

融合深度先验与几何约束的鲁棒车辆多任务感知系统研究

庄奕娜、许诗晗、陈宝伊、郑唯敏、黄和、汪雪枫、黄思宇^{*}，曾海鹏老师[†]

中山大学 智能工程学院，深圳 518083

【摘要】智能交通系统的构建依赖于对道路场景中车辆全要素信息的精准感知。然而，在非受控的自然道路场景下，多目标车辆的细粒度分类、大角度畸变车牌识别以及单目视觉下的高精度测速仍面临严峻挑战。本文提出了一套集车辆检测、细粒度车型识别、鲁棒车牌识别及多模态车速估计于一体的综合感知框架。首先，针对车型识别中类间相似度高与长尾分布问题，我们构建了基于YOLOv11-Large的端到端检测架构，通过Mosaic数据增强与加权交叉熵损失优化，在BIT-Vehicle数据集上实现了97.73%的mAP@0.5与159.9 FPS的实时推理速度，并对比了基于ResNet50/Swin-Transformer的级联架构，揭示了端到端范式在全局特征提取上的优势。其次，针对车牌识别中的透视畸变难题，提出了一种“关键点检测-几何矫正-序列识别”的解耦方案。利用YOLOv8-Pose精确回归车牌角点，结合透视变换与CRNN网络，在CCPD高难度测试集上取得了96.44%的全字匹配率，显著优于通用OCR方案。最后，在车速估计任务中，我们系统性地探索了基于单应性矩阵的几何视觉、基于MiDaS的相对深度估计及基于MoGe的绝对深度估计三种技术路线。实验表明，基于单应性投影的几何方案在固定监控场景下实现了精度与效率(24.8 FPS)的最佳平衡，而基于MoGe的免标定方案展现了极强的前瞻性。此外，本文还设计了一套可视化的交互式评估系统，验证了所提算法在实际工程部署中的有效性。

【关键词】智能交通系统；细粒度车型识别；车牌矫正与识别；单目车速估计；深度估计；YOLO

1 引言

随着城市化进程的加速，智能交通系统(Intelligent Transportation Systems, ITS)已成为缓解交通拥堵、提升道路安全及实现自动驾驶协同感知的重要基础设施。作为ITS的“眼睛”，计算机视觉技术被广泛应用于交通监控数据的自动化分析中。然而，尽管通用目标检测算法已取得显著进展，但在面对复杂多变的真实道路场景时，现有的感知系统仍面临诸多技术瓶颈：(1) **细粒度特征混淆**：不同类别的车辆(如SUV与Sedan)在特定视角下外观高度相似，且数据分布往往呈现显著的长尾特性，导致少数类样本识别率低；(2) **非平面几何畸变**：路侧监控相机的安装角度导致车牌图像普遍存在倾斜与透视变形，严重影响了基于水平框的传统OCR模型的识别精度；(3) **单目深度缺失**：在缺乏雷达等昂贵传感器辅助的情况下，仅依靠单目摄像头恢复场景的三维尺度并进行精

确测速，始终是一个病态问题。针对上述挑战，本文旨在构建一个高效、鲁棒且具备工程落地价值的多任务车辆感知系统。在**车型识别**方面，我们深入探究了“端到端检测”与“级联分类”两种范式的性能边界。通过在BIT-Vehicle数据集上的广泛实验，我们发现基于YOLOv11-Large的端到端架构能够更有效地利用全图上下文信息，在处理长尾类别(如Minivan)时表现出优于两阶段方法的召回能力。在**车牌识别**方面，不同于直接回归边界框的传统方法，我们引入了基于Pose的关键点检测机制，通过精确定位车牌角点并实施几何矫正，将不规则的车牌图像标准化，从而大幅提升了后端CRNN网络的序列解码精度。在最具挑战的**车速估计**任务中，本文并未局限于单一技术路线，而是对比了经典的几何视觉方法与前沿的深度学习方法。我们不仅实现了基于RANSAC单应性矩阵的高效测速方案，还创新性地探索了基于MoGe V2端到端绝对深度估计的“免标定”测速新范式，为

[†] 指导教师：曾海鹏老师

*E-mail: huangsyy223@mail2.sysu.edu.cn

解决单目视觉尺度不确定性问题提供了新的思路。

本文的主要贡献总结如下：

1. **多范式架构评估与优选：**系统性评估了 CNN (ResNet, MobileNet, ConvNeXt) 与 Transformer (Swin-T) 架构在细粒度车型分类中的性能，确立了以 YOLO11 为核心的高效端到端感知基线。
2. **鲁棒的畸变车牌识别流水线：**提出并验证了基于 YOLOv8-Pose 与透视变换的几何矫正模块，有效解决了大角度监控场景下的车牌识别难题，在 CCPD Hard 数据集上保持了 SOTA 级别的识别率。
3. **多模态单目测速技术探索：**首次在同一基准下对比了单应性投影、相对深度估计 (MiDaS) 与绝对深度估计 (MoGe) 三种测速方案，揭示了各方案在精度、速度与部署难度之间的权衡关系。
4. **全栈式系统集成：**开发了基于 Streamlit 的可视化交互系统，集成了上述核心算法，支持视频流的实时处理与多维度数据分析，为算法的工程化落地提供了直观验证。

我们的代码与预训练模型已开源于 GitHub 仓库：<https://github.com/Henry-coder-H/Deepl-Learning-coding-task>，我们强烈建议您阅读这个仓库，里面有详细的模型实现原理与使用说明。

2 基于 YOLO 系列的模型对比与选型

为了平衡车辆检测任务中检测精度与实时推理速度、明确适用于车辆识别系统的最优目标检测骨干网络，我们首先构建了标准化基准测试流程，选取业界主流的 **YOLOv8** 系列与最新迭代的 **YOLO11** 系列共 10 个不同参数量级的预训练模型开展横向对比实验，通过量化指标的系统分析筛选最优模型。该模型将作为本项目车辆识别系统的基础检测器，负责对图像或视频流中的车辆进行精确的**目标检测与空间定位**，从而为后续的车型分类、车速估计及车牌识别等**级联任务**提供坚实的支撑。

2.1 实验环境与评估指标

项目的所有对比实验均在搭载 NVIDIA GeForce RTX 5070 Laptop GPU (8 GB 显存) 的工作站上进行，软件环境基于 PyTorch 2.x 框架与 CUDA 12.1 加速库。

实验评估数据源自 Kaggle 平台的 **Vehicle Detection Dataset**。考虑到所选预训练模型均基于 COCO 数据集训练（涵盖 80 类通用目标），直接用于本任务评估时，非车辆类别的输出会对评估结果产生干扰。为此，我们开发了专用的 **YOLOModelEvaluator** 评估模块以实现类别自适应过滤机制：在模型推理阶段，自动识别并保留与数据集标签空间（含 Car、Bus、Truck 等车辆类别）语义匹配的输出通道，屏蔽无关类别输出，确保对不同预训练模型泛化能力的评估公平性。

评估指标体系采用多维度量化标准：以**全类平均精度均值 (mAP₅₀₋₉₅)**作为检测性能的核心量度；同时记录**每秒传输帧数 (FPS)**与**推理延迟 (Inference Latency)**以量化模型的实时处理能力；并引入**参数量 (Parameters)**与**浮点运算次数 (FLOPs)**作为衡量模型计算复杂度的标准。

2.2 骨干网络量化评估与模型选型决策

基于标准化基准测试流程，不同架构与参数量级的模型对比结果如表 1 所示。

参数效率与检测精度的权衡分析。实验数据揭示了 **yolo11** 架构在参数利用率上具有显著的代际优势。具体而言，**yolo11-s** 在仅有 9.44 M 参数量的情况下，取得了 0.350 的 mAP，该性能表现优于参数量为其 2.7 倍的上一代 **YOLOv8-m** (25.89 M 参数, mAP 0.346)。这表明新一代架构在特征提取与多尺度融合机制上实现了更高的编码效率。

在对检测精度极限的探索中，**yolo11-l (Large)** 展现了最佳的综合性能，以 0.370 的 mAP₅₀₋₉₅ 位居榜首。值得注意的是，模型缩放并未呈现线性的性能收益：随着规模进一步扩大至 Extra-Large (x) 版本，性能反而出现饱和甚至衰退迹象 (**yolo11-x** 为 0.359, **YOLOv8-x** 为 0.369)。这种现象推测归因于超大模型在特定领域数据上的过拟合倾向，或是预训练权重在特定车辆子类上的泛化瓶颈。

实时性考量与最终决策。尽管轻量级的 **YOLOv8-n** 实现了最高的 400 FPS，但其较低的精度 (0.307 mAP) 难以满足复杂交通场景下对远距

离小目标的精确召回需求。相比之下，**yolo11-l** 在保持最高检测精度的同时，在 RTX 5070 移动端上的单帧推理延迟仅为 7.8 ms (约 128.5 FPS)。这一处理速度远超实时视频流通常要求的 30 FPS 标准，意味着该模型在确保高精度检测的同时，仍保留了约 75% 的算力余量。这部分冗余算力对于保障后续车牌识别级联网络、多目标追踪及复杂逻辑判断的实时运行至关重要。

综上所述，综合权衡检测精度、推理速度以及算力资源占用，本实验最终确定选用 **yolo11-l** 作为本系统的基础骨干网络。该选择既确立了系统在复杂场景下的鲁棒性基线，也为多任务并行处理奠定了坚实的计算基础。

表 1 YOLO 系列模型在 Vehicle Detection Dataset 上的实测性能对比

模型 (Model)	参数量 (M)	FLOPs (G)	mAP ₅₀₋₉₅	延迟 (ms)	FPS
YOLOv8-n	3.15	8.7	0.307	2.5	400.0
YOLOv8-s	11.16	28.6	0.319	3.0	328.9
YOLOv8-m	25.89	78.9	0.346	6.6	151.3
YOLOv8-l	43.67	165.2	0.369	9.4	106.2
YOLOv8-x	68.20	257.8	0.369	14.1	70.7
yolo11-n	2.62	6.5	0.325	22.3	44.8
yolo11-s	9.44	21.5	0.350	11.2	89.0
yolo11-m	20.09	68.0	0.344	11.3	88.3
yolo11-l	25.34	86.9	0.370	7.8	128.5
yolo11-x	56.90	194.9	0.359	26.8	37.3

3 车型识别

3.1 任务概述与技术路线

3.1.1 任务定义

车型识别 (Vehicle Type Recognition, VTR) 是智能交通系统中的关键环节，其核心目标是从复杂交通场景的图像中，准确识别出车辆所属的具体类别。本实验的具体任务是利用 BIT-Vehicle 数据集，在复杂的城市道路场景下，实现对 Bus(大巴)、Microbus(面包车)、Minivan(小型货车)、Sedan(轿车)、SUV(运动型多用途车) 和 Truck(卡车) 六类车辆的精准定位与分类。

相比于常规的车辆检测，本任务更侧重于对相似车型 (如 Sedan 与 SUV、Microbus 与 Minivan) 的细粒度判别，对模型的表征能力与判别力提出了更高要求。因此任务的核心难点在于车辆外观

的类内差异大 (如不同品牌的轿车)、类间相似度高 (如 SUV 与 Sedan 的视角混淆) 以及数据集中显著的长尾分布问题。

3.1.2 技术路线选择

为探究不同深度学习架构在车辆特征提取上的性能差异，针对性解决车型识别任务的核心技术挑战 (即有效提取车辆判别性特征以区分外形相似车型、平衡模型识别精度与推理速度)，本实验设计两种差异化技术方案 (如图1所示)，具体设计思路与选型依据如下：

