用 Transformer 编码器和 CA 模块实现特征选取。在主干网络的最后一层, 引入了 Transformer 编码器进行全局特征信息融合, 由于传统的卷积在对图像进行卷积运算的过程主要是对空间局部特征信息提取, 很难提取图像的全局信息。因此为了更准确地通过实例起始点预测每条滑行道线的具体形状, 融合全局信息至关重要。此外, 在实例检测过程中, 应对密集和分叉曲线情况下, 需要强大的全局特征信息融合能力确保对检测的起始点预测多条曲线, 否则不利于滑行道线的检测, Transformer 编码器结构如图 3 所示。

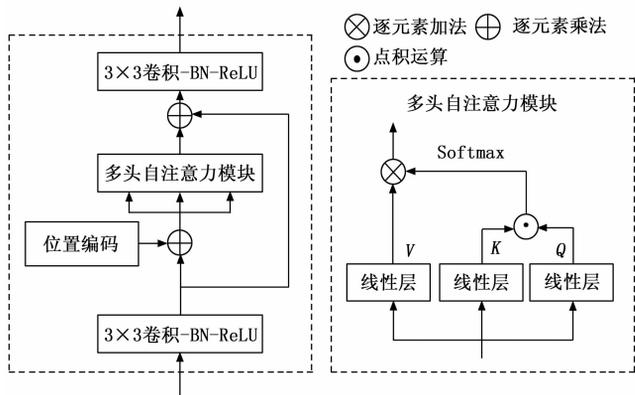


图 3 Transformer 编码器

主干网络输出的其它 3 个特征图经过 CA 模块。在 CA 模块中, 每个特征图分别经过最大池化层和平均池化层, 随后将这两个池化层的结果进行相加, 并将相加后的结果经过 Sigmoid 激活函数, 得到每个通道的权重。最后将得到的权重信息与对应的特征图相乘, 从而得到滤波后的特征图。其中最大池化层的目的是从特征图每个通道的数据中选取最相关的数据, 而平均池化层的目的是从特征图每个通道中均匀的获取所有数据, 以最大程度的保留数据。通过将两个池化层的结果相加, 既可以从通道中选取最具有代表性的数据, 又可以最大限度的减小有效数据的损失。为了进行特征融合, 需要对滤波后的特征图进行 1×1 卷积运算。

1.1.2 特征融合模块

主干网络能够生成具有不同空间分辨率的特征图。其中高分辨率的特征图具有低级特征, 而低分辨率的特征图具有较强的语义特征。通过简单将高分辨的特征与低分辨率的特征相加, 可以增加每层的语义信息。但对于具有细长且形状弯曲的滑行道线, 如果只是简单像素值的相加, 这会导致后续预测的结果较低。因此提出 SFF 模块, 首先利用双线性插值对 Transformer 编码器输出的高级特征进行处理, 得到与其他 3 个低级特征相同维度的高级特征。然后利用 CA 模块计算高级特征的

权重, 并将权重与每个低级特征相乘, 最后在与高级特征相加实现特征融合, 从而避免简单像素相加, 这样可以增强模型的特征表示能力。

1.2 实例卷积循环单元

机场滑行道边线以相邻的形式存在, 并且在滑行道出口处滑行道中线会出现分叉的情况。所以在检测每条滑行道线实例的过程中, 由于相邻和分叉的影响, 导致在检测实例起始点中多条曲线仅检测出一条曲线, 从而严重影响检测性能。因此, 为了解决滑行道线相邻以及分叉情况下检测效果不足的问题, 本文提出 ICRU, 结构如图 4 所示。首先对输入进行卷积运算, 为了避免信息丢失以及出现梯度消失和网络退化问题, 使用残差连接方式。然后将得到的信息经过式 (2) 和式 (3) 运算过程得到相应输出:

