

基于道路标记语义分割 的车辆测速技术研究



曹博豪

目录

C O N T E N T S

- 01 选题背景
- 02 道路标记语义分割算法设计
- 03 车辆测速算法设计
- 04 软件平台设计与实现
- 05 总结与展望

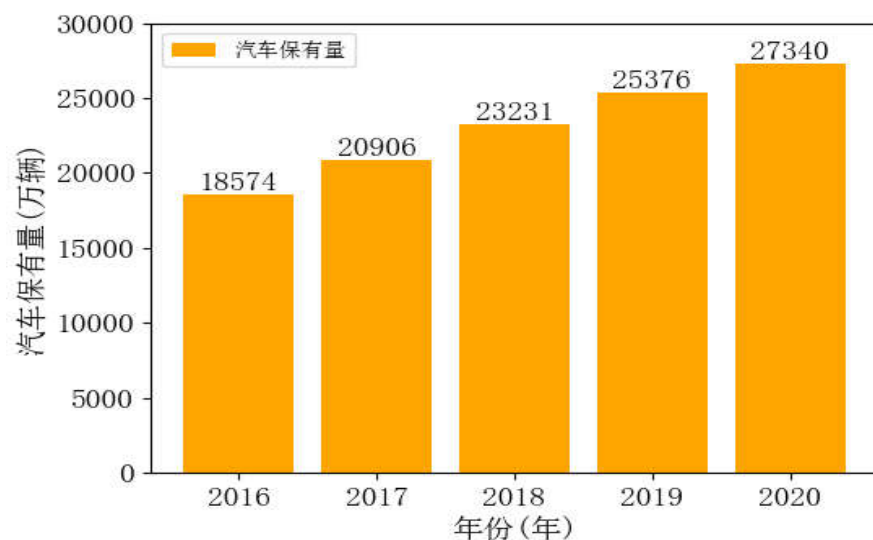


01

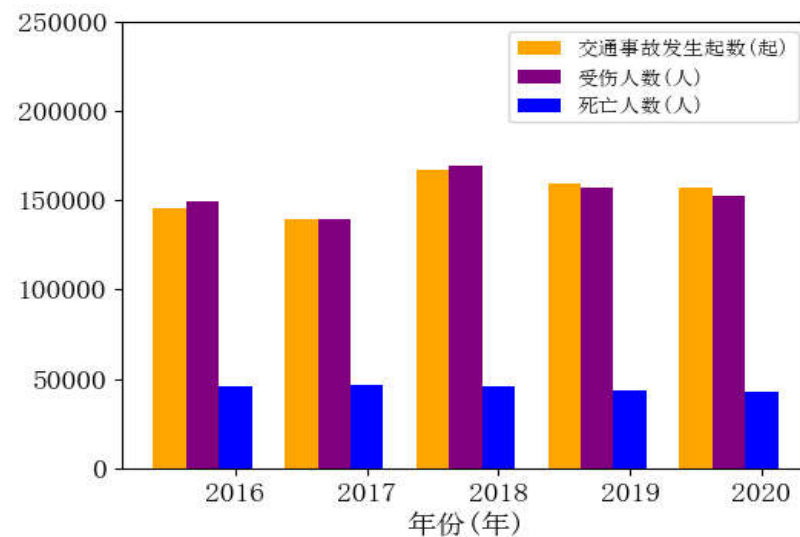
选题背景

1.1 选题背景

随着国家的飞速发展,道路上行驶的车辆数量逐渐增多。汽车虽然带来了便利,但由于道路建设以及驾驶员疏忽等问题,交通事故的数量也始终居高不下。



2016-2020年汽车保有量情况



2016-2020年汽车类型交通事故起数以及伤亡人数

1.2 视频测速研究现状

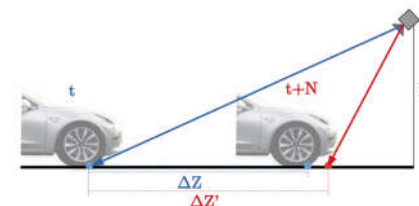
视频测速算法

基于固定式监控视频的测速

将相机标定、目标识别、目标跟踪等技术结合起来实现对车辆的定位和测速。

优点：可直接应用到现有监控视频

缺点：需要相机标定、测距的精度范围较窄

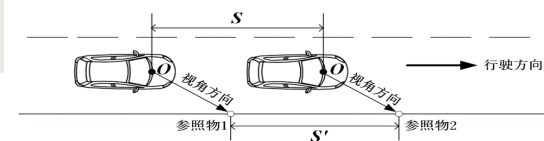
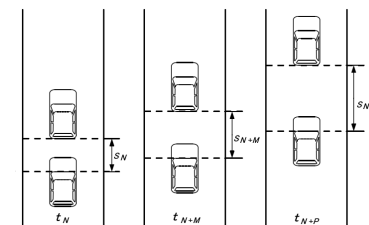


基于车载视频的测速

测前车车速时与固定式监控测速相似。测自车时，需要借助道路参照物手动进行车速测量。

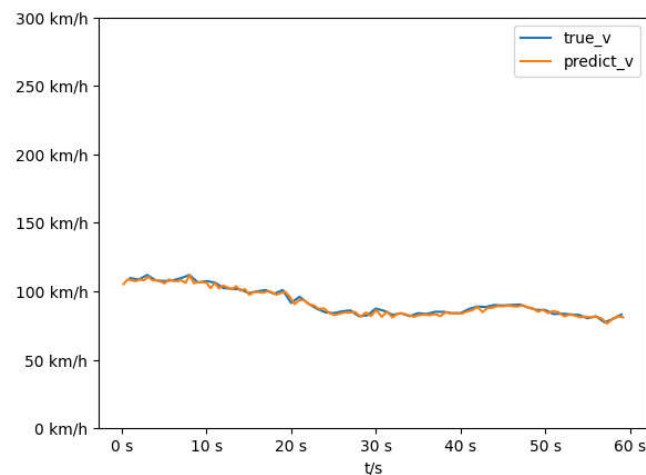
优点：可以通过车载视频判断车辆运行状态，相比于监控视频会更加方便

缺点：测前车车速时需要用到相机标定技术、测自车时由于是人工，因此存在效率低、受经验影响等问题



1.3 研究目的

研究目的：针对目前车载视频测速多采用人工手动测速的方法，效率低，费时费力且受人工经验影响较大的问题，提出一种基于道路标记语义分割的车辆行驶速度计算方法。利用语义分割的图像作为测速图像，自动获取测速所需参照物信息，最终完成对整个视频中车速的测量，方便鉴定人员对车辆的整体行驶过程进行分析。



测速示例

1.4 主要工作

一、数据集

完成道路标记图像的采集和标注工作，建立适用于语义分割的道路标记数据集并进行数据增强。

01

02

二、网络模型

搭建适用于道路标记分割的语义分割网络(Deeplabv3+网络)，并针对训练结果对网络进行改进。

三、车速测量

参考手动测速思想，结合语义分割网络模型、车行道分界线的消失位置实现车辆的自动测速。

03

04

四、软件平台

基于Flask框架搭建车辆检测和车流量统计软件平台，展示算法效果。



02

道路标记语义分割算法
设计

2.1 数据集建立

车辆在行驶过程中，道路上存在的标记有很多种类。除了常见的车行道分界线和车行道边缘线之外，还有很多标记。通过对车辆行驶道路的分析，本文选择了8种道路标记进行分割，分别为车行道边缘线，车行道分界线，人行横道线，停止线，指示直行标记，指示向前或向右转标记，指示向前或向左转标记，双黄线。