- **方案 A：端到端目标检测。** 端到端模型能够直接从原始图像中同时完成车辆定位与类别识别，避免了多阶段流程中可能出现的误差累积。而选择 **YOLO11-large** 作为基础模型是因为该架构在目标检测任务中兼具高精度与高速度，其深层特征提取能力能够捕捉车辆的全局特征与局部细节，适合处理交通场景中车辆密集、背景复杂的情况 (第2.2 节所述)。该方案的设计初衷是验证端到端架构在车型识别任务中的可行性与高效性，为实时性要求较高的应用场景提供解决方案。
- **方案 B：级联式检测与识别。** 考虑到端到端模型在细粒度分类任务中可能存在精度瓶颈，尤其是对于外形高度相似的车型，单独的分类分支可能无法充分学习到类别间的细微差异。因此设计级联式架构，采用“检测 + 分类”的两阶段策略。首先利用检测器定位车辆并裁剪 (Crop) 感兴趣区域 (ROI)，随后将 ROI 送入专门的分类网络进行识别。该方案解耦了定位与分类任务，聚焦车辆本身的特征学习，提升细粒度车型的识别精度，同时对比不同经典分类模型在该任务中的适配性。

3.2 数据集准备

3.2.1 BITVehicle

BIT-Vehicle 数据集是车型识别任务专用的公开数据集，其图像通过两台固定视角相机采集，涵盖 Bus、Microbus、Minivan、Sedan、SUV 和 Truck 6 个核心车型类别。数据集图像以 1080p 高清分辨率为主，车辆目标在图像中的占比存在显著差异，且部分样本包含光照变化、轻微遮挡等场景干扰，

Vehicle Recognition Technical Scheme Comparison

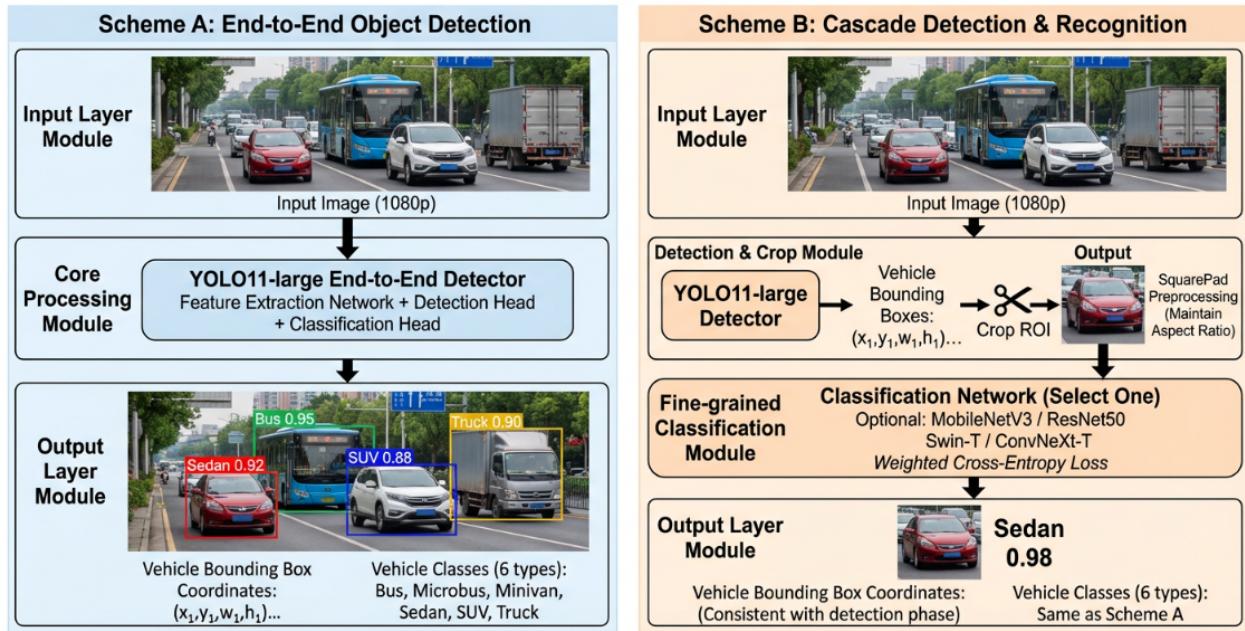


图 1 车辆识别方案对比

能够真实模拟复杂交通环境下的车型识别场景，为模型泛化能力验证提供了贴合实际应用的数据源。

同时，该数据集原始标注信息以.mat文件格式存储，包含图像名称、图像尺寸、车辆边界框坐标及车型类别等关键标注字段，为模型训练的标签解析提供了完整基础信息。值得注意的是，6类车型的样本数量分布存在明显不均衡性(如图2所示)，其中Sedan样本占比最高，Minivan样本占比最低，这种长尾分布特性会导致模型训练过程中对少数类样本的特征学习不充分，为模型分类性能的均衡优化带来了特定挑战。

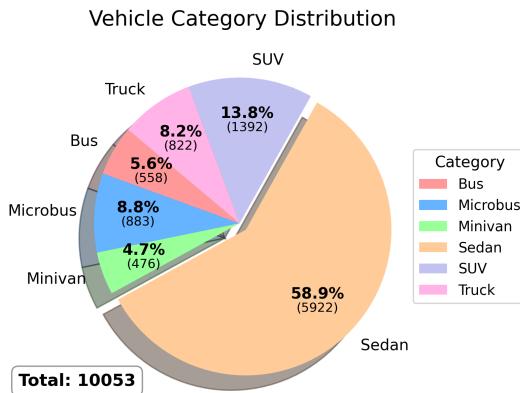
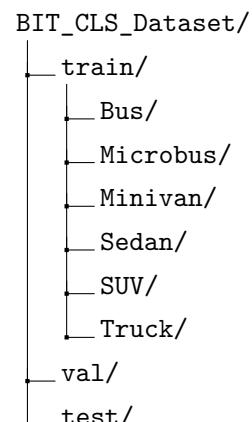


图 2 BITVehicle 各类别数量分布

3.2.2 数据处理

为适配两种技术方案的训练需求，本实验对BIT-Vehicle数据集进行了多步核心处理，确保数据格式规范、划分合理：

- (1) **格式转换**: 读取.mat标注文件，将边界框坐标转换为YOLO格式(.txt标注文件)，并按7:2:1的比例划分为训练集、验证集和测试集。
- (2) **分类数据集生成**: 基于YOLO格式的标签，我们将每辆车的边界框区域从原图中裁剪出来，按类别归档，构建了可用于图像分类模型直接训练的数据集BIT_CLS_Dataset，其目录结构为：



(3) **类别不平衡处理:** 我们在训练分类模型时引入了加权交叉熵损失，损失权重设置为各类别最大样本数与该类样本数的比值，以提升模型对少数类的关注度。

(4) **几何特征保持:** 针对方案二的分类任务，为防止直接 Resize 导致车辆长宽比失真(例如将狭长的 Truck 压缩成方形，导致其特征与 Microbus 混淆)，实验实现了 **SquarePad(Letterbox Resize)** 预处理类。该方法通过计算图像长边，在短边填充黑色像素(0)，将图像补全为正方形后再缩放至 224x224。

3.3 方案一：端到端目标检测

3.3.1 训练策略

为适配车型识别任务的特性，我们基于 Ultralytics 框架对 YOLO11-l 模型进行微调，核心训练策略如表2所示：

表 2 YOLO11-l 训练超参数配置详解

模块	参数名称	设定值	配置策略与依据
基础配置	Epochs	100	配合 Early Stopping (Patience=10)
	Batch Size	32	适配 32GB 显存，平衡收敛速度
	Input Size	1088 ²	适配 1080p 源数据，保留小目标细节
数据增强	Mosaic	1.0	提升对遮挡与密集场景的鲁棒性
	Mixup	0.1	轻微引入以增强正则化，防止过拟合
	HSV-H/S/V	0.015/0.7/0.4	模拟复杂光照与天气变化
	Degrees	±10°	模拟车载摄像头轻微震动与视角偏差
	Translate	0.1	增强位置不变性
	FlipUD	0.0	禁用，遵循车辆行驶的物理重力约束
求解器	Optimizer	SGD	初始学习率 $lr_0 = 0.01$
	Momentum	0.937	保持训练梯度的平滑与稳定性
	Box Loss	CloU	强化边界框回归的重叠度与中心对齐

3.3.2 训练结果

基于 YOLO11-l 的微调训练共进行 95 个 Epoch，第 85 轮已得到 best.pt。训练过程中的损失变化曲线及验证集评估指标(Precision、Recall、mAP)如图 3 所示，对其分析如下：

(1) **收敛动态与高精度定位:** 训练过程显示损失函数在前 20 epoch 迅速下降，并于 80 epoch 附近进入稳态平台期，且验证集损失表现出优异的泛化能力，未出现过拟合震荡。最终模型在 mAP@0.5 上达到 0.980，更为关键的是，在严苛的 mAP@0.5:0.95 指标上高达 0.966。这一结果显著优于同类模型，我们认为是因为

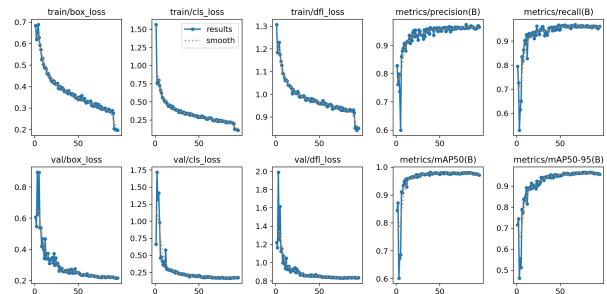


图 3 YOLO 111 的损失变化与评估指标曲线

1088 × 1088 的高分辨率输入策略发挥了作用——高像素密度保留了丰富的边缘几何细节，极大提升了边界框回归的亚像素级精度。

(2) **细粒度类别性能分析:** 结合表 3 的量化数据，我们对模型处理各类别的检测性能分析如下：

- **优势类别:** 特征显著的“Bus”和样本充足的“Sedan”表现最佳，AP50 分别达到 **0.990** 和 **0.992**，基本无漏检。

- **难点类别:** 针对长尾分布的“Minivan”(仅 400 余样本)，模型依然取得了 **0.983** 的 AP50，且 Precision 高达 **0.989**，说明模型对检出的样本判断非常准确；然而其 Recall 仅为 **0.898**(所有类别中最低)，表明小样本导致模型存在一定的漏检现象。而对于易混淆的“SUV”，模型未能展现出很强的检测能力，其 AP50 为 **0.959**(所有类别中最低)，且 Precision 仅为 **0.938**，这反映出该类别存在较高的误检率，验证了 SUV 与 Sedan 在特定视角下外观高度相似，导致模型难以区分细粒度特征。

表 3 YOLO11-l 验证集各类别详细检测指标。表中加粗数据表示最优值，下划线数据表示相对短板。

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Bus	0.978	0.980	0.990	0.973
Microbus	0.951	0.965	0.970	0.964
Minivan	0.989	<u>0.898</u>	0.983	0.968
Sedan	0.993	0.977	0.992	0.981
SUV	<u>0.938</u>	0.952	<u>0.959</u>	0.955
Truck	0.946	0.994	0.986	0.954
Overall	0.966	0.961	0.980	0.966

3.3.3 性能评估

为了全面评估方案一在未见数据上的泛化能力，我们将微调后的 YOLO11-l 模型部署于独立的测试集上进行全量推理。详细的量化评估结果汇总于表 4。

表 4 方案一 (YOLO11-l) 测试集性能量化评估

Class	Precision	Recall	AP@0.5	FPS
Bus	1.0000	1.0000	0.9950	-
Minivan	0.9509	0.9229	0.9536	-
SUV	0.9628	0.9477	0.9706	-
Sedan	0.9775	0.9848	0.9878	-
Overall (all)	0.9749	0.9634	0.9773	159.9