$$C_i = C_{i-1} \times \sigma(f_i) + [1 - \sigma(f_i)] \quad (2)$$

$$h_i = \sigma(c_i) \times \tanh C_i + [1 - \sigma(c_i)] \quad (3)$$

其中: σ 表示 sigmoid 激活函数, \tanh 表示 tanh 激活函数, C_{i-1} 表示上一次的输出。

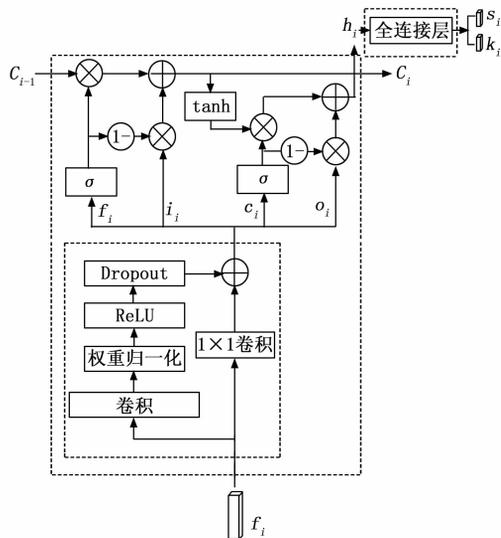


图 4 实例卷积循环单元

最后, 将得到的 h_i 经过全连接层分别得到状态向量 s_i 和卷积核参数向量 k_i 。其中 s_i 表示是否需要继续预测该实例的形状, 通过循环预测该实例起始点参数可以解决一个起始点包含多条滑行道线的情况。而 k_i 用于预测相应滑行道线形状时的动态卷积核参数。

1.3 损失函数设计

在整个检测过程中, 首先对滑行道线进行实例检测。鉴于滑行道线往往具有细长且弯曲的结构特性, 若采用常规的实例检测方法去预测每个实例的中心点, 会降低检测精度。但如果预测每条线的起始点会变得相对简单, 这不仅是由于在密集场景下能有效减少误检, 而且每条滑行道线的起始点在视觉上也非常明显。因此,

遵循文献 [20] 和 [27] 的方法, 通过检测滑行道线的起始点来实现实例检测, 并且采用焦点损失进行约束:

$$\ell_{\text{point}} = -\frac{1}{N_p} \sum_k \begin{cases} (1 - \hat{P}_{xy})^\alpha \log(P_{xy}) & P_{xy} = 1 \\ (1 - \hat{P}_{xy})^\beta (\hat{P}_{xy})^\alpha \log(1 - \hat{P}_{xy}) & \text{otherwise} \end{cases} \quad (4)$$

其中: P_{xy} 是起始点坐标 (x, y) 处的实际值, 而 \hat{P}_{xy} 是坐标 (x, y) 处的预测值, N_p 是输入图像中起始点处的数量。

为了解决分叉及密集场景下滑行道线的检测问题, 在 1.2 节中提出了 ICRU, 本文采用交叉熵损失函数来约束状态输出, 如式 (5) 所示:

$$\ell_{\text{state}} = -\frac{1}{N_s} \sum_k [y_k \cdot \log(s_k) + (1 - y_k) \cdot \log(1 - s_k)] \quad (5)$$

其中: s_k 是第 k 个实例输出的状态向量, y_k 是第 k 个实例的实际值, N_s 是一个批次状态输出的总数。

在预测每条滑行道线具体形状时, 采用逐行公式中的逐行位置和垂直范围, 以及偏移映射。将输入图像划分为 $X \times Y$ 的网格。对于每一条滑行道线, 每一行都会有一个相应的横坐标。在垂直方向上, 可以根据每条滑行道线是否穿过当前行来确定。所以在垂直范围上对每一行进行二元分类, 即可确定垂直范围。因此, 采用交叉熵损失函数来引导训练过程:

$$\ell_{\text{range}} = -\sum_i (y_{gr}^i \log(v_i) + (1 - y_{gr}^i) \log[1 - v_i]) \quad (6)$$