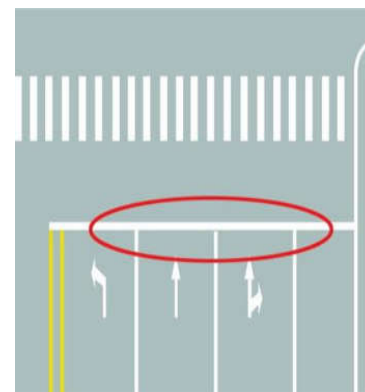
车行道分界线



车行道边缘线



双黄线

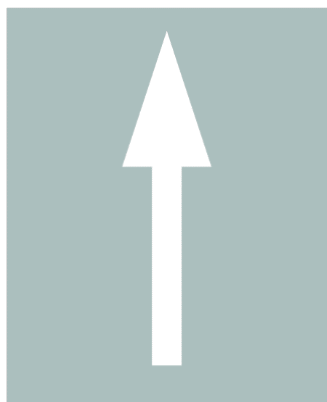


停止线

2.1 数据集建立



人行横道线



指示直行



指示前方可直行或左转



指示前方可直行或右转

2.1 数据集建立

本文总共收集了15组行车记录仪视频，均为同一辆车通过不同路段的视频。其中前10个视频用作训练集和验证集的制作。后面5个视频中，视频11用来对语义分割模型效果进行评判。剩下的4个，作为测速算法的测试视频。每个视频时长3分钟，帧率30fps。每个视频以25fps进行采样，总共得到的原始数据集为2160幅图。



Video_1



Video_2



Video_3



Video_4



Video_5



Video_6



Video_7



Video_8



Video_9



Video_10

训练集和验证集视频



Video_11



Video_12



Video_13



Video_14



Video_15

测试集

2.2 数据增强

原图



水平翻转



亮度



对比度



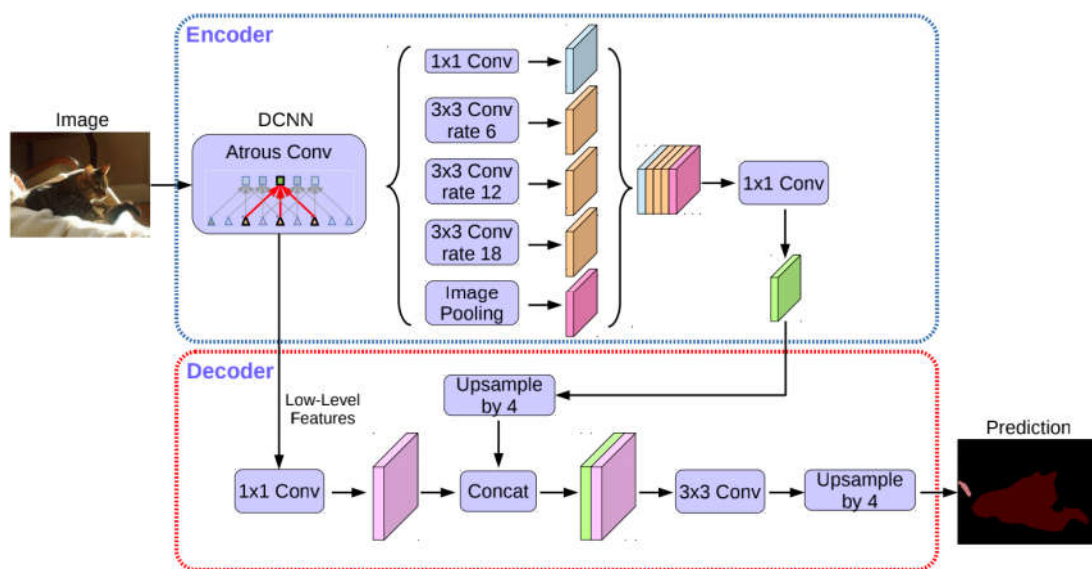
噪声



平移



2.3 网络初步训练-网络模型



DeepLab v3+ 网络结构：

- ✦ 由编码器和解码器构成
- ✦ 编码器结构包含两部分：DCNN和多空空间金字塔池化结构ASPP结构。其中DCNN代表特征提取网络，可采用Xception、ResNet等网络。
- ✦ 通道数叠加实现特征融合
- ✦ 端到端的语义分割网络

特点：

- ✦ **优点：**分割精度高、对位置及细节信息敏感
- ✦ **缺点：**网络层数深、计算量大、对硬件的运算性能有较高的要求

2.3 网络初步训练-训练参数配置

训练环境配置：

- ✦ 计算机系统：Ubuntu 18.04
- ✦ CPU: Intel(R) Core(TM) i7-10700KF CPU @3.8GHZ
- ✦ GPU：NVIDIA Geforce RTX 2080 Ti
- ✦ 运行内存：32GB
- ✦ 软件平台：Pycharm 2019
- ✦ 深度学习框架：Pytorch 1.8.0

训练参数设置

网络模型	DeepLab v3+ (ResNet)
数据集数量	12949
训练集数量	10359
验证集数量	2590
数据集尺寸	960×540
初始学习率	0.007
学习率更新策略	poly
批尺寸	4
训练周期数	100

2.3 网络初步训练-评价方式

01

主观感受

关注道路标记分割的准确性以及精细程度。

02

客观评价指标（mIoU）

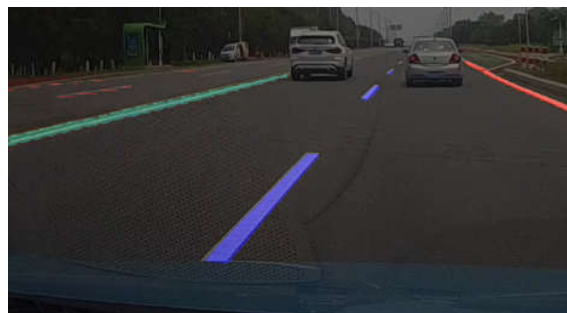
通过计算每个类别IOU值的平均值得到，其中IOU为两个集合的交集和并集之比，这两个集合是真实标签图中标记像素（或背景像素）的集合和DeepLab v3+网络分割结果中标记像素（或背景像素）的集合。

语义分割任务中标准的评价指标，用来衡量该模型分割的准确度，即网络分割结果与标签图的吻合程度。

$$mIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{p_{ii}}{\sum_{j=0}^k p_{ij} + \sum_{j=0}^k (p_{ji} - p_{ii})}$$

2.3 网络初步训练-实验结果分析

类别	颜色
车行道边缘线	红色
车行道分界线	蓝色
人行横道线	白色
停止线	绿色
指示直行	黄色
指示前方可直行或左转	紫色
指示前方可直行或右转	棕色
双黄线	深青色



分割不精确



误检

2.3 网络初步训练-实验结果分析

网络模型	背景	车行道边缘线	车行道分界线	人行横道线	停止线	指示直行	指示前方可直行或左转	指示前方可直行或右转	双黄线	mIOU
Deeplabv3+(resnet101)	0.99	0.68	0.74	0.82	0	0.79	0.32	0	0.67	0.557

从交并比的分数来看，也可以发现停止线以及指示前方可直行或右转标记的交并比分数为0，因此这两个类别不可能被分割正确。并且从平均交并比分数来看，整体分数偏低，因此网络需要改进和优化。

2.4 网络模型优化-权重平衡

类别不平衡问题指的是不同类别训练样本差别很大，这样对于网络模型的学习过程就会造成影响。例如在二分类问题中，有998个反例，但是正例只有两个。那么学习方法只需要返回一个永远将新样本预测为反例的模型，就可以达到99.8%的精度。但是这样的模型没有任何价值，因为不能够预测出任何的正例。