基于上述实验数据，我们对模型的性能表现主要归纳为以下两点：

- **泛化稳健性**：对比测试集 (**97.73%**) 与验证集 (98.0%) 的 mAP@0.5 指标，两者间的性能差异微乎其微 (< 0.3%)。这一结果强有力地证明了模型未出现过拟合现象。该优异表现主要归因于 1088×1088 的高分辨率输入策略与 Mosaic/Mixup 强数据增强的协同作用，使得模型成功捕获了车辆外观的尺度不变性特征，在面对未见样本时仍能保持高度一致的判别力。
- **长尾类别召回能力**：针对极具挑战性的少样本类别 **Minivan**，模型实现了 **92.29%** 的高召回率。这表明 YOLO11-l 的端到端架构在处理全图上下文信息时，能够有效利用全局语义特征来辅助局部目标的判别，从而在数据分布极不平衡的情况下，仍保持了对少数类样本的高敏感度，有效缓解了长尾分布带来的漏检问题。

3.4 方案二：级联式检测与识别

本方案旨在通过解耦定位与分类任务，利用专用分类网络强化对车辆细微外观特征的提取能力。首先利用官方的 YOLO11-l 模型作为定位器，根据预测框裁剪出车辆子图 (ROI)；随后经过几何特征保持预处理 (SquarePad)，送入后端的细粒度分类网络输出最终类别。本节重点探究 CNN 与

Transformer 等不同架构在车辆细粒度分类任务中的性能表现。

3.4.1 细粒度分类网络

为全面评估不同深度神经网络架构在车型细粒度特征提取上的性能差异，探究卷积神经网络 (CNN) 与视觉 Transformer(ViT) 在处理类间相似度高、类内差异大的车辆图像时的各自优势，本实验选取了四种具有代表性的骨干网络作为分类器后端：

(1) MobileNetV3-Small(轻量级 CNN 代表)

- **架构特性**：该模型作为轻量级卷积神经网络的里程碑式工作，引入了基于 **神经架构搜索 (NAS)** 优化的网络拓扑。其核心算子采用 **深度可分离卷积 (Depthwise Separable Convolution)**，将标准卷积解耦为负责“空间滤波”的深度卷积与负责“通道线性组合”的逐点卷积。这种设计在保持特征提取能力的同时，显著降低了参数量与计算复杂度。此外，MobileNetV3 进一步集成了轻量级的 **SE (Squeeze-and-Excitation) 通道注意力模块**，通过显式建模通道间的依赖关系，实现特征通道权重的自适应重标定。
- **选型依据**：在车型识别的实际应用中（如路侧嵌入式设备），计算资源往往受限。选取该模型旨在探究在极低参数量（约 2.5M）下，模型能否通过注意力机制有效聚焦车辆的关键判别区域（如车灯、格栅），从而确立本任务在低算力场景下的性能下界。

(2) ResNet50 (标准 CNN 基准)

- **架构特性**：ResNet50 是深度卷积神经网络的经典范式，其核心贡献在于引入了 **残差学习机制**。通过构建恒等映射捷径，即 $y = \mathcal{F}(x) + x$ ，该架构有效解决了深层网络中的退化问题，使得梯度的反向传播更加顺畅，从而能够训练超深层网络以提取高阶语义。作为纯 CNN 架构，它具备显著的 **归纳偏置**，即平移不变性与局部性，这使其在捕捉车辆轮廓、纹理等底层几何特征时具有天然优势。

- **选型依据**：作为一个在工业界经过广泛验证的基准模型，ResNet50 提供了极强的训练稳

定性与泛化能力。在 BIT-Vehicle 中等规模数据集上，其稳健的特征提取能力使其成为衡量其他架构（如轻量化模型或 Transformer）性能优劣的最佳 Baseline。

(3) Swin-Transformer Tiny (ViT 代表)

- 架构特性：**Swin-Transformer 代表了层级式 Vision Transformer 的前沿进展。不同于 CNN 固定的局部感受野，该模型将图像分割为 Patch 序列，并引入了独特的 **移动窗口自注意力机制**。这种设计不仅将计算复杂度限制在线性级别，还通过在连续层间移动窗口位置，实现了跨窗口的信息交互。这使得模型既保留了层级化的特征提取能力，又具备了建立像素间 **长距离依赖** 的能力，能够有效捕捉全局上下文信息。

- 选型依据：**针对数据集中 Sedan 与 SUV 等外形轮廓高度相似的细粒度分类难题，单纯依赖局部特征（如车灯、车窗）往往导致混淆。引入该模型旨在验证 Transformer 强大的全局建模能力与结构感知特性，能否通过捕捉车辆整体比例与宏观形状特征，提升细粒度分类的准确率上限。

(4) ConvNeXt-Tiny (现代化 CNN 代表)

- 架构特性：**ConvNeXt 代表了纯卷积架构在 Transformer 时代的“现代化复兴”。它并未引入复杂的注意力机制，而是借鉴了 Swin Transformer 的宏观设计理念对传统 ResNet 进行重构：包括采用大步长的 **Patchify 处理**、引入 7×7 大尺寸深度卷积核以及倒残差结构。这种设计大幅扩展了网络的 **有效感受野**，使其在保留 CNN 归纳偏置优势（如训练效率高）的同时，获得了媲美 Transformer 的全局特征捕获能力。

- 选型依据：**该模型代表了当前卷积架构设计的最新趋势。引入 ConvNeXt 旨在探究一个核心问题：在不引入自注意力机制高计算开销的前提下，通过优化卷积拓扑结构，纯 CNN 架构是否能在车型识别任务上达到甚至超越 Transformer 的性能，从而在精度与效率之间找到更优的平衡点。

3.4.2 训练策略与实验配置

为确保各模型架构在 BIT-Vehicle 数据集上的公平性对比与性能潜力的充分释放，本实验构建了标准化的训练流水线。基于 PyTorch 深度学习框架，所有分类模型均采用 ImageNet 预训练权重进行初始化，以加速特征收敛。详细的超参数配置概览如表 5 所示。

表 5 分类模型统一训练参数配置。针对 CNN 与 Transformer 的架构特性，我们在优化器与正则化策略上进行了自适应调整。

模块	参数项	设定值	配置策略与依据
输入预处理	分辨率	224×224	适配骨干网络标准输入
	填充策略	SquarePad	保持几何长宽比，防止形变失真
数据增强	水平翻转	$p = 0.5$	模拟不同视角的拍摄差异
	随机旋转	$\pm 10^\circ$	增强对摄像头安装角度的鲁棒性
	颜色抖动	0.2 (B/C)	模拟多样化的光照环境
优化器	CNNs	Adam ($1e^{-3}$)	利用归纳偏置实现快速收敛
	ViT	AdamW ($5e^{-5}$)	强正则化以稳定注意力机制训练
调度器	衰减策略	StepLR ($\gamma = 0.5$)	精细化搜索损失函数极小值
	早停机制	Patience=10	监控 Kappa 系数，抑制过拟合
损失函数	类型	Weighted CE	长尾分布重加权

特定策略说明：

- 架构自适应优化策略：**鉴于卷积神经网络与 Vision Transformer (ViT) 在优化地形与归纳偏置上的显著差异，实验采取了差异化的求解策略。对于 ResNet50 和 MobileNetV3，利用其天然的局部性归纳偏置，采用标准 **Adam** 优化器配合 1×10^{-3} 的初始学习率即可实现快速稳定收敛。相反，Swin-Transformer 由于缺乏固有的位置先验且对参数正则化更为敏感，我们选用 **AdamW** 优化器，并将初始学习率下调至 5×10^{-5} ，同时引入权重衰减 (Weight Decay = 0.05)。这种强正则化手段有效避免了训练初期的梯度震荡与局部极小值陷阱。
- 长尾类别重加权：**针对数据集呈现的显著长尾分布（例如 Sedan 样本数 $N = 5922$ 远多于 Minivan $N = 476$ ），为了防止模型在训练过程中被多数类主导，我们设计了基于类别频率的加权交叉熵损失。类别权重 W_c 计算公式如下：

$$W_c = \frac{N_{max}}{N_c} \quad (1)$$

其中 N_{max} 为最大类别样本数， N_c 为当前类

别样本数。该权重向量被集成至损失函数中，通过对错误分类少数类样本施加更大的惩罚，强制模型关注并学习尾部类别的判别性特征。

3.4.3 训练动态与验证集评估

为了保证实验的公平性与可复现性，所有模型均统一设定最大迭代轮次 (Max Epochs) 为 100，并引入 Patience=10 的早停机制以防止过拟合。训练过程中的损失函数下降趋势及验证集关键指标的变化曲线如图4 所示。此外，表 6 详尽统计了各模型在验证集上的最佳性能指标。

(1) 收敛特性与归纳偏置分析：

如图 4 所示，四种架构均展现了稳健的收敛态势，且均在 40 轮内触发早停，这表明经过预处理 (检测 + 裁剪) 后的车辆数据集具有清晰的特征分布。

- MobileNetV3 (Best Trade-off):** 该模型展现了极高的参数效率，于第 34 轮收敛。其验证集 Accuracy 高达 **97.47%**，位居榜首。这表明在 BIT-Vehicle 这种中等规模数据集上，轻量级网络较少的参数量反而降低了过拟合风险，实现了特征拟合与泛化的最佳平衡。
- ResNet50 (Stability):** 作为基准模型，其 Loss 曲线最为平滑，验证集 Accuracy 稳定在 97.17%。这得益于残差连接提供的优化稳定性，使其能够持续提取深层语义特征而无梯度退化之虞。
- Swin-Transformer (Fast Convergence):** 该模型在第 18 轮即快速收敛，且取得了最低的验证集 Loss (**0.1527**)。然而，其 Accuracy (96.97%) 略低于 CNN 模型。这种“低 Loss 但非最高 Acc”的现象表明，Swin-T 在样本预测的置信度上更为果断 (校准性好)，但在决策边界的划分上，对于部分困难样本的区分能力略逊于具备强局部归纳偏置的 CNN。
- ConvNeXt-Tiny:** 尽管引入了大核卷积设计，其最终训练 Loss (0.0077) 略高于其他模型，Kappa 系数 (0.9528) 亦为最低。这可能归因于纯卷积大核架构对数据量的依赖性较强，在缺乏海量数据预训练微调的情况下，其优势难以完全释放。

(2) 验证集性能评估：

表 6 的量化数据验证了我们提出的“级联识别流水线”的有效性：

- 高精度基线：**所有模型的验证集准确率均突破 96.9%，Kappa 系数均高于 0.95。这证明了前端 YOLO 检测与 SquarePad 预处理成功消除了背景噪声与几何畸变，为分类器提供了高质量的输入特征。
- 长尾分布的鲁棒性：**重点关注对类别不平衡高度敏感的 **Kappa 系数**，所有模型均保持在 0.95 以上的高位。其中 MobileNetV3 更是达到了 **0.9616**。这一结果强有力地证明了“加权交叉熵损失”策略的成功——它有效惩罚了多数类的过度自信，确保了模型在 Minivan 等少数类上的判别能力，避免了精度虚高陷阱。

表 6 验证集最佳性能指标统计。Epoch 指触发早停时的最佳轮次；加粗数据表示各列最优值。

Model	Epoch	Train Loss	Val Loss	Val Acc (%)	Val Kappa
MobileNetV3	34	0.0020	0.1731	97.47	0.9616
ResNet50	38	0.0021	0.1861	97.17	0.9599
Swin-T	18	0.0270	0.1527	96.97	0.9575
ConvNeXt-T	33	0.0077	0.1704	96.92	0.9528