其中: v_i 表示第 i 行预测存在的概率, y_{gr}^i 是第 i 行实际值。

相应的对于每行的横坐标可以通过该行中的概率分布来确定。对于每一行, 预测滑行道线出现在每个网格中的概率:

$$p_i = \text{softmax}(f_{\text{loc}}^i) \quad (7)$$

其中: i 表示第 i 行, f_{loc}^i 表示第 i 行的特征向量, p_i 是第 i 行概率向量。

在每行上预测的横坐标为:

$$E(\hat{x}_i) = \sum_j j p_{ij} \quad (8)$$

其中: p_{ij} 是滑行道线通过坐标 (i, j) 的概率, j 是第 i 行中的第 j 个网格。

逐行上损失定义为:

$$\ell_{\text{row}} = \frac{1}{N_v} \sum_{i \in V} |E(\hat{x}_i) - x_i| \quad (9)$$

其中: V 表示垂直范围, N_v 表示有效行数, x_i 表示每行中实际位置。

在逐行预测过程中, 式 (8) 仅利用每个网格左侧的横坐标来进行预测。然而在实际中并不一定是最精确的位置, 如果按照该方法将预测的结果作为预测点的逐行位置, 将会引入一定误差, 这在密集滑行道线中将会

带来显著的影响。所以在此基础上预测每个网格的偏移量, 并使用 L_1 损失来约束偏移值:

$$\ell_{\text{offset}} = \frac{1}{N_U} \sum_{(i,j) \in U} |\hat{\delta}_{ij} - \delta_{ij}| \quad (10)$$

其中: $\hat{\delta}_{ij}$ 和 δ_{ij} 表示在点 (i, j) 上预测偏移值和实际偏移值。我们将 U 定义为滑行道线附近固定宽度区域, N_U 是 U 中的像素数。

如上所述, 本文采用基于行的方式检测滑行道线, 所以滑行道线可以看作是一系列离散点的集合。本文中首先对每个实例的起始点进行检测, 然而在滑行道线密集场景下对每条滑行道线的终止位置进行约束也是至关重要。这样可以更加有效避免在密集场景中预测出的一条曲线包含两个不同实例信息的情况, 所以本文提出 $PIoU$ 损失函数。 $PIoU$ 是每条滑行道线最后一个点之间的交集与并集的比, 如图 5 所示。对于每条曲线的最后一个点, 首先以该点坐标为中心, 以 e 为半径扩展为矩形, 进而计算每条曲线最后一点实际与预测扩展矩形之间的交并比。 $PIoU$ 定义为:

$$PIoU = \frac{[\min(x_{\text{end}}^p + e, x_{\text{end}}^g + e) - \max(x_{\text{end}}^p - e, x_{\text{end}}^g - e)] \times 2e}{[\max(x_{\text{end}}^p + e, x_{\text{end}}^g + e) - \min(x_{\text{end}}^p - e, x_{\text{end}}^g - e)] \times 2e} \quad (11)$$

其中: $x_{\text{end}}^p + e$ 和 $x_{\text{end}}^p - e$ 是最后一个预测点的横坐标 x_{end}^p 的扩展点, $x_{\text{end}}^g + e$ 和 $x_{\text{end}}^g - e$ 是最后一个实际点的横坐标 x_{end}^g 的扩展点。

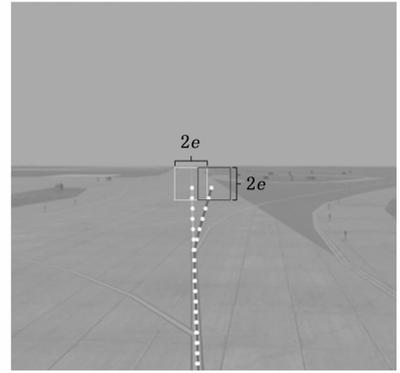


图 5 点交并比

所以 $PIoU$ 的损失就可以被定义为:

$$\ell_{\text{PIoU}} = 1 - PIoU \quad (12)$$

因此在训练过程中的总损失定义为:

$$\ell_{\text{total}} = \alpha \ell_{\text{range}} + \beta \ell_{\text{row}} + \gamma \ell_{\text{offset}} + \eta \ell_{\text{state}} + \theta \ell_{\text{point}} + \xi \ell_{\text{PIoU}} \quad (13)$$