通过对目前数据集各类别图像进行分析发现停止线还有指示标志由于仅在路口处出现，因此在数据集中所占比例很低。因此为了减少类别不平衡的影响，这里使用带权重的loss函数来对网络进行训练。

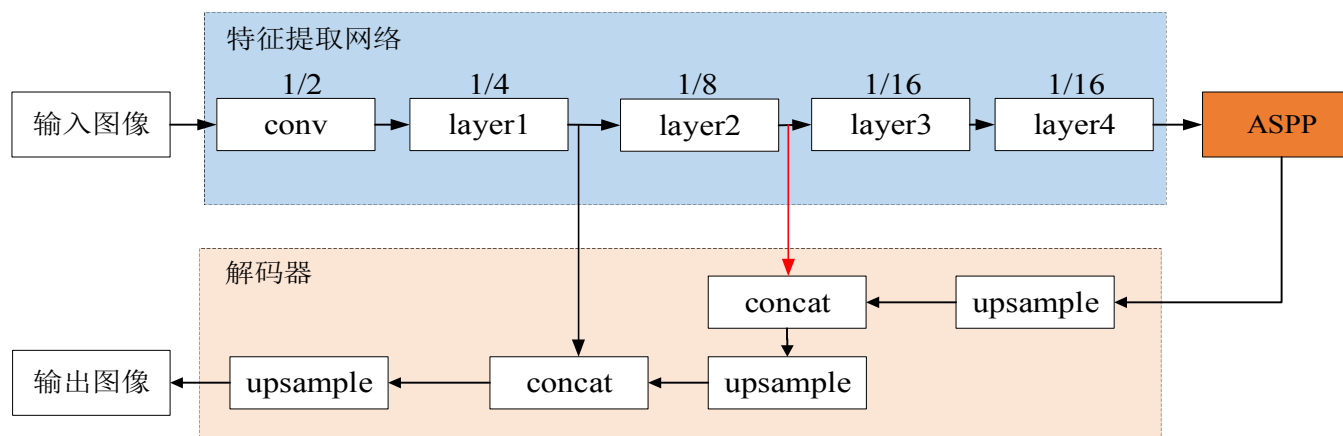
$$L = -\sum_{c=1}^M w_c y_c \log(p_c) \quad \text{loss函数}$$

$$w_c = \frac{1}{(\log(1.02) + (\frac{N_c}{N}))} \quad \text{权重计算}$$

2.4 网络模型优化-特征融合

在语义分割中高层的特征包含较好的语义信息，而低层特征语义信息较弱，但是特征尺度大，包含大量的细节信息。若是将高层的特征与低层特征进行融合，则会使得高层特征利用低层特征信息，进而达到更好的分割性能。

在本文中，因为车道标记属于小目标，而高层特征的大小实在太小，若是直接与1/4图像进行融合边缘信息会不够清晰，因此本文在原deeplabv3+网络的基础上又加入了一层特征融合，将编码器输出的1/16大小的特征图先与1/8大小的特征图进行融合，之后在与1/4大小的特征图融合，逐步恢复到原图大小。



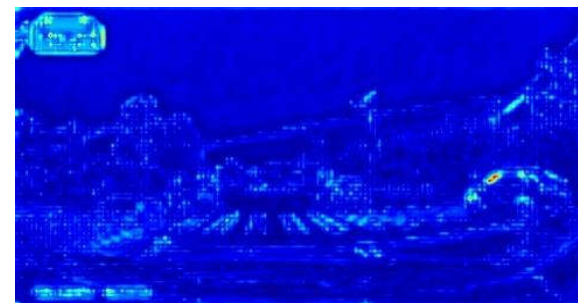
2.4 网络模型优化-特征融合



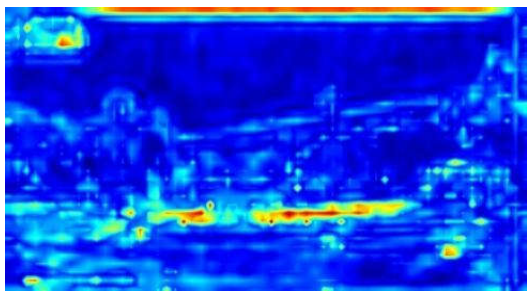
原图



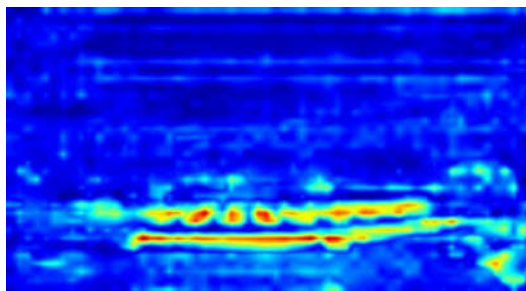
layer1输出



layer2输出



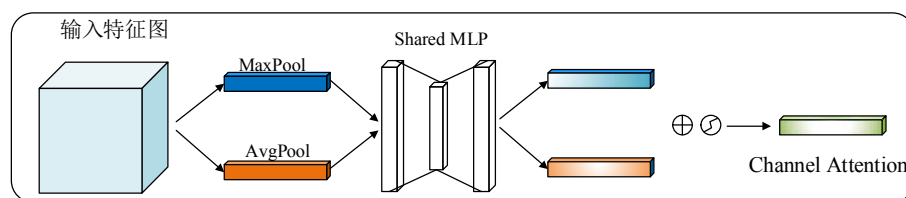
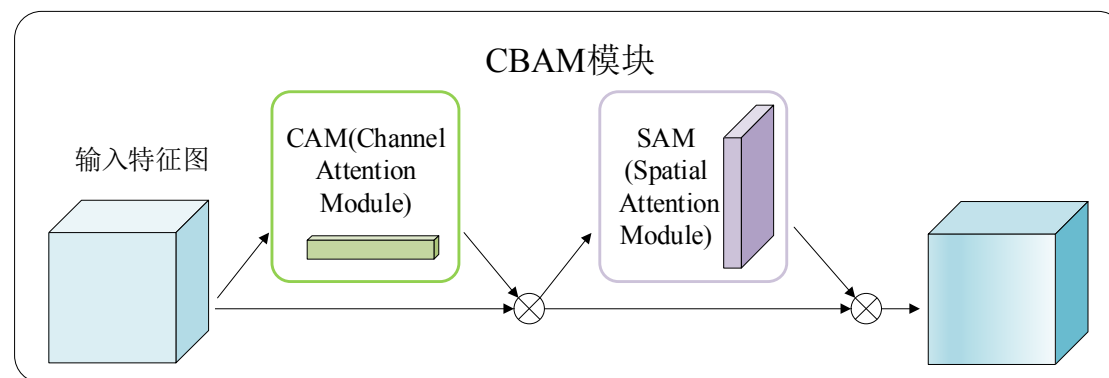
layer3输出



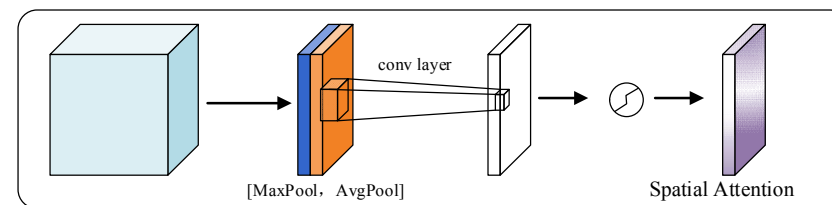
ASPP输出

2.4 网络模型优化-注意力机制

注意力机制是机器学习中的一种数据处理方法，广泛应用在自然语言处理、图像识别以及语音识别等各种不同类型的机器学习任务中。主要的作用就是帮助网络学习图像中应该关注的区域，不会将所有的注意力平等分配给图像中的所有像素。