3.4.4 测试集性能对比与模型优选

在完成模型微调与初步验证后，本节利用完全独立的测试集方案 B (包含 1002 张真实场景图片) 进行最终评测，以客观量化模型在未见数据上的泛化边界。基于表 7 和表 8 的实测数据，我们从分类可靠性、细粒度判别力及推理效率三个维度对候选架构进行深度剖析。

表 7 四种分类模型在测试集上的综合性能对比

Model	Acc (%)	Kappa	Macro-F1	Latency (ms)	FPS
MobileNetV3-S	97.11	0.9524	0.9590	4.15	240.79
ResNet50	97.80	0.9640	0.9626	5.04	198.53
Swin-Tiny	97.60	0.9605	0.9658	10.52	95.09
ConvNeXt-T	96.51	0.9430	0.9536	4.61	216.90

(1) 综合判别能力分析：

实验结果显示，**ResNet50** 在各项核心精度指标上均确立了领先优势，其测试集准确率达到 **97.80%**，Cohen's Kappa 系数高达 **0.9640**。

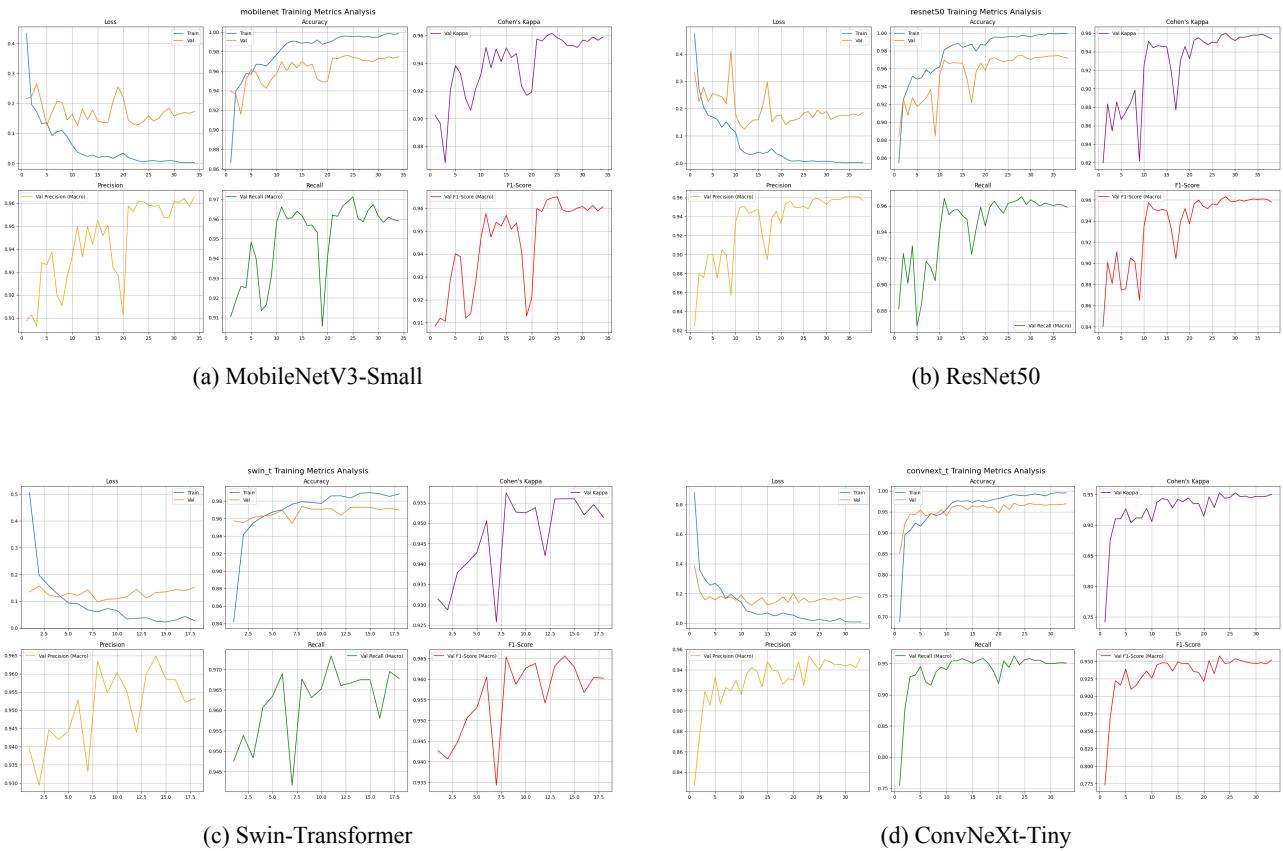


图 4 分类模型训练动态对比。图 (a)-(d) 展示了不同架构模型在训练集损失 (Train Loss) 收敛过程及验证集精度 (Val Accuracy/Kappa) 的同步变化趋势。所有模型均在 40 Epochs 内达到收敛。

- 架构优势:** 相比于现代化卷积架构 ConvNeXt-T (Acc: 96.51%) 和 Transformer 架构 Swin-Tiny (Acc: 97.60%), ResNet50 展现出更稳健的特征建模能力。这证明在 BIT-Vehicle 这种具有强几何先验的车辆数据集上, 传统的残差连接与深度卷积核能够更有效地提取具有判别性的视觉特征。

- 一致性评估:** ResNet50 的高 Kappa 系数证明该架构在处理样本分布极度不均 (Sedan 与 Minivan 样本量相差逾 10 倍) 的情况下, 预测结果与实际标签之间具有极高的一致性, 而非仅仅通过拟合多数类来获得高准确率。

(2) 细粒度与长尾类别表现:

针对车型识别中的相似类混淆 (SUV vs Sedan) 和长尾分布 (Minivan) 难题, 各模型的表现呈现出显著差异 (详见表 8):

- 相似类区分:** ResNet50 在 SUV 类别的 F1-Score 达到 **0.9579** (Recall 约为 96.7%), 显著优

于 MobileNetV3。这表明 ResNet50 能够更精准地捕捉车辆底盘高度、D 柱角度等细微几何特征, 从而有效抑制将 SUV 误判为 Sedan 的现象。

- 小样本鲁棒性:** 针对样本量最少的 Minivan 类别, 轻量级模型 MobileNetV3 取得了最高的 F1-Score (**0.9286**)。分析认为, 轻量级模型由于参数量较小, 在加权损失函数的作用下, 更不容易对多数类产生过拟合, 从而在极小众类别上表现出更灵活的捕捉能力。值得注意的是, Swin-Tiny 取得了最高的 Macro-F1 (**0.9658**), 说明自注意力机制在各类别的综合平衡性上具有微弱优势。

(3) 推理效率与硬件适配:

推理效率是系统在边缘设备部署的关键。

- 高吞吐量:** MobileNetV3 凭借极简结构实现了 **240.79 FPS** 的极速推理。ResNet50 紧随其后, 在 RTX 3090 上保持了 **198.53 FPS** 的高吞

表 8 不同架构在各车型类别上的 F1-Score 指标对比。该指标综合反映了模型在细粒度分类上的精确率与召回率平衡。

Class	MobileNetV3-S	ResNet50	Swin-Tiny	ConvNeXt-T
Bus	1.0000	1.0000	1.0000	1.0000
Microbus	0.9529	0.9770	0.9581	0.9419
Minivan	0.9286	0.9024	0.9383	0.9250
Sedan	0.9865	0.9907	0.9891	0.9822
SUV	0.9320	0.9579	0.9419	0.9177
Truck	0.9536	0.9474	0.9673	0.9548

吐量，单帧时延仅 5.04ms。

- **计算瓶颈：**Swin-Tiny 的推理 FPS 跌至 95.09 (时延 >10ms)。在级联系统中，分类时延会与检测时延累加，Swin-Tiny 显著增加了系统的整体负担，不符合高效 ITS 系统的实时性需求。

(4) 最终模型选择：

综合考虑整体精度、对易混淆类别的判别力以及推理效率，本方案最终选用 **ResNet50** 作为级联识别系统的核心分类模型。

- **精度天花板：**ResNet50 确立了本任务的性能上限 (Acc 97.80%)，确保了系统的判别可靠性。
- **最佳权衡：**尽管 MobileNetV3 在推理速度上略占优势，但 ResNet50 以极微小的时延代价 (约 0.89ms) 换取了更强的全局准确率与 SUV 判别力，是兼顾精度与实时性的最佳选择。

3.5 方案对比与综合分析

3.5.1 定量指标全面对比

表 9 系统汇总了端到端检测方案 A 与优选后的级联识别方案 B (ResNet50) 在同一测试集下的核心性能指标。数据表明，两者在整体精度上并未拉开显著差距 (mAP 97.73% vs. Acc 97.80%)，但在推理效率与特定类别的召回能力上呈现出截然不同的特性。

3.5.2 性能差异根源分析

(1) 精度表现与误差传播

- **细粒度优势：**在样本充足的 **Sedan** 类别上，方案 B 的 F1-Score (0.9907) 略优于方案 A (0.9878)。这证明了解耦后的级联架构优势：

一旦获得精准的车辆 ROI，专用的 ResNet50 分类网络能够排除背景干扰，更聚焦于车灯、进气格栅等细粒度视觉特征的提取。

- **级联误差累积：**然而，在 **Minivan** 的召回率上，方案 A 反而高出方案 B 约 **4.19%**。深入分析认为，方案 B 存在典型的“**两阶段误差累积**”问题——若前置检测器的回归框存在细微偏差 (如裁剪过紧导致车顶缺失)，将导致后续分类网络的输入特征受损，从而造成误判。相比之下，方案 A 通过端到端训练直接学习“全图到类别”的映射，对局部几何形变具有更强的容错性与鲁棒性。

(2) 计算复杂度与推理开销：

- **并行 vs 串行：**方案 A 的推理速度 (159.9 FPS) 是方案 B (84.8 FPS) 的 **1.88 倍**。方案 B 的推理流水线必须串行执行“车辆检测 → 图像裁剪/重对齐 → 分类推理”三个步骤。
- **场景扩展性：**更为关键的是，方案 B 的计算开销随画面中车辆数量呈**线性增长** (需对每个目标单独分类)，而方案 A 得益于单阶段检测器的特性，其推理耗时基本保持恒定，在大规模多目标实时监控场景下具有绝对的算力优势。

3.5.3 适用场景建议与最终选型

- **方案 A (End-to-End) 适用场景：**适用于对实时性要求极高 (如高速公路违章抓拍、车流统计)、边缘端显存资源受限、或需同时完成高帧率定位与分类任务的在线系统。
- **方案 B (Cascaded) 适用场景：**适用于非实时、对特定目标 (如 Sedan) 细粒度分类精度要求极高、或需要对识别结果进行可解释性分析 (如查看裁剪后的车辆特写) 的离线任务，例如停车场计费核验或事故现场取证。
- **最终选型结论：**综合考量本实验对智能交通系统 (ITS) 高效性、鲁棒性及部署成本的要求，本研究最终推荐 **方案 A (端到端 YOLOv11-l)** 作为核心交付方案。该方案在保持 SOTA 级精度的前提下，极大地优化了推理时延，能够以更低的算力成本应对复杂多变的实战需求。

表 9 方案 A (端到端) 与方案 B (级联式) 综合性能对比。方案 A 在推理速度与部署简易性上具有显著优势, 而方案 B 在多数类别的细粒度分类上略胜一筹。

Pipeline Strategy	Core Model	Metric (Overall)	Kappa	Sedan F1	Minivan Recall	Latency (ms)	FPS	Complexity
Scheme A (End-to-End)	YOLOv1-1	mAP@50: 97.73%	-	0.9878	92.29%	6.25	159.9	Low (Single-stage)
Scheme B (Cascaded)	YOLO + ResNet50	Acc: 97.80%	0.9640	0.9907	88.10%	11.79	84.8	High (Multi-stage)