其中在训练过程中对于不同损失权重大小对模型检测结果有着不同的影响, 根据不同损失对检测结果的不同程度的影响以及在实验过程中的调整, 超参数 $\alpha, \beta,$

γ, η, θ 和 ξ 分别设置为 0.4、1.0、0.4、1.0、1.0 和 1.0。

2 实验结果

2.1 数据集及评价标准

2.1.1 数据集

目前,尚未检索到国内外公开的滑行道线数据集。为此,本文利用模拟飞行软件,驾驶飞机在机场从停机位滑行到跑道以及从跑道滑行到停机位整个滑行过程,并录制整个滑行过程,从而实现采集飞机在机场场面滑行阶段滑行道线图像,自制滑行道线仿真数据集 CAT-Lines (China airport taxiway lines)。该数据集包含 3 481 帧图像,每帧图像大小为 $1\ 280 \times 720$ 像素,由于数据量规模较少,为了能够确保模型得到充分学习,并且保证模型的参数能够得到调整,将数据按照 8:1:1 的比例随机划分训练集、验证集和测试集。鉴于当前缺乏公开的滑行道线数据集,且滑行道线和车道线在结构上具有较强的相似性。为了广泛评估本文提出的方法,本文还在 CurveLanes^[12]、CULane^[11] 和 TuSimple^[28] 三个公共数据集上也进行了实验。CurveLanes 是一个包含了大量曲线的车道线数据集;CULane 是一个广泛使用的大型车道线检测数据集,涵盖了 9 种不同的场景;TuSimple 则是另一个广泛使用的高速公路驾驶场景数据集。这 3 个公共数据集以及自制数据集划分的详细信息如表 1 所示。

表 1 数据集划分详细信息

数据集	总帧数	训练集	验证集	测试集
CATLines	3 481	2 785	348	348
Tusimple ^[28]	6 408	3 268	258	2 782
CULane ^[11]	133 235	88 880	9 675	34 680
CurveLanes ^[12]	150 000	100 000	20 000	30 000

2.1.2 评价标准

为了能够有效的与其他方法进行对比, CurveLanes 数据集、CULane 数据集和 TuSimple 数据集均采用数据集的官方评价指标。对于本文自制的 CATLines 数据集,根据上述 3 个车道线数据集以及自制滑行道数据集的特点和实际需求,采用 F_1 作为评价标准。其中 CurveLanes 数据集和 CULane 数据集官方评价指标以 F_1 作为评价标准。该标准通过计算预测的曲线与真实标签之间的 IoU 来判断预测的准确性。将每条曲线标记视为宽度为 30 像素的线,当 IoU 大于 0.5 时认为预测正确,视为真阳性 (TP, true positive); 小于 0.5 时表明预测错误,视为假阳性 (FP, false positive); 因漏检而未被检测出来的视为假阴性 (FN, false negative)。因此, F_1 定义为:

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (16)$$

对于 TuSimple 数据集,官方数据集的评价指标为准确率,定义如下:

$$accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}} \quad (17)$$

其中: C_{clip} 是一张图像中正确预测车道线点的数量,而 S_{clip} 是图像中车道线点的总数。当车道线点的预测值与实际真实值之间的距离小于 20 像素时,则认为预测正确。