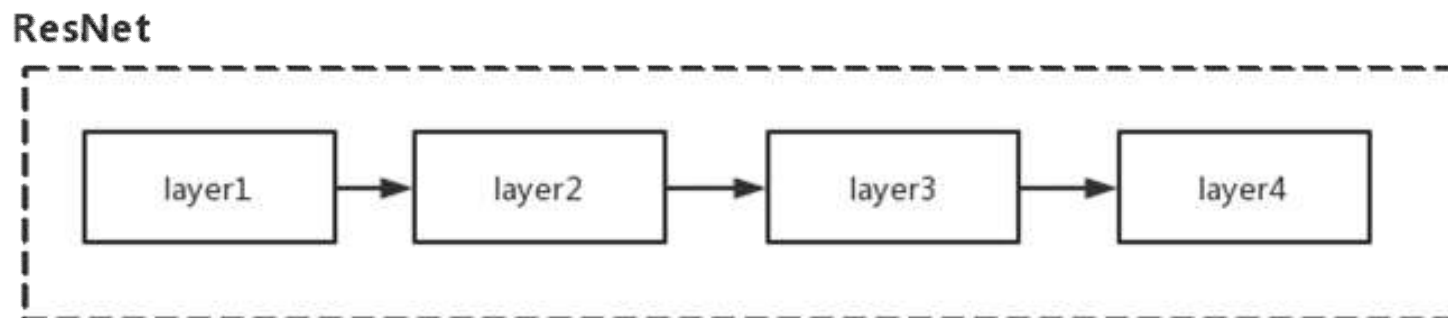
CAM模块



SAM模块

2.4 网络模型优化-CBAM模块的位置

本文采用CBAM模块主要是为了提高骨干网络Resnet的特征提取能力。对于CBAM模块在特征提取网络中的位置，做了两个实验。分别把CBAM模块放在Resnet网络每一层中bottleneck后面，以及将CBAM模块放入Resnet中每一个layer的后面。