3.5.4 实验局限性与未来优化方向

- (1) **样本瓶颈突破:** 目前 Minivan 等长尾类别的样本稀缺仍是制约精度的主要瓶颈。未来工作可引入 **生成对抗网络 (GAN)** 或基于 Unity 引擎的合成数据技术, 进行针对性的样本扩充与域适应训练。
- (2) **多任务架构融合:** 为兼顾两方案优点, 后续可探索多头任务架构。即在 YOLO 检测头内部引入更深层的、解耦的分类分支, 在不破坏单阶段推理速度的前提下, 进一步提升细粒度特征的判别力。
- (3) **边缘侧硬件加速:** 针对推荐的方案 A, 后续计划导出 **TensorRT** 引擎并进行 FP16/INT8 混合精度量化。这将进一步压缩模型体积并降低显存占用, 使其能够部署于算力更低的嵌入式终端(如 NVIDIA Jetson 系列)。

4 车牌识别

4.1 任务概述与数据集

车牌识别 (License Plate Recognition, LPR) 是智能交通系统中的核心环节。本任务的目标是在复杂的自然场景下, 实现对车牌的精确定位、几何矫正与字符识别。实验主要采用 **CCPD (Chinese City Parking Dataset)** 数据集 (包括 CCPD2019 与 CCPD2020), 该数据集涵盖了多种复杂环境 (如倾斜、模糊、雨雪天气、逆光等), 能够充分验证模型的鲁棒性。为全面评估模型性能, 我们将数据集重新划分为常规测试集 (`all_test`) 和高难度鲁棒性测试集 (`all_hardtest`)。

针对该任务, 我们自主探索了三种不同的技术路线:

1. 基于 YOLOv8-Pose 与 ResNet-CRNN 的高精度识别。
2. 基于 PaddleOCR 的工业级微调方案。

3. 基于 LPRNet 的轻量级快速识别。

接下来, 我们将深入剖析这三种方案的实现机理, 并结合实验数据对其性能进行系统性评估。

4.2 方案一: 基于 YOLOv8-Pose 与 ResNet-CRNN 的高精度识别

4.2.1 算法原理与系统架构

本方案采用“检测-矫正-识别”的端到端流水线, 旨在解决复杂场景下车牌倾斜导致的识别率下降问题。系统的网络框架图如图5所示。

1. **检测模块 (Detection):** 我们使用 **YOLOv8-Pose** 关键点检测模型。不同于传统的矩形框检测, 该模型被训练用于回归车牌的 4 个角点 (Top-Left, Top-Right, Bottom-Right, Bottom-Left), 从而精确描述车牌在空间中的几何形态。
2. **矫正模块 (Rectification):** 利用检测模块得到的 4 个角点计算透视变换矩阵 (Perspective Transformation Matrix), 将倾斜、变形的车牌图像“拉正”为标准的 160×32 矩形图像, 消除视角畸变对后续识别的影响。
3. **识别模块 (Recognition):** 采用改进版 **ResNet18 + BiLSTM + CTC (CRNN)** 架构。ResNet18 用于提取矫正后图像的视觉特征序列, BiLSTM 建模字符间的上下文语义依赖, 最终通过 CTC Loss 解决不定长字符序列的对齐问题。

4.2.2 实验设置

- **数据预处理:** 将 CCPD 原始标注转换为 YOLO 关键点格式 $(x, y, w, h, px_1, py_1, \dots, px_4, py_4)$ 用于检测训练; 同时利用透视变换裁剪出车牌图像构建 CRNN 训练集。
- **训练策略:** 方案一包括了两个训练流程, YOLOv8-Pose 负责检测模块, CRNN 负责识

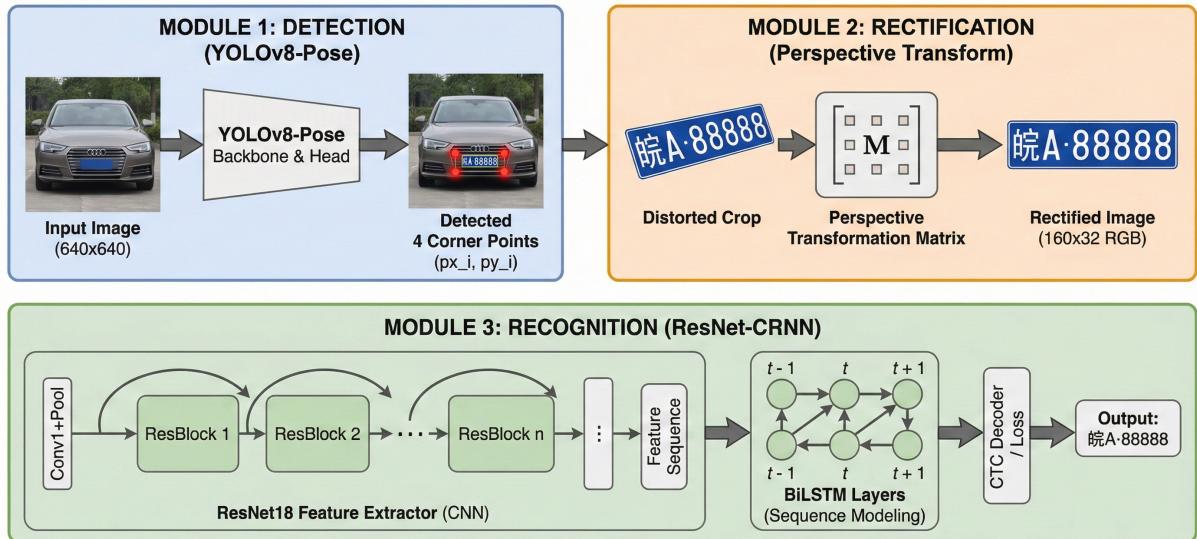


图 5 方案一系统整体架构图。该系统包含三个核心模块：(1) 基于关键点的检测模块；(2) 基于透视变换的矫正模块；(3) 基于 CRNN 的序列识别模块。

别模块，两个模块需要分别根据不同的数据集格式单独训练，具体的训练配置如表10所示。

表 10 实现细节与超参数设置。本方案包含两个训练阶段，每个阶段都有特定的配置。

Module	Model	Epochs	Batch	Specific Config
Detection	YOLOv8-Pose	50	32	Input 640 × 640
Recognition	CRNN	20	256	Adam ($lr = 1e^{-3}$)

4.2.3 实验结果与分析

模型在不同测试集上的量化性能评估如表 11 所示。

表 11 方案一 (YOLOv8-Pose + CRNN) 性能评估结果。所有测试均在单张 NVIDIA RTX 3090 GPU 上完成。

Dataset	Full Match	Char Acc.	FPS
All Test (Regular)	98.70%	99.71%	73.9
Hard Test (Robustness)	96.44%	99.38%	70.9

结果分析：实验数据表明，本方案在精度与速度之间取得了良好的平衡。首先，得益于显式的透视变换矫正，模型在 Hard Test 复杂场景下的全字匹配率高达 96.44%，与常规场景 (98.70%) 相比仅有微小的性能衰减，证明了“关键点检测 + 几何矫正”策略对于解决车牌倾斜问题的有效性。其次，系统在 Char Acc. 指标上始终保持在 99.3% 以上的高位，说明 ResNet-CRNN 架构对局部字符特征具有极强的提取能力。最后，在 RTX 3090 平

台上，模型推理速度稳定在 70 FPS 以上，能够轻松应对高帧率视频流的实时处理需求。

4.3 方案二：基于 PaddleOCR 的工业级微调方案

4.3.1 技术路线与改进

- 几何感知数据增强：**针对 CCPD 数据集中普遍存在的倾斜与形变问题，我们在数据加载流水线中集成了一个定制化的透视变换模块。该模块利用 GT 中的四个角点坐标对图像进行动态几何校正，显著增强了模型对非受控视角下车牌的特征提取能力。
- 标签标准化流水线：**针对原始数据集基于文件名的特殊标注格式，我们构建了自动化的解析与映射流水线。该流程将非结构化的文件名元数据转换为标准化的序列识别标签与字符字典，解决了异构数据源与识别模型输入之间的对齐问题。
- 域适应迁移学：**为了解决通用 OCR 模型在特定领域（中文车牌）上的适配问题，我们选取 PP-OCRv5 服务器端高精度模型作为 Backbone，采用迁移学习策略对参数进行微调。这不仅加速了模型的收敛，还有效提升了模型对汉字及车牌特定字符组合的判别力。

4.3.2 模型评估

我们分别评估了“仅识别 (Rec-only)”与“端到端 (End-to-End)”两种模式的性能，结果对比见

表 12 所示。

表 12 方案二 (PaddleOCR) 性能评估对比

Eval. Mode	Dataset	Full Match	Char Acc.	Remark
Rec-only (GT Crop)	All Test	99.30%	99.78%	Using GT Box
Rec-only (GT Crop)	Hard Test	98.21%	-	High Robustness
End-to-End	All Test	90.04%	94.05%	Inc. Det. Error
End-to-End	Hard Test	64.87%	-	Limited by Det.

结果分析:

- 识别模型的极限性能:** 在解耦检测误差的 Rec-only 模式下, 微调后的 PP-OCRv5 展现了 SOTA 级别的特征提取能力。其在 All Test 上达到了 99.30% 的全字匹配率, 即便在 Hard Test 中也维持在 98.21% 的高位。这证明了该识别器已极好地掌握了车牌字符的语义特征, 具有极强的鲁棒性。
- 检测器的性能瓶颈:** 相比之下, 模型在 End-to-End 模式的性能出现了显著衰减。在 Hard Test 中, 全字匹配率从 98.21% 骤降至 64.87%。这一巨大的性能鸿沟、表明, 通用的文本检测架构(如 DBNet)在处理车牌这种对几何边界极其敏感的目标时存在局限性。通用检测器往往无法精确回归车牌的四个角点, 导致裁剪出的图像包含背景噪声或关键字符缺失, 从而限制了强大的识别模型发挥其应有的性能。

4.4 方案三：基于 LPRNet 的轻量级快速识别

4.4.1 系统简述

本方案追求极致的推理速度与轻量化部署。系统采用 YOLOv8 进行车牌区域粗定位, 后接改进版 LPRNet 进行无需几何矫正的端到端字符识别。LPRNet 内置的空间变换网络 (STN) 能够自动学习对图像进行空间校准, 减少了对预处理算法的依赖。此外, 针对视频流数据, 设计了“时序投票 (Voting)”算法, 利用多帧结果平滑预测, 解决单帧识别抖动问题。系统的网络框架图如图6所示。

4.4.2 性能指标

模型在不同测试集上的量化性能评估指标如表13所示:

虽然 LPRNet 的绝对精度略低于前两种方案, 但其极小的参数量和无需复杂透视变换预处理的

表 13 方案三 (LPRNet) 性能评估结果

Dataset	Full Match	Char Acc.	FPS
All Test (Regular)	85.13%	93.14%	24.10
Hard Test (Robustness)	59.44%	79.59%	29.94