2.2 实验设置

本文基于不同的主干网络构建了大型、中型和小型 3 个版本,其唯一差异在于分别采用了 ResNet-18、ResNet-34 和 ResNet-101 作为主干网络。在优化策略上,本文使用了 Adam 优化器^[29]和步长学习衰减率^[30],初始学习率为 3×10^{-4} 。针对 CATLines、CurveLanes、CULane 和 TuSimple 这四个数据集,我们分别设置了 100、14、16 和 70 的训练轮数,批处理数据量设置为 32。对于 CurveLanes 遵循 CurveLane,在验证集上测试结果。实验环境为 Ubuntu 20.04 操作系统, CPU 为 11th Gen Intel (R) Core (TM) i9-11900, GPU 为 NVIDIA GeForce RTX 3060。

2.3 实验结果

图 6 中展示了 CATLines 数据集上的可视化结果。第一行为标签的真实情况,第二行至第五行分别呈现了 CLRNet-S^[18]、UFLDv2-M^[19]、CondLaneNet-L^[20] 以及本文提出方法的可视化结果。结果表明,本文的方法不仅能够有效检测常规的直线和曲线,还可以处理复杂拓扑结构,即使是滑行道边线相邻的密集曲线或滑行道出口处的分叉曲线,该方法依然能够实现准确检测,表现出了良好的鲁棒性和适应性。

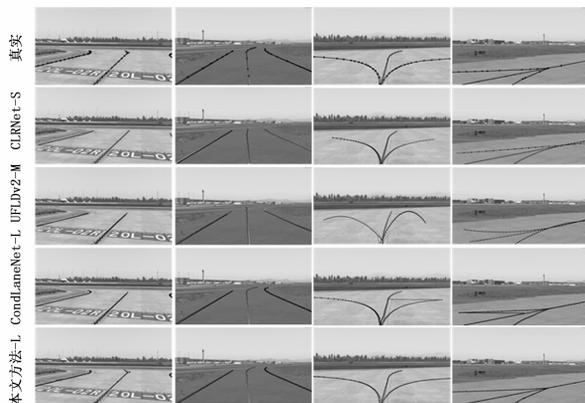


图 6 不同方法可视化检测结果对比

在 CATLines 数据集上比较结果如表 2 所示。该数据集不仅包含一般的直线和曲线，还涵盖了大量的拓扑结构的滑行道线。其中在滑行道边上存在大量相邻的曲线和直线，在滑行道出口处主要是分叉的曲线，所以一些在车道线上检测良好的方法在 CATLines 数据集上表现出较低的检测结果。针对这些问题，本文提出的方法在 F_1 上取得了良好的效果。

表 2 CATLines 数据集上不同方法比较

方法	F_1 / %	精确率 / %	召回率 / %
CLRNet-S ^[18]	55.13	47.60	65.48
UFLDv2-S ^[19]	76.58	77.34	75.83
UFLDv2-M ^[19]	77.48	78.89	76.12
CondLaneNet-S ^[20]	83.62	84.94	82.33
CondLaneNet-M ^[20]	83.76	84.31	83.23
CondLaneNet-L ^[20]	84.38	85.43	83.35
CLRerNet-DLA34 ^[21]	79.61	82.61	76.82
本文方法-S	85.08	86.60	83.61
本文方法-M	85.20	86.03	84.38
本文方法-L	85.81	86.88	84.76

在 CurveLanes 数据集上比较结果如表 3 所示。相较于其他两个公共车道线数据集，CurveLanes 数据集中除了包含一般的直线和多车道外，每张图像中车道线数量也更多，并且涵盖了大量的曲线车道线。除此之外，该数据集里面还包含了一些“S”型以及“Y”型车道线。结果表明本文所提出的方法在面对这些情况也表现出了很好的效果。

表 3 CurveLanes 数据集上不同方法比较

方法	F_1 / %	精确率 / %	召回率 / %
SCNN ^[11]	65.02	76.13	56.74
Enet-SAD ^[31]	50.31	63.60	41.60
UFLDv2-S ^[19]	80.45	81.49	79.44
UFLDv2-M ^[19]	81.34	81.93	80.76
本文方法-S	84.34	87.48	84.42
本文方法-M	85.69	88.08	83.42
本文方法-L	86.17	88.65	83.83