ResNet网络结构简图

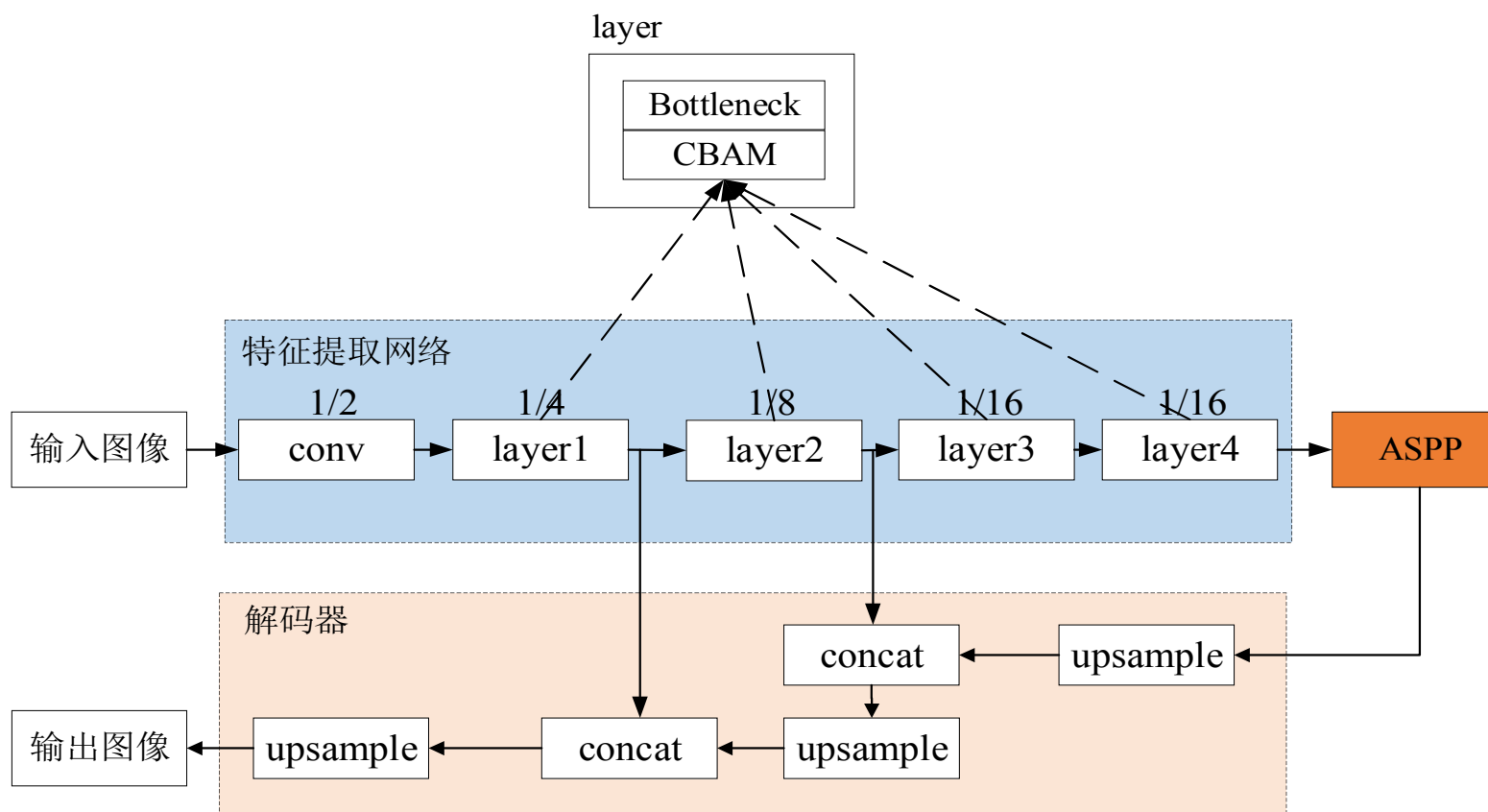


将CBAM模块加入
bottleneck后面



CBAM模块加到
layer输出后

2.4 网络模型优化-最终的神经网络模型



2.4 网络模型优化-模型性能评估

(a) GroundTruth

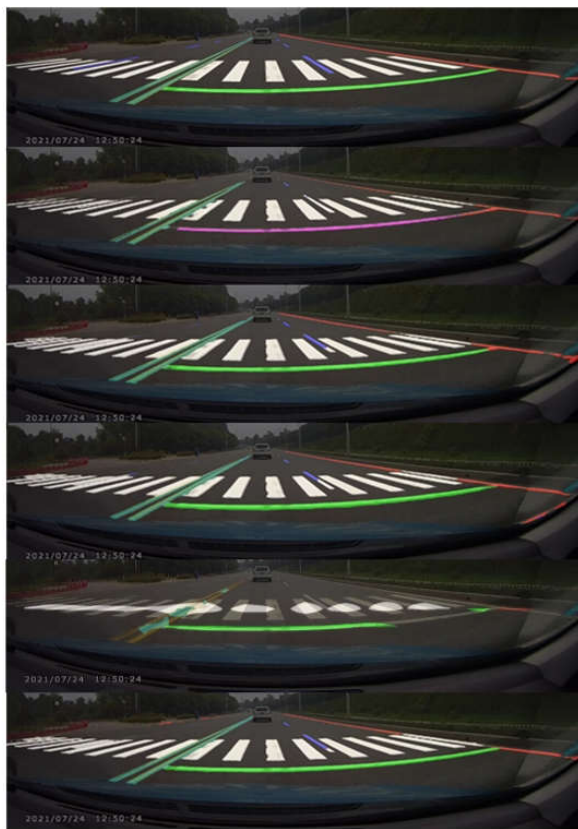
(b) Deeplabv3+

(c) Balanced

(d) Balanced+slf+cbam-final

(e) PSPNet

(f) 本文网络



(a) GroundTruth

(b) Deeplabv3+

(c) Balanced

(d) Balanced+slf+cbam-final

(e) PSPNet

(f) 本文网络



2.4 网络模型优化-模型性能评估

(a) GroundTruth

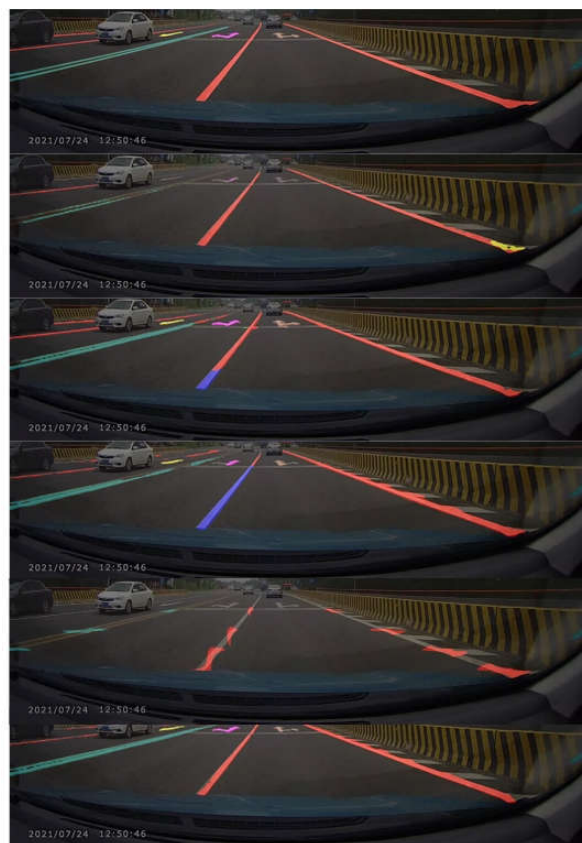
(b) Deeplabv3+

(c) Balanced

(d) Balanced+slf+cbam-final

(e) PSPNet

(f) 本文网络



(a) GroundTruth

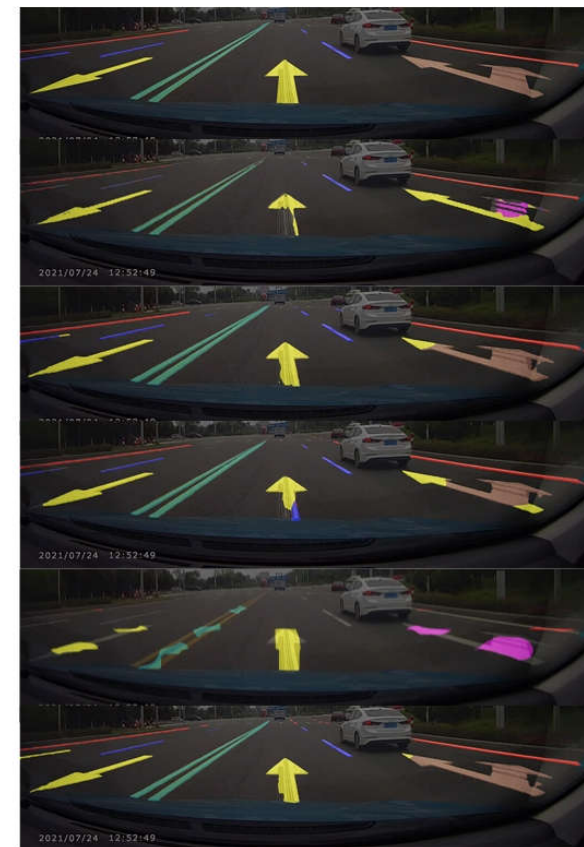
(b) Deeplabv3+

(c) Balanced

(d) Balanced+slf+cbam-final

(e) PSPNet

(f) 本文网络



2.4 网络模型优化-模型性能评估

网络模型	背景	车行道边缘线	车行道分界线	人行横道线	停止线	指示直行	指示前方可直行或左转	指示前方可直行或右转	双黄线	mIOU
Deeplabv3+(resnet 101)	0.99	0.68	0.74	0.82	0	0.79	0.32	0	0.67	0.557
balanced	0.98	0.58	0.64	0.71	0.52	0.73	0.71	0.73	0.59	0.688
balanced+slf+cbam-final	0.99	0.58	0.74	0.76	0.55	0.72	0.58	0.66	0.61	0.688
PSPNet	0.99	0.25	0.29	0.35	0.28	0.52	0.24	0.05	0.19	0.35
本文方法	0.99	0.69	0.79	0.81	0.61	0.81	0.80	0.80	0.71	0.779

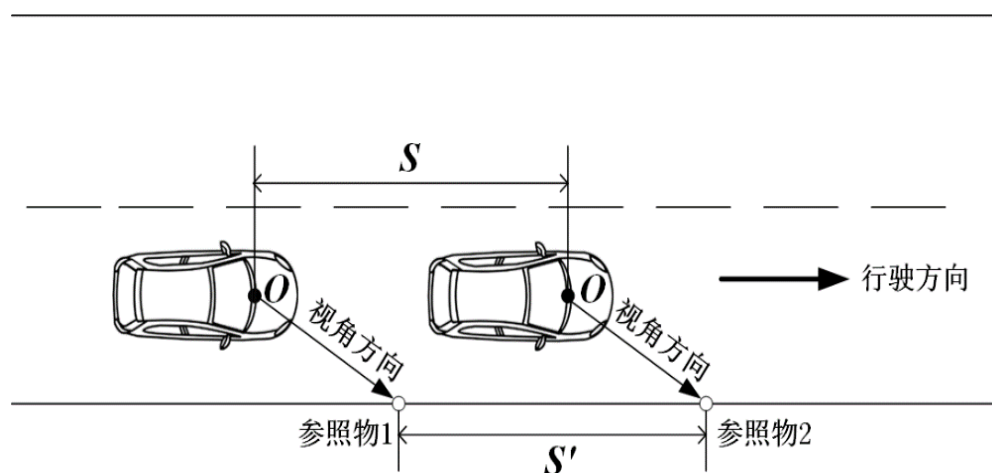


03

车辆测速算法设计

3.1 传统测速算法

目前大多数的视频测速方法都是基于摄像机标定，然后推测出车辆的行驶距离以及时间来得到车辆的行驶速度。但是在车载视频测速时，获取摄像头的标定会很方便。因此在事故鉴定时，常用的方法是根据图像中的参照物来推算出车辆行驶距离。根据GAT 1133-2014标准中所规定的，车载式视频图像的车辆行驶速度计算示意图：