特性, 使其非常适合在嵌入式设备或算力受限的场景下部署。

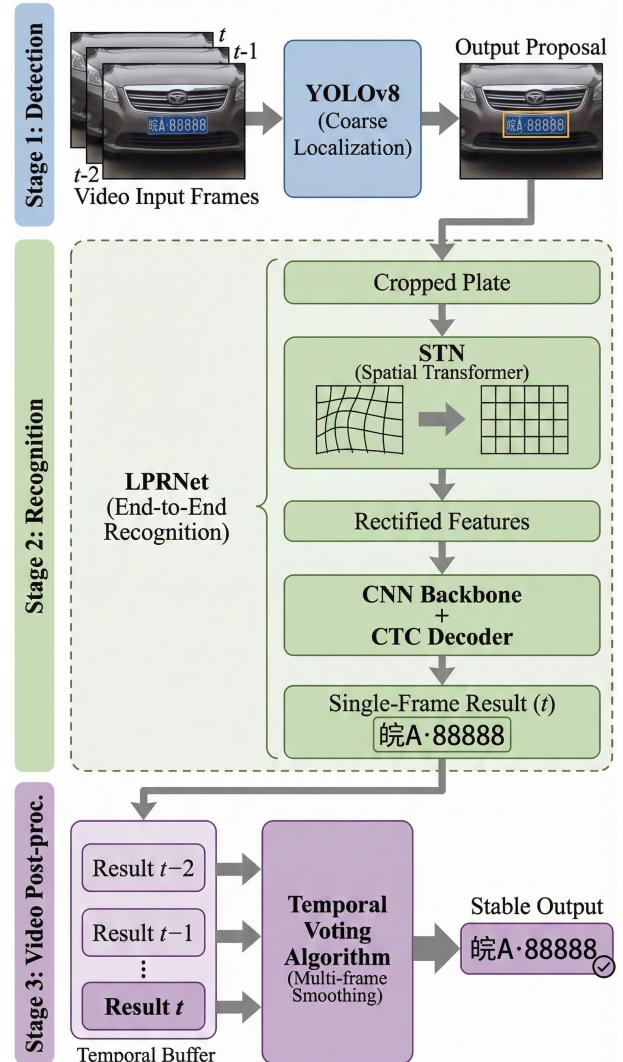


图 6 基于 LPRNet 的轻量级车牌识别系统架构图。该系统包含三个核心处理阶段: (1) 粗定位阶段: 利用 YOLOv8 从视频流中快速提取车牌区域; (2) 端到端识别阶段: 采用内嵌 STN 的 LPRNet 进行自适应空间校准与字符序列解码, 无需显式的几何矫正预处理; (3) 时序后处理阶段: 设计多帧投票机制 (Voting) 平滑预测结果, 有效解决了视频流中的单帧识别抖动问题。

4.5 实验总结与模型优选

综合对比三种方案, 我们得出以下结论:

1. 精度最佳: 方案一(YOLOv8-Pose + CRNN)。通过显式的几何矫正, 极大提升了困难场景下的鲁棒性, 是本次任务的 SOTA 方案。
2. 识别能力最强: 方案二(PaddleOCR)。其识别头(Recognition Head)的性能极强, 但受限于通用文本检测器的定位精度。未来改进方向是用方案一的 YOLO-Pose 替换方案二的 DBNet 检测器。
3. 速度与部署: 方案三(LPRNet)。适合对精度要求不苛刻但对实时性要求极高的场景。

5 车辆速度估计

5.1 任务概述与技术路线

车速估计(Vehicle Speed Estimation)是构建智慧交通系统(ITS)的核心感知环节。本任务旨在非受控的自然道路场景下, 实现对多目标车辆的实时、精确速度测量。为全面评估算法的鲁棒性, 实验构建了包含高速公路、城市快速路及复杂路口的多场景视频数据集, 涵盖了拥堵遮挡、高速运动及多类型交通工具共存等挑战性工况。

针对该任务, 我们探索了三种基于不同视觉机理的技术路线, 其系统架构对比如图 7 所示。

1. **方案一(几何视觉):** 基于 YOLOv11-l 检测与单应性矩阵投影的纯视觉测速方案;
2. **方案二(相对深度):** 基于 MiDaS 单目相对深度估计模型的测速方案;
3. **方案三(绝对深度):** 基于 MoGe V2 单目绝对深度估计模型的测速方案。

下文将结合实验数据, 对这三种方案的实现机理与性能边界进行系统性评估。

5.2 方案一: 基于单应性投影的几何测速

5.2.1 算法原理

方案一采用“检测-投影-差分”的经典几何视觉流水线(见图 7 左面板)。核心在于通过解算图像平面与物理地平面的单应性变换, 建立像素坐标系到世界坐标系的度量映射。主要包括以下四个核心模块:

1. **交互式标定(Interactive Calibration):** 选取 6-12 个地面控制点(GCPs), 利用 RANSAC 算法

鲁棒解算单应性矩阵 $H \in \mathbb{R}^{3 \times 3}$, 剔除误匹配点。

2. **检测与追踪(Detection & Tracking):** 部署 YOLOv11-l 提取车辆边界框, 并选取车辆底部中心点作为接地点假设, 利用改进 IoU 算法关联跨帧轨迹。
3. **坐标投影(Coordinate Projection):** 利用矩阵 H 将图像坐标 (u, v) 逆投影至真实世界坐标 (X_w, Y_w) , 将像素位移转化为物理位移。
4. **速度计算(Speed Calculation):** 基于世界坐标欧氏距离计算瞬时速度, 集成卡尔曼滤波器抑制检测抖动噪声。

5.2.2 实验设置与结果

- **实施细节:** 方案一无需额外训练模型, YOLOv11-l 直接加载 COCO 预训练权重。其在 RTX 5070 上推理速度达 **24.8 FPS**, 并且我们在系统内置标定验证机制, 当车道线反投影宽度误差超过 0.5m 时触发预警。

表 14 方案一实现细节

Module	Algorithm	Key Parameters	Speed	Hardware
Detection	YOLOv11l	Conf=0.5, IoU=0.45	24.8 FPS	RTX 5070
Smoothing	Kalman Filter	Window=40 Frames		

- **结果分析:** 如图 8 所示, 该方案在标定误差可控下具有极高的几何一致性与物理可解释性。它是唯一能满足实时性要求(Real-time)的方案。但是缺点在于模型在拥堵场景下检测框漂移会导致投影误差放大。

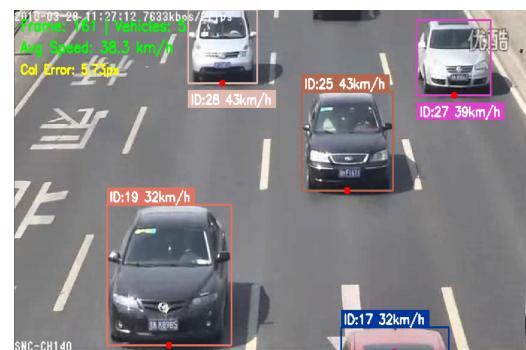


图 8 方案一实测效果。绿色边界框显示检测结果, 数值为平滑后的车速。

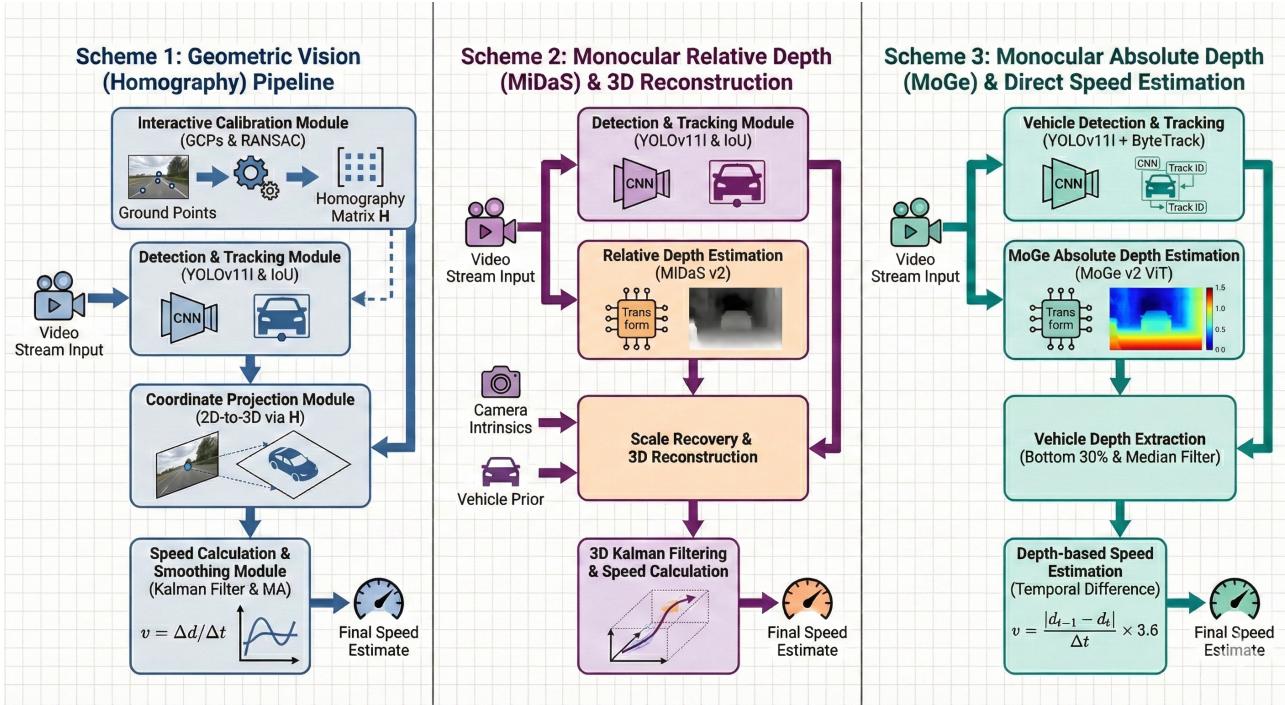


图 7 三种车速估计方案的系统架构对比。左图 (Left Panel): 方案一, 基于单应性矩阵的经典几何视觉流; 中图 (Middle Panel): 方案二, 基于 MiDaS 相对深度估计与 3D 重建流; 右图 (Right Panel): 方案三, 基于 MoGe 端到端绝对深度估计流。

5.3 方案二: 基于 MiDaS 相对深度估计

5.3.1 算法原理

方案二构建了“深度估计-3D 重建-时序差分”流水线(见图 7 中面板), 利用 MiDaS v2 恢复场景相对几何结构。模型主要包括以下三个核心模块:

- 1. 相对深度估计:** MiDaS 模型输出逆深度图, 具备良好的零样本泛化能力。
- 2. 尺度恢复与重建:** 结合相机内参及车辆先验尺寸(如轿车高 1.5m), 将相对深度映射为绝对深度, 并反投影重建 3D 空间坐标。
- 3. 3D 追踪与测速:** 在三维空间中进行卡尔曼滤波与速度解算, 解决遮挡问题。

5.3.2 实验设置与结果

- 实施细节:** 方案二采用深度-尺寸融合策略(权重 0.7:0.3)解决单目尺度不确定性。但受限于大模型的引入, 模型的整体帧率仅为 7.2 FPS(见表 15)。

表 15 方案二实现细节

Module	Algorithm	Key Config	Speed	Cost
Depth Recon	MiDaS DPT Inverse Proj.	384 ² Input Cam Height=12m	7.2 FPS	High

- 结果分析:** 方案二对拥堵场景适应性更强, 但其面临严重的工程化瓶颈: 极高的参数敏感性(依赖精确内参)和低推理速度使其难以大规模部署。此外, 长焦镜头下的深度压缩现象会导致远距离测速偏低。

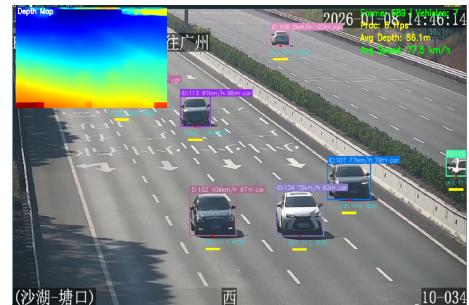


图 9 方案二实测效果。基于 3D 重建深度的测速展示。

5.4 方案三: 基于 MoGe 绝对深度估计

5.4.1 算法原理

本方案采用端到端的“免标定”测速流程(见图 7 右面板), 利用 MoGe v2 模型直接回归公制深度。模型主要包括以下三个核心模块:

- 绝对深度估计:** MoGe ViT 模型直接输出以米

为单位的深度图，解决了传统方法的尺度模糊。

2. **检测与深度提取：**集成 YOLOv11-l 与 ByteTrack，智能提取车辆底部区域 (Bottom 30%) 深度值并进行中值滤波。
3. **基于深度差分的测速：**利用前后帧绝对深度差 Δd 直接计算纵向速度，公式为 $v = \frac{|\Delta d|}{\Delta t} \times 3.6$ 。

5.4.2 实验设置与结果

- **实施细节：**使用稀疏更新策略 (每 5 帧推理一次深度) 以平衡性能，系统帧率约为 **10.2 FPS**。本方案的亮点是其完全不依赖相机标定。

表 16 方案三实现细节

Module	Model	Strategy	Speed	Innovation
Depth Tracking	MoGe v2 ViT ByteTrack	Sparse Update (5 frames) Persistent ID	10.2 FPS	Calib-Free

- **结果分析：**如图 10 所示，方案三最大的优势在于极低的用户部署成本。尽管目前速度未达实时标准且存在长焦泛化问题，但其端到端范式具有最强的技术前瞻性。

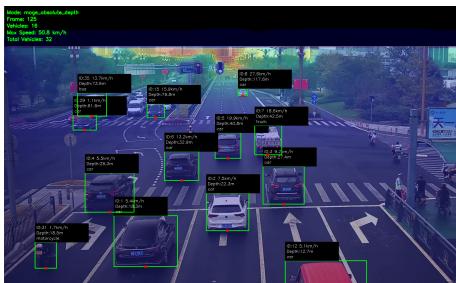


图 10 方案三实测效果。免标定绝对深度测速。

5.5 实验总结与模型优选

综合对比三种技术路线，结论如下：

1. **综合性能最佳：方案一（单应性矩阵法）。**在固定机位监控场景下，它实现了精度、稳定性与计算效率的最佳平衡 (24.8 FPS)，是当前唯一可行的工程化方案。
2. **技术前瞻性最强：方案三（MoGe 绝对深度法）。**其彻底打破了标定依赖，代表了单目视觉测速的未来。

3. **局限性：**基于深度学习的方案 (二和三) 目前仍受限于推理速度和大模型对长焦数据的域偏差。

6 UI 界面

基于前文对车型识别、车牌识别、车速识别三大核心任务的技术方案研发与性能验证，为直观呈现研究成果、推动技术落地应用，项目整合全功能的可视化用户界面 (User Interface, UI)，为用户提供高效、直观的智能交通识别交互体验。

本系统 UI 基于 Streamlit 框架构建，如图 11 所示。系统界面采用宽屏布局，并划分为左侧的“侧边栏控制面板”与右侧的“主展示区”两大核心板块。

在操作逻辑上，系统采用了高度流程化的设计，引导用户遵循“文件上传 → 功能配置 → 参数标定 (仅车速识别任务) → 启动检测 → 结果可视化与导出”的线性工作流进行操作。这种设计不仅降低了用户的上手门槛，还确保了数据处理流程的规范性。此外，系统实现了“图表联动”的可视化输出，能够同步展示检测画面识别结果与统计数据。



图 11 UI 基础界面

6.1 功能 1：车型识别

模型选择：车型识别功能采用第3.5.3节优选的端到端检测方案 A，核心模型为 YOLO11-l。该模型经 BIT-Vehicle 数据集微调优化，在兼顾 159.9 FPS 高推理速度的同时，实现了 97.73% 的 mAP@0.5 检测精度，能够精准完成 Bus、Microbus、Minivan、Sedan、SUV、Truck 六类车型的定位与分类，有效应对复杂交通场景下的细粒度识别需求与长尾分布问题。

交互逻辑：该功能支持识别图片与视频。用户在侧边栏控制面板中勾选“车型识别”选项即可激

活该模块。检测完成后，系统将在主展示区的画面中对识别到的目标绘制边界框，并在框体上方清晰标注车型类别名称。

可视化效果：除直观的图像标注外，系统在统计区提供了详尽的数据分析：

- 图片模式：采用左右分栏布局。左侧展示各检测目标车型信息的表格；右侧生成环形图，直观呈现不同车型在当前场景中的数量占比。

- 视频模式：系统将在视频播放器下方生成车型占比环形图与车流量统计图表，反映交通流中车流量随时间的变化，如图 12 所示。



图 12 车型识别数据统计可视化

6.2 功能 2：车牌识别

模型选择：针对图片与视频两种不同的输入源特性，本功能采用了差异化的模型组合策略，以确保在不同场景下均能获得最佳的识别效能：

- 图片模式：采用 4.5 节中确立的 SOTA 方案（方案一：YOLOv8-Pose + ResNet-CRNN）。该方案利用关键点检测与透视变换技术，在静态图像测试中实现了高达 99.71% 的字符识别准确率，能够有效应对倾斜、畸变等复杂车牌场景。

- 视频模式：采用方案二（基于 PaddleOCR 的工业级微调方案）。实验表明，虽然 YOLOv8-Pose 在高清静态图上表现优异，但受限于 CCPD 训练数据集的分布特性，该模型在处理视频流中尺寸较小的车牌时，关键点定位精度会出现明显下降。相比之下，基于 PaddleOCR 的工业级方案在视频流的低分辨率与小目标场景下展现出了更强的鲁棒性与适应性。

交互逻辑：用户勾选“车牌识别”后，系统后台将启动逻辑关联算法，自动计算车牌边界框与车辆边界框的空间包含关系。

可视化效果：系统根据功能组合提供两种层级的视觉呈现：

当仅开启“车牌识别”功能时，系统聚焦于车牌本体。界面将在画面中精准定位并绘制车牌区域的边界框，标签直接显示识别到的文本结果（例如：“皖 A88888”）。同时，右侧数据表将独立列出所有检出的车牌号码及其置信度，适用于专注于车牌信息的快速核验场景。

为了适配与其他功能的联动测试，UI 界面不再独立展示孤立的车牌结果，而是将识别到的车牌文本智能归属到对应的车辆实体上，实现车与牌信息的深度绑定。例如与“车型识别”功能同时使用时，车辆边界框的标签会自动扩展，以“车型 | 车牌号”的组合形式显示（例如：“Sedan | 皖 A88888”），如图 13 所示。实现了从车辆实体到身份信息的统一视觉表达，避免了画面中框体重叠造成的视觉杂乱。

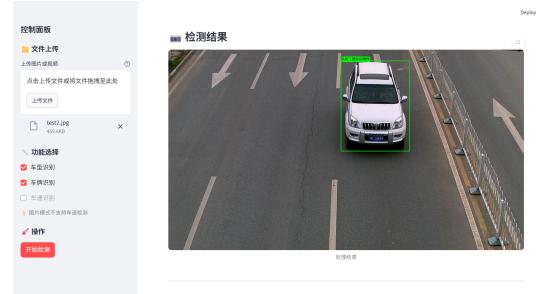


图 13 车型识别与车牌识别联动检测

6.3 功能 3：车速识别

模型选择：车速识别功能基于第 5.5 节优选的“纯视觉单应性矩阵方案”。尽管 MoGe V2 等深度估计方案具有技术前瞻性，但考虑到系统实时性与部署成本，本 UI 最终采用基于几何视觉的测速方案。该方案在 RTX 5070 平台上处理速度可达 24.8 FPS，且计算资源需求最低，能够通过建立图像平面与世界平面的精确映射，在固定监控场景下实现稳定可靠的速度估计。

交互逻辑：车速识别功能仅在视频文件模式下可用。鉴于单目测速对空间映射的依赖性，本模块设计了交互式标定流程：

1. 交互式标定：系统展示视频首帧，配合距离标定操作介绍，引导用户点击画面中的地面特征点（如车道线端点），并输入对应的相对距离坐标。
2. 精度验证：内置车道宽度验证机制，用户选取

车道左右边缘，系统计算其物理宽度并与标准值(3.75米)对比，以确保标定参数的准确性。

3. 阈值设定：支持用户自定义路段限速阈值(如60km/h)，用于后续的超速判定。

可视化效果：检测过程中，系统通过卡尔曼滤波平滑计算瞬时速度并标注在检测框上，并提供动态的视觉反馈。引入色彩编码机制，当检测速度低于限速阈值时，车辆检测框显示为绿色；一旦发生超速行为，检测框立即转为醒目的红色，实现即时违规预警。效果如图14所示。



图14 车速识别效果

7 遇到的问题及解决方案

1. 输入图像几何失真与预处理策略修正

在车型识别方案二的初期实验中，为了适配卷积神经网络对固定输入尺寸(如 224×224)的需求，我们最初采用了直接强制缩放(Direct Resize)的策略。由于原始数据集中的车辆图像长宽比各异，且大多数图像为宽大于高的矩形，直接将其压缩至正方形尺寸导致了严重的几何失真。车辆的物理特征沿特定轴向发生了非刚性形变，例如圆形的轮胎被压缩成椭圆形，车身比例严重失调。这种形变破坏了图像的空间结构特征，增加了模型学习不变量的难度。针对这一问题，我们引入了**SquarePad(类Letterbox Resize)**预处理方案。该方案遵循保持长宽比的原则，首先计算原始图像长边的缩放比例，将图像等比缩放至目标尺寸范围内，随后创建一个目标尺寸的纯色背景张量，将缩放后的图像居中填充，并在短边两侧进行像素补齐。这种处理方式确保了输入网络的图像保留了原始物体的真实几何比例，消除了由强制形变引入的特征噪声，有效提升了特征提取的有效性。

2. 类别标签索引不对齐导致的特定类别误检

在车型识别方案二的测试阶段，我们观察到一个严重的异常现象：所有分类模型的SUV与Sedan两个类别的识别准确率接近于零，混淆矩阵显示这两类之间存在系统性的严重误检。经过对数据加载流程的深入排查，发现问题的根源在于训练集数据加载器(DataLoader)在自动生成类别索引(Class Index)时产生了映射不一致。在使用datasets.ImageFolder读取数据时，框架默认根据文件夹名称的字母顺序生成类别索引。为了解决这一问题，我们尝试手动指定ds.class_to_idx为固定的字典映射，但测试结果显示问题依旧存在。

进一步分析ImageFolder的源码逻辑发现，该类在初始化阶段就已经调用内部函数生成了包含文件路径与标签元组的样本列表(ds.samples)。仅仅在初始化后修改class_to_idx属性，并不会自动更新已经生成的样本列表，导致实际读取的图像数据与修改后的标签索引依旧处于错位状态。因此，我们在代码中增加了一个关键步骤：在强制指定类别映射关系后，显式调用ds.make_dataset方法。通过执行ds.samples = ds.make_dataset(os.path.join(DATA_DIR, x), ds.class_to_idx, extensions=(...))，强制数据加载器利用正确的映射关系重新扫描并生成样本列表，从而确保了图像数据与数字标签的严格对齐。修正该逻辑后，SUV与Sedan的识别准确率恢复至正常水平(90%以上)。正确的代码如下：

```

1 for x in ['train', 'val']:
2     ds = datasets.ImageFolder(os.path.join(
3         DATA_DIR, x), data_transforms[x])
4     # 强制覆盖默认生成的类别索引映射
5     ds.class_to_idx = target_class_to_idx
6
7     # 必须加这一行，基于新的映射关系重新生成样本列
8     # 表(samples)
9     ds.samples = ds.make_dataset(
10        os.path.join(DATA_DIR, x),
11        ds.class_to_idx,
12        extensions=('.jpg', '.jpeg', '.png'))
13
14     image_datasets[x] = ds