在 Tusimple 数据集上结果如表 4 所示。相较于另外两个公共数据集，Tusimple 数据集的数据量较小，且场景较为单一，因此不同方法之间的性能差距较小。尽管如此，本文提出的方法在该数据集上仍能达到 96.45% 的准确率。

在 CULane 数据集上比较结果如表 5 所示。表中对比了不同场景下车道线的检测效果，从中可以看出本文提出的方法在多数场景中取得了良好的效果，证明了本文算法在应对复杂场景下的车道线检测也有很好的检测精度和鲁棒性。

表 4 TuSimple 数据集上不同方法比较

方法	准确率 / %	FPS / (帧 / 秒)
SCNN ^[11]	96.53	7.5
UFAST-S ^[19]	95.82	312.5
UFAST-M ^[19]	95.86	169.5
LSTR ^[15]	96.18	420.0
LaneATT-S ^[17]	95.57	250.0
CondLaneNet-S ^[20]	95.48	220.0
CondLaneNet-M ^[20]	95.37	154.0
CondLaneNet-L ^[20]	96.54	58.0
本文方法-S	95.49	138.0
本文方法-M	95.61	85.0
本文方法-L	96.45	31.0

表 5 CULane 数据集上不同方法比较

场景	SCNN ^[11]	UFAST-S ^[19]	RESA-M ^[13]	CondLaneNet-S ^[20]	本文方法-S	本文方法-M
正常 / %	90.60	87.70	91.90	92.87	92.42	93.15
拥挤 / %	69.70	66.00	72.40	75.79	75.55	76.22
夜间 / %	66.10	62.10	69.80	73.23	71.84	73.79
无车道线 / %	43.40	40.20	46.30	52.39	52.45	52.78
阴影 / %	66.90	62.80	72.00	80.01	79.08	76.46
箭头 / %	84.10	81.00	88.10	89.37	88.66	89.83
亮光 / %	58.50	58.40	66.50	70.72	71.21	70.93
弯道 / %	64.40	57.90	68.60	72.40	72.74	72.94
十字路口(个)	1 990	1 743	1 896	1 364	1 577	1 660
平均值 / %	71.60	68.40	74.50	78.14	77.50	78.26
FPS(帧/秒)	7.5	322.5	45.0	220.0	120.0	85.0

2.4 消融实验

本文在自制滑行道线数据集 CATLines 上进行消融实验，结果如表 6 所示。以 ResNet-34 为主干网络，FPN 作为网络的颈部，基于行的检测方法为基准。第一行显示了基准的结果。在第二行中，使用了本文提出的 ICRU 模块。在第三行中将颈部的 FPN 替换为本文提出的 HC-FPN 模块。最后一行在损失中添加了 PIoU。

表 6 消融实验结果对比

基准	ICRU	HC-FPN	PIoU	F_1 / %
✓				53.07
	✓			78.36(+25.29)
	✓	✓		82.29(+29.22)
	✓	✓	✓	85.20(+32.13)

通过比较表 6 的前两行结果，可以看出本文提出的 ICRU 在处理分叉以及相邻滑行道线方面展现出了显著的效果。进一步比较前三行的结果，我们可以看到 HC-FPN 能够更好的多级融合高层全局信息，对滑行道线的检测具有提升作用。最后一行中表明在基于行的检测基础上添加本文提出的 PIoU 更利于滑行道线的检测。

为了验证本文所提出的 HC-FPN 方法的有效性, 与其他多尺度特征融合方法进行了比较, 结果如表 7 所示。从表中可以看出, HC-FPN 在多尺度特征融合上要优于其他方法, 这表明该方法有效地增强了模型对滑行道线的检测能力。