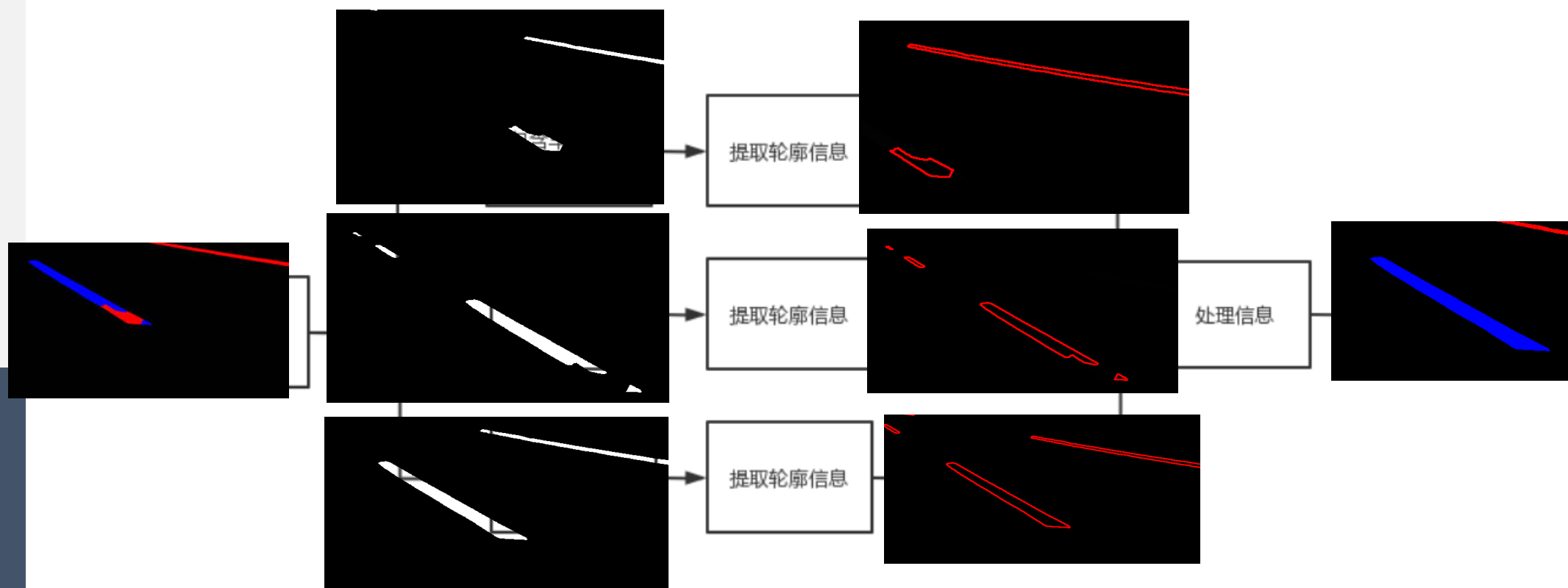
3.1 传统测速算法

目前车速鉴定中使用的测速方法具体示例如下。下图是一个以车行道分界线为参照物进行测速的例子。



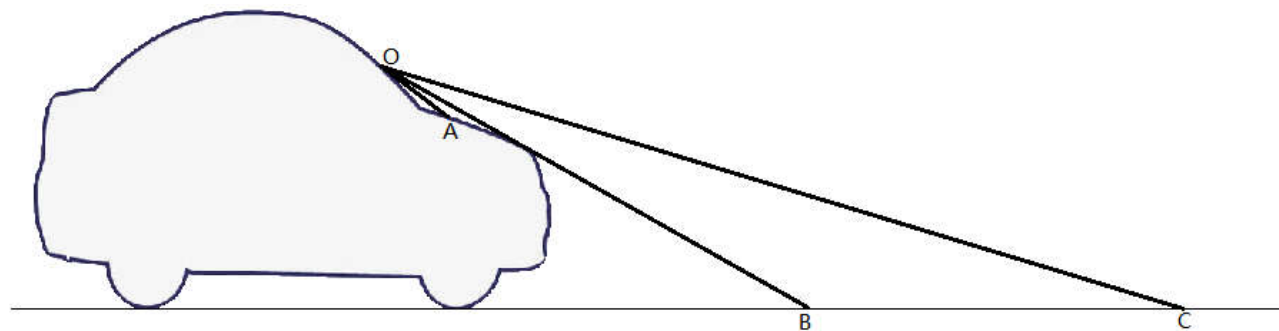
$$v = \frac{s}{t}$$

3.2 测速算法设计-优化分割结果



3.2 测速算法设计-消失线确定

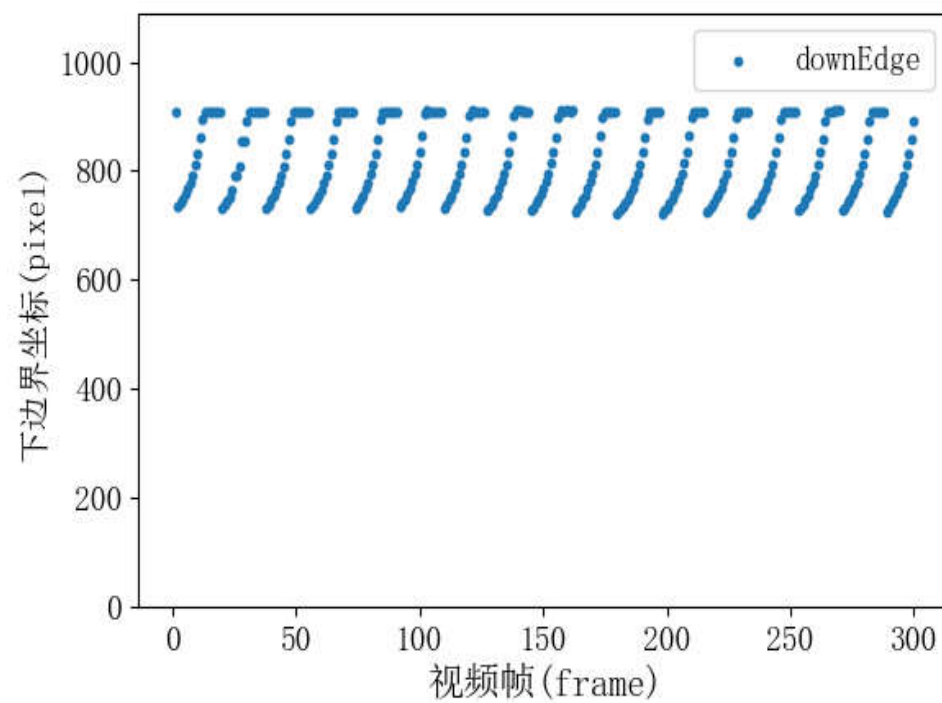
侧视图



正面图

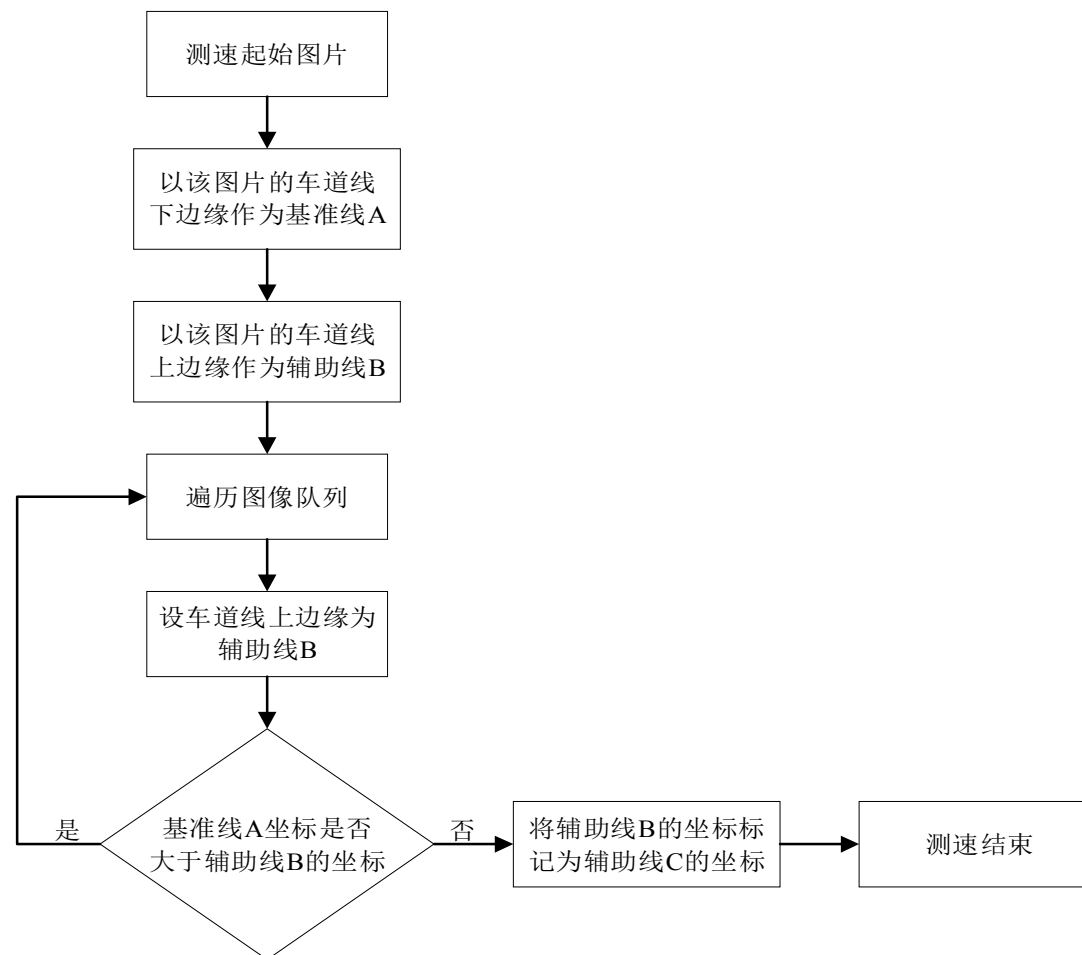


3.2 测速算法设计-消失线确定