```

3. 特征提取网络退化导致的梯度消失与时序对齐失效

在车牌识别方案一 (CRNN) 的初期探索阶段, 我们尝试使用经典的 VGG 网络作为特征提取器 (Backbone)。然而, 实验过程中出现了极端的收敛异常: 经过 7 至 15 个 Epoch 的训练后, CTC Loss 依然停滞在 2.5 左右的高位, 全字匹配准确率 (Full Match Accuracy) 持续为 0%, 模型完全未能学习到车牌序列的有效特征。经过深度分析, 我们确定了导致该失败现象的两个根本原因: 首先, VGG 这种直筒型深层网络在缺乏残差连接的情况下, 极易在反向传播过程中遭遇梯度消失, 导致 RNN 层传回的梯度无法有效更新浅层卷积核; 其次, 也是更为致命的, 是下采样率与序列长度的不匹配。标准 VGG 网络通常进行 5 次下采样 (Stride=32), 对于输入宽度为 160 的车牌图像, 最终特征图宽度仅为 $160/32 = 5$ 。根据 CTC 算法原理, 时间步长 T 必须满足 $T \geq 2L + 1$ (其中 L 为标签长度, 车牌通常为 7-8 位), 显然 $T = 5$ 根本无法提供足够的时序空间来容纳目标序列, 直接导致 CTC Loss 无法计算有效对齐。

针对上述问题, 我们采取了 ResNet18-CRNN 架构升级方案。具体改进措施如下:

1. 引入残差架构重塑梯度流: 利用 ResNet 的 Shortcut 机制构建“梯度高速公路”, 确保深层监督信号能直接传递至浅层, 彻底解决了梯度消失问题, 使模型在第 1-2 个 Epoch 即开始快速收敛。
2. 关键的步长适配: 我们对 ResNet18 的 Layer3 和 Layer4 进行了针对性修改, 将其在宽度方向的步长从默认的 2 调整为 1。这一改动将网络的总下采样率从 32 倍降低至 4 倍。对于 160×32 的输入, 输出特征图尺寸变为 40×1 。由此得到的序列长度 $T = 40$ 远大于 CTC 所需的最小长度, 为字符与 Blank 符号的对齐留出了充足的空间。
3. 序列建模增强: 部署双层双向 LSTM (Bi-LSTM), 进一步增强模型对车牌字符上下文依赖关系的捕捉能力。

经过上述架构重构, 模型性能实现了质的飞跃: 在常规测试集上的全字匹配率从 0% 飙升至 **98.70%**, 即使在包含倾斜、模糊样本的困难测试集上也能保持 **96.44%** 的高鲁棒性。这一结果有力

证明了在 OCR 任务中, 保留足够的特征图空间分辨率与采用现代化的残差骨干网络是成功的关键。

4. 通用文本检测模型的误检与修正

在车牌识别方案二中, 我们采用了 PaddleOCR 的通用检测模型作为前端定位器。由于该模型旨在尽可能召回图像中的所有文本区域, 缺乏对“车牌文本”与“环境文本”的语义辨别能力, 导致在复杂场景下, 车身上的喷漆广告(如“物流”、“核载”)或背景中的路边标语经常被误检为车牌, 严重影响了系统的精确率。针对这一语义层面的混淆问题, 我们设计并实现了一套基于先验知识的多维特征过滤算法:

- **几何约束:** 利用车牌固有的物理形状特征, 计算检测框的长宽比。我们将有效阈值设定在 [2.0, 14.0] 的宽容区间以兼容一定角度的倾斜, 直接剔除正方形图标 ($\text{Ratio} \approx 1$) 或竖排文字干扰。
- **正则加权:** 引入宽松的正则表达式 ([汉字][A-Z][字符]) 作为先验知识。在评分阶段, 符合车牌格式的候选框将获得高额的权重加分, 确保在多文字框并存时, 即使车牌清晰度略低, 其优先级也始终高于清晰的非车牌文字。

5. 视频流识别结果的时序不稳定

在车牌识别方案二的视频流处理中, 受帧间光照波动、运动模糊及局部遮挡的影响, 同一车辆的识别结果表现出极强的时序不稳定性。在连续帧中, 识别结果常在“正确车牌”、“形近错别字”与“乱码”之间跳变, 导致用户体验极差。为了解决这一问题, 我们利用视频的时序冗余性构建了基于 ID 追踪的质量评估与缓存机制:

1. **状态绑定:** 利用 ByteTrack 算法分配的 `Track_ID` 作为索引, 在内存中维护一个全局字典 `best_plate`。
2. **综合质量评估:** 定义了一个综合评分函数 S_{final} 来衡量当前帧识别结果的可靠性, 而不仅仅依赖 OCR 模型的原始置信度:

$$S_{final} = S_{ocr} + W_{regex} + W_{geo} \quad (2)$$

其中, S_{ocr} 为 OCR 模型输出的置信度, W_{regex} 为符合车牌正则的奖励权重, W_{geo} 为基于车辆底部位置的先验加分。

3. 最优解更新策略: 系统逐帧计算当前识别结果的综合评分。仅当当前帧评分 > 历史最高评分时, 才更新缓存并刷新 UI 显示。该机制确保了屏幕上始终展示该车辆出现以来最清晰、最符合车牌特征的那一帧结果。

6. 车速估计结果的强跳跃性、不稳定性

在单目视觉车速估计系统的初期实现中, 原始方案采用简单的两帧差分法计算瞬时速度: $v = \frac{\Delta d}{\Delta t} \times 3.6$, 其中 Δd 为相邻两帧间的位移, $\Delta t = 1/fps$ 。该方法对检测框的微小抖动、投影变换误差以及车辆检测的边界框波动极度敏感, 导致计算出的速度值剧烈波动, 有时甚至严重超出合理范围, 使系统输出的车速值不稳定、帧间跳跃性极强。为了解决这一问题, 我们实施了多层次的多级平滑与融合策略:

1. 空间层面的坐标滤波

在坐标转换层引入卡尔曼滤波器进行状态估计与预测。针对每个车辆目标, 我们建立了一个 4 状态(位置 x, y 和速度 v_x, v_y)、2 观测(位置 x, y)的线性动态系统模型:

状态转移矩阵:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

观测矩阵:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (4)$$

每帧执行预测-校正步骤: $\hat{x}_k = \mathbf{F}x_{k-1}$ (预测), $x_k = \hat{x}_k + \mathbf{K}_k(z_k - \mathbf{H}\hat{x}_k)$ (更新), 其中 z_k 为当前帧观测到的世界坐标。该滤波器有效抑制了由单应性变换误差和检测框抖动引起的坐标噪声, 为后续速度计算提供了平滑的位置序列。

2. 时间层面的多帧平均

摒弃完全两帧差分的方案, 引入基于历史轨迹的多帧平均方法计算速度。对于每个车辆, 维护其最近 N 个滤波后的世界坐标点(实践中 $N = 5$)。

设历史点的时间戳序列为 t_1, t_2, \dots, t_N , 对应位置为 $(x_1, y_1), \dots, (x_N, y_N)$ 。计算整体位移和总时间:

$$\Delta d_{total} = \sqrt{(x_N - x_1)^2 + (y_N - y_1)^2}, \quad (5)$$

$$\Delta t_{total} = \frac{t_N - t_1}{fps} \quad (6)$$

得到多帧平均速度 $v_{avg} = \frac{\Delta d_{total}}{\Delta t_{total}} \times 3.6$ 。同时保留两帧差分速度 v_{inst} 作为次要依据。最终采用加权平均, 赋予平均速度更高权重:

$$v_{final} = 0.7 \times v_{avg} + 0.3 \times v_{inst} \quad (7)$$

3. 速度层面的滑动窗口平滑

为每个车辆维护一个长度为 40 的滑动窗口队列, 存储历史速度值。当前帧输出的最终速度为窗口内所有速度的均值:

$$v_{output} = \frac{1}{M} \sum_{i=1}^M v_i, \quad M = \min(40, \text{历史帧数}) \quad (8)$$

4. 物理约束层面的合理性校验

引入速度合理性检查机制, 包括:

- **范围校验:** $0 \leq v \leq 150 \text{ km/h}$
- **加速度校验:** $|\frac{\Delta v}{\Delta t}| \leq 50 \text{ km/h/s} \approx 13.9 \text{ m/s}^2$
- **连续性校验:** 当前速度与历史平均速度的偏差不超过阈值

对于不合理的速度值, 系统自动回退到历史平滑值, 避免异常输出。

8 不足与改进

经过对实验结果的深入分析及真实场景视频流的测试验证, 我们发现当前系统在 BITVehicle 数据集划分策略及特定类别的样本分布上存在显著的局限性。这些不足在一定程度上限制了模型的泛化能力与鲁棒性。

8.1 基于随机划分导致的数据泄露风险

问题分析: BIT-Vehicle 数据集源自道路监控视频的截帧, 具有极强的时空相关性。同一辆车往往在连续的多帧图像中出现, 且这些图像在光照、角度和背景上的差异极小。在当前的数据预处理流程中, 我们采用了基于图像数量的随机划分策略(Random Split)。这种策略忽略了样本间的时序关联, 极易导致数据泄露: 即同一辆车的第 t 帧



图 15 BIT-Vehicle 数据集中 Truck 类别的图像实例

图像被划入训练集，而高度相似的第 $t + 1$ 帧图像却出现在测试集中。在这种情况下，模型实际上是在“记忆”特定的车辆实例，而非学习通用的车辆特征。这导致测试集上的评估指标（如 mAP 和 Accuracy）虚高，无法真实反映模型在未见车辆上的泛化性能。

改进方向：未来的实验应摒弃简单的随机划分，转而采用基于车辆 ID 或视频序列的划分策略。

- **按车辆 ID 隔离：**确保同一辆车的所有图像序列要么全部分配在训练集，要么全部分配在测试集，严格保证训练集与测试集的样本互斥。
- **去重与稀疏采样：**对原始视频序列进行关键帧提取，去除冗余的连续帧，降低样本分布的重复性，强迫模型关注车辆的本质结构特征而非背景噪声。

8.2 部分类别特征分布偏差

问题分析：在视频流实测中，我们发现“Truck”类别的检测误报率较高，且对侧面视角的卡车识别能力极差。经排查，主要原因是 BIT-Vehicle 数据集中 Truck 类别的样本存在严重的视角偏差。由于数据采集自固定的监控机位，绝大多数卡车样本仅包含正面车头的特写截面（如图15所示），缺乏完整的车身及侧面、背面视图。这种单一的特征分布导致模型产生了错误的归纳偏置：它倾向于将“巨大的矩形平面结构”或“充满视野的纹理”误判为卡车。因此，在视频检测中，当画面出现大面积背景或非典型角度物体时，模型极易产生严重的误检。

改进方向：为了修正模型对特定类别的认知偏差，需从数据多样性和模型训练机制两方面入

手：

- **多视角数据增强与扩充：**引入外部数据集（如 UA-DETRAC 或 COCO），专门补充卡车、客车等大型车辆的侧视、后视及全景图像。通过构建**多视角数据集**，使模型学习到卡车从车头到车厢的完整拓扑结构。
- **高级数据增强策略：**在训练阶段增加 **Mosaic** 和 **Mixup** 的使用频率。通过将多张图像拼接，模拟远近不同的视野，防止模型因物体充满画幅而产生对尺度的过拟合。
- **困难样本挖掘：**针对视频中误检的“全屏大画面”背景，将其加入负样本集进行再训练，抑制模型对单纯大面积纹理的错误响应。