表 7 不同特征融合方法对比

颈部	$F_1 / \%$
FPN ^[26]	83.22
BiFPN ^[32]	83.60
PaFPN ^[33]	83.50
HS-FPN ^[34]	83.89
HC-FPN	85.20

3 结束语

本文提出一种融合 HC-FPN 与 ICRU 的机载视觉滑行道线检测方法。该方法首先预测滑行道线实例, 在基于行锚检测与动态卷积的方法, 预测每条滑行道线的形状。为更准确预测滑行道线实例与每条滑行道线的形状, 设计了高层融合特征金字塔, 进行多级融合高层信息。为了能够检测分叉与相邻的滑行道线, 设计了实例卷积循环单元。为了进一步提高检测精度, 利用点交并比对每条滑行道线的最后一点进行约束。实验结果表明, 该方法不仅在自制滑行道线数据集 CATLines 上 F_1 达到了 85.81%, 在 CurveLanes 数据集和 CULane 数据集上 F_1 达到了 86.17% 和 78.63%, 在 TuSimple 数据集上准确率达到 96.45%。下一步将采集实际飞机在滑行道上的数据, 进而扩充 CATLines 数据集, 同时在滑行道线检测的基础上解算出飞机滑行方向与滑行道中线的偏离角度。

参考文献:

- [1] 王 浩, 钟珍伟, 陶志成, 等. 基于机场协同的飞行器滑行道线引导系统及方法 [P]. 中国: 202310445610. 1, 2023-08-08.
- [2] 施陶丁格 T C, 卡拉汉 K S, 张 I, 等. 滑行道上的飞行器定位 [P]. 中国: 201911273588.7, 2020-06-26.
- [3] AIRBUS. Airbus concludes ATTOL with fully autonomous flight tests [EB/OL]. 2020; 1-16. [2024-09-01]. <https://www.airbus.com/en/newsroom/press-releases/2020-06-airbus-concludes-attol-with-fully-autonomous-flight-tests>.
- [4] 吴骅跃, 赵祥模. 基于 IPM 和边缘图像过滤的多干扰车道线检测 [J]. 中国公路学报, 2020, 33 (5): 153-164.
- [5] ANDRADE D C, BUENO F, FRANCO F R, et al. A novel strategy for road lane detection and tracking based on a vehicle's forward monocular camera [J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 20 (4): 1497-1507.
- [6] LIU J, LOU L, HUANG D, et al. Lane detection based on straight line model and k-means clustering [C] // 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), 2018: 527-532.
- [7] LUO S, ZHANG X, HU J, et al. Multiple lane detection via combining complementary structural constraints [J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22 (12): 7597-7606.
- [8] 蔡英凤, 张田田, 王海, 等. 基于实例分割和自适应透视变换算法的多车道线检测 [J]. 东南大学学报 (自然科学版), 2020, 50 (4): 775-781.
- [9] NEVEN D, DE BRABANDERE B, GEORGOULIS S, et al. Towards end-to-end lane detection: an instance segmentation approach [C] // 2018 IEEE Intelligent Vehicles Symposium, 2018: 286-291.
- [10] ZHANG Y, LU Z, ZHANG X, et al. Deep learning in lane marking detection: a survey [J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23 (7): 5976-5992.
- [11] PAN X, SHI J, LUO P, et al. Spatial as deep: spatial CNN for traffic scene understanding [C] // Proceedings of the AAAI Conference on Artificial Intelligence, 2018: 7276-7283.
- [12] XU H, WANG S, CAI X, et al. Curvelane-NAS: unifying lane-sensitive architecture search and adaptive point blending [C] // Computer Vision-ECCV 2020: 16th European Conference, 2020: 689-704.
- [13] ZHENG T, FANG H, ZHANG Y, et al. RESA: recurrent feature-shift aggregator for lane detection [C] // Proceedings of the AAAI Conference on Artificial Intelligence, 2021: 3547-3554.
- [14] TABELINI L, BERRIEL R, PAIXAO T M, et al. PolyLanenet: lane estimation via deep polynomial regression [C] // 2020 25th International Conference on Pattern Recognition (ICPR), 2021: 6150-6156.
- [15] LIU R, YUAN Z, LIU T, et al. End-to-end lane shape prediction with transformers [C] // Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021: 3694-3702.
- [16] LI X, LI J, HU X, et al. Line-CNN: end-to-end traffic line detection with line proposal unit [J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21 (1): 248-258.
- [17] TABELINI L, BERRIEL R, PAIXAO T M, et al. Keep your eyes on the lane: real-time attention-guided lane detection [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021: 294-302.