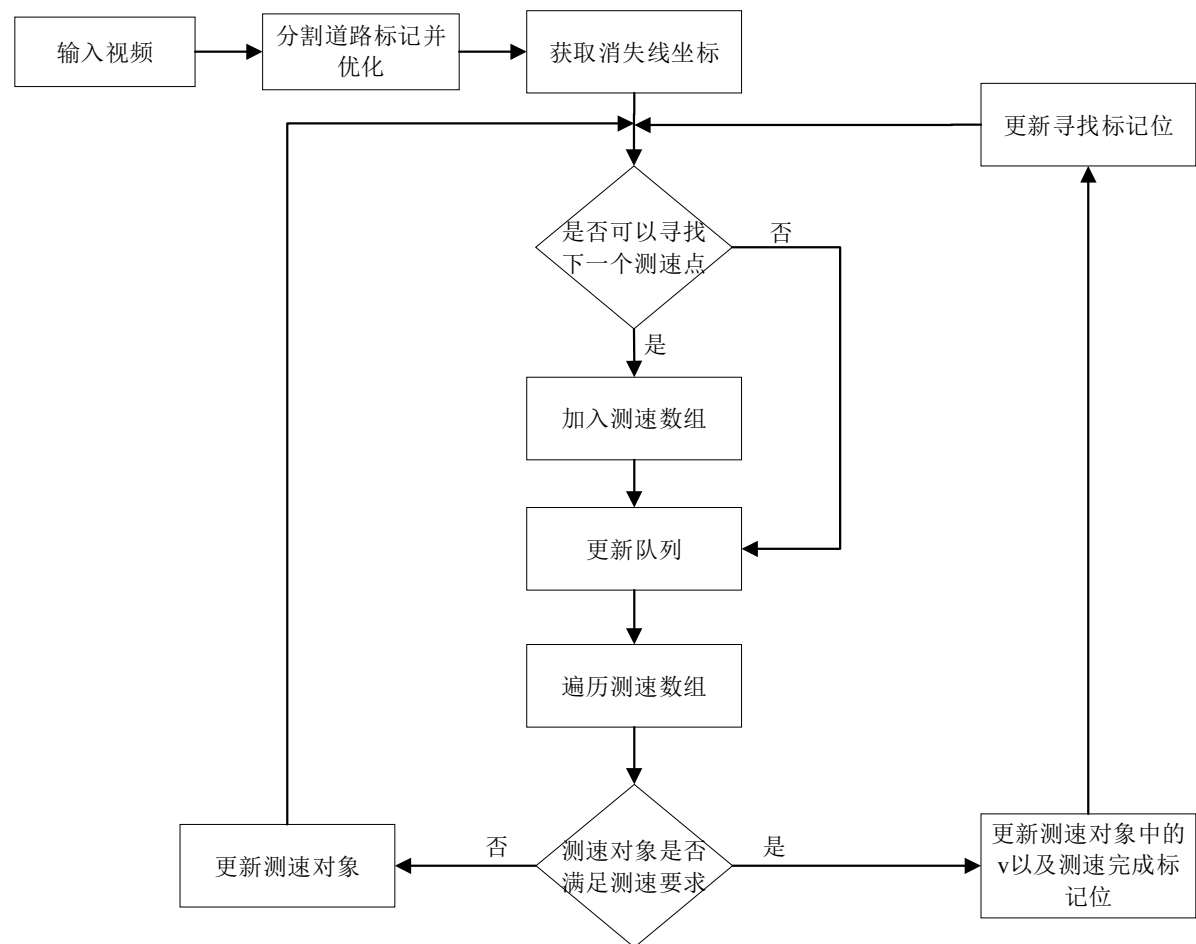


车行道分界线下边界坐标

3.2 测速算法设计-测速模块设计



3.2 测速算法设计-整体测速框架



3.3 实验评估-车速真值获取



3.3 实验评估-实验1

实验1采用的视频为制作数据集剩下的4个视频来进行测速实验。这四个视频的参数如下表所示。

视频序列	时长(min)	分辨率	帧率(fps)	测速点数(个)
Video_12	3	1920×1080	30	180
Video_13	3	1920×1080	30	180
Video_14	1	1920×1080	30	60
Video_15	1	1920×1080	30	60