[18] ZHENG T, HUANG Y, LIU Y, et al. CLRNet: cross layer refinement network for lane detection [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022: 898 – 907.

[19] QIN Z, WANG H, LI X. Ultra fast structure-aware deep lane detection [C] // Computer Vision-ECCV 2020: 16th European Conference, 2020: 276 – 291.

[20] LIU L, CHEN X, ZHU S, et al. Condlanenet: a top-to-down lane detection framework based on conditional convolution [C] // Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021: 3773 – 3782.

[21] HONDA H, UCHIDA Y. CLRerNet: improving confidence of lane detection with LaneIoU [C] // Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024: 1176 – 1185.

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

[23] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770 – 778.

[24] JIA X, DE BRABANDERE B, TUYTELAARS T, et al. Dynamic filter networks [J]. Advances in Neural Information Processing Systems, 2016, 29: 667 – 675.

[25] YANG B, BENDER G, LE Q V, et al. Condconv: conditionally parameterized convolutions for efficient inference [J]. Advances in Neural Information Processing Systems, 2019, 32: 1307 – 1318.

[26] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature

（上接第 41 页）

[14] LI J, LI D, XIONG C, et al. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation [C] //International conference on machine learning, PMLR, 2022: 12888 – 12900.

[15] GIRSHICK R. Fast r-cnn [C] //Proceedings of the IEEE international conference on computer vision. 2015: 1440 – 1448.

[16] 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.

[17] LIU W, ANGUELOV D, ERHAN D, et al. Ssd: Single shot multibox detector [C] //Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11 – 14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21 – 37.

[18] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. YOLOv7: pyramid networks for object detection [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2117 – 2125.

[27] DUAN K, BAI S, XIE L, et al. Centernet: keypoint triplets for object detection [C] // Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 6569 – 6578.

[28] ZHOU K. Tusimple benchmark [EB/OL]. 2018: 10 – 20. [2024 – 09 – 01]. <https://github.com/TuSimple/tusimple-benchmark>.

[29] KINGMA D P, BA J. Adam: a method for stochastic optimization [J]. ArXiv Preprint ArXiv: 1412.6980, 2014.

[30] LOSHCHILOV I, HUTTER F. Decoupled weight decay regularization [J]. ArXiv Preprint ArXiv: 1711.05101, 2017.

[31] HOU Y, MA Z, LIU C, et al. Learning lightweight lane detection cnns by self attention distillation [C] // Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 1013 – 1021.

[32] TAN M, PANG R, LE Q V. Efficientdet: scalable and efficient object detection [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 10781 – 10790.

[33] LIU S, QI L, QIN H, et al. Path aggregation network for instance segmentation [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 8759 – 8768.

[34] CHEN Y, ZHANG C, CHEN B, et al. Accurate leukocyte detection based on deformable-DETR and multi-level feature fusion for aiding diagnosis of blood diseases [J]. Computers in Biology and Medicine, 2024, 170: 107917.

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.

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

[20] GHIASI G, CUI Y, SRINIVAS A, et al. Simple copy-paste is a strong data augmentation method for instance segmentation [C] //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021: 2918 – 2928.

[21] 王浩然, 戴 鹏, 刘俊博, 等. 高速铁路线路环境异物入侵视频检测系统研制 [J]. 计算机测量与控制, 2024, 32 (10): 86 – 91.

[22] 周 瑶, 周 石. 基于 YOLOv5 改进的铁路工人安全帽检测算法研究 [J]. 计算机测量与控制, 2024, 32 (3): 71 – 78.