3.3 实验评估-实验1



Video_12



Video_13

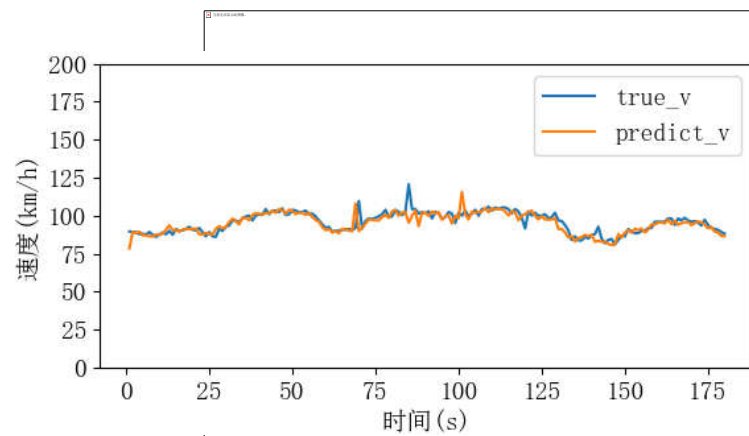


Video_14

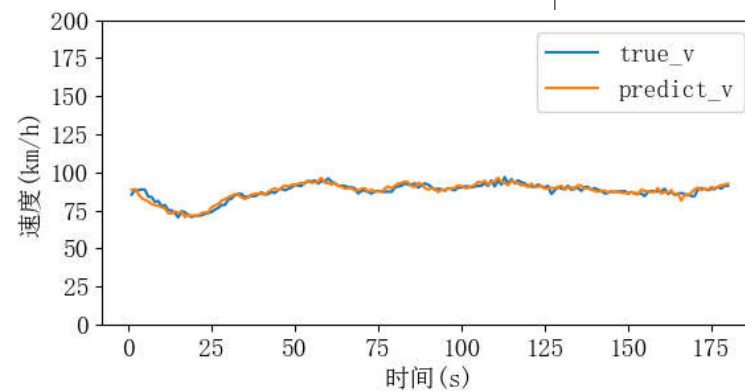


Video_15

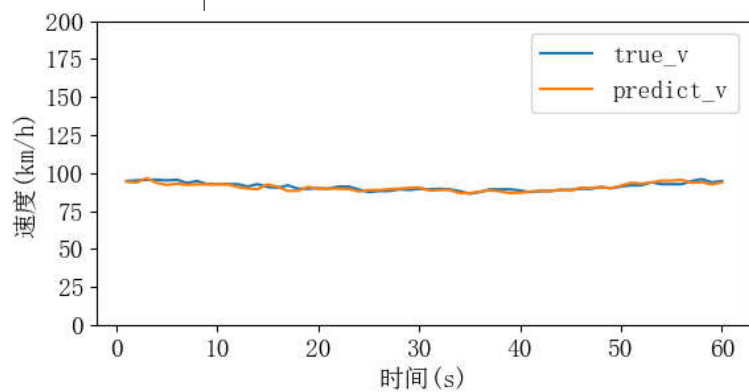
3.3 实验评估-实验1



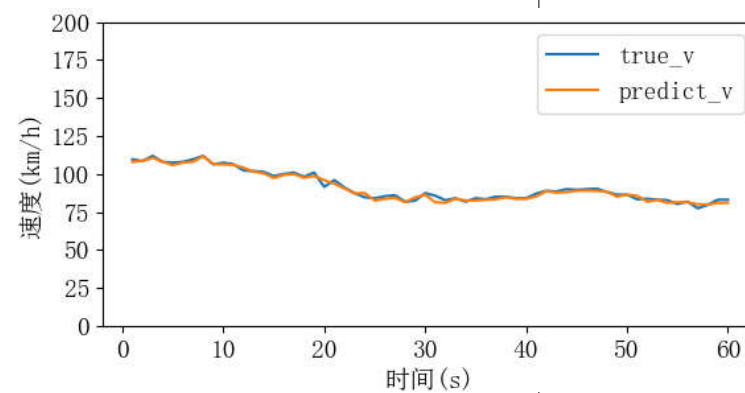
Video_12



Video_13



Video_14



Video_15

3.3 实验评估-实验1

对于最终测速结果评价，本文使用平均绝对误差(Mean Absolute Error, MAE)来进行衡量，公式如下所示。

$$MAE(v_p, v_t) = \frac{1}{n} \sum_{k=1}^n |v_p - v_t|$$

视频序列	Video_12	Video_13	Video_14	Video_15
MAE(km/h)	2.38	1.49	1.15	1.18

3.3 实验评估-实验2

实验2采用的视频为实际中发生交通事故的四个视频来进行测速实验。这四个视频的参数如下表所示。

视频序列	时长	分辨率	帧率
Video_16	2m49s	1920×1080	30
Video_17	2m	1920×1080	25
Video_18	1m	1920×1080	30
Video_19	1m13s	1280×720	25

3.3 实验评估-实验2



Video_16



Video_17

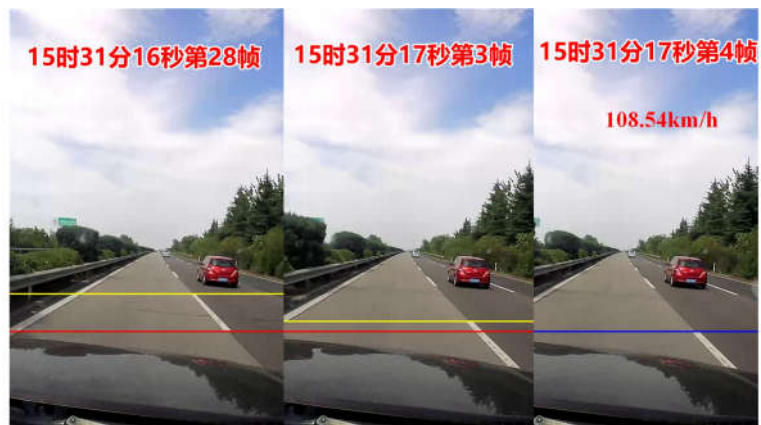


Video_18

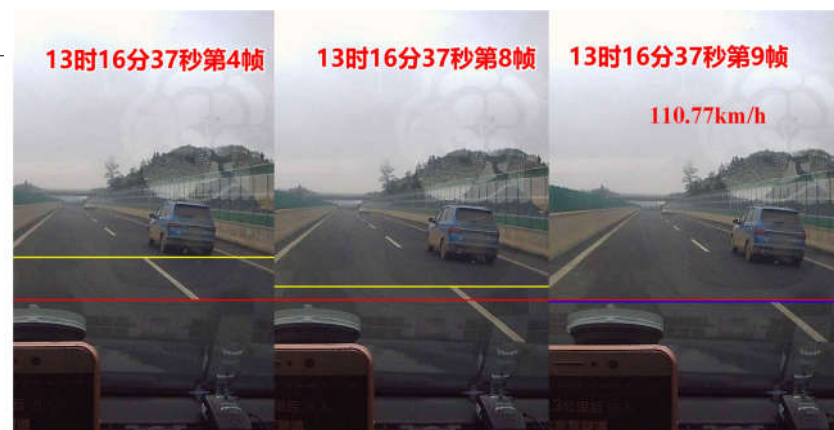


Video_19

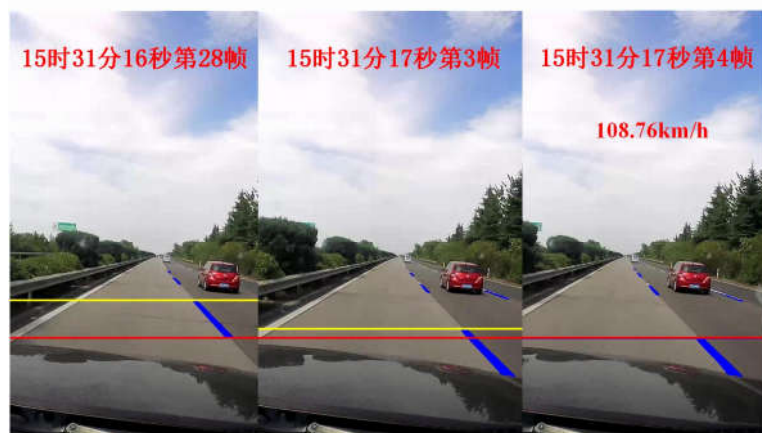
3.3 实验评估-实验2



手动



手动



本文方法

Video_16



本文方法

Video_17

3.3 实验评估-实验2



手动



本文方法

Video_18



手动



本文方法

Video_19

3.3 实验评估-实验2

视频名	帧率(fps)	手动		本文方法		误差值(km/h)
		时间(s)	速度(km/h)	时间(s)	速度(km/h)	
Video_16	30	0.1990	108.54	0.1986	108.76	0.22
Video_17	25	0.1949	110.77	0.1970	109.64	1.13
Video_18	30	0.2330	92.70	0.2333	92.57	0.13
Video_19	25	0.2539	85.04	0.2523	85.61	0.57

可以看出手动和本文方法的测速最大的误差为1.13km/h,基本相差不大。验证了本文算法的有效性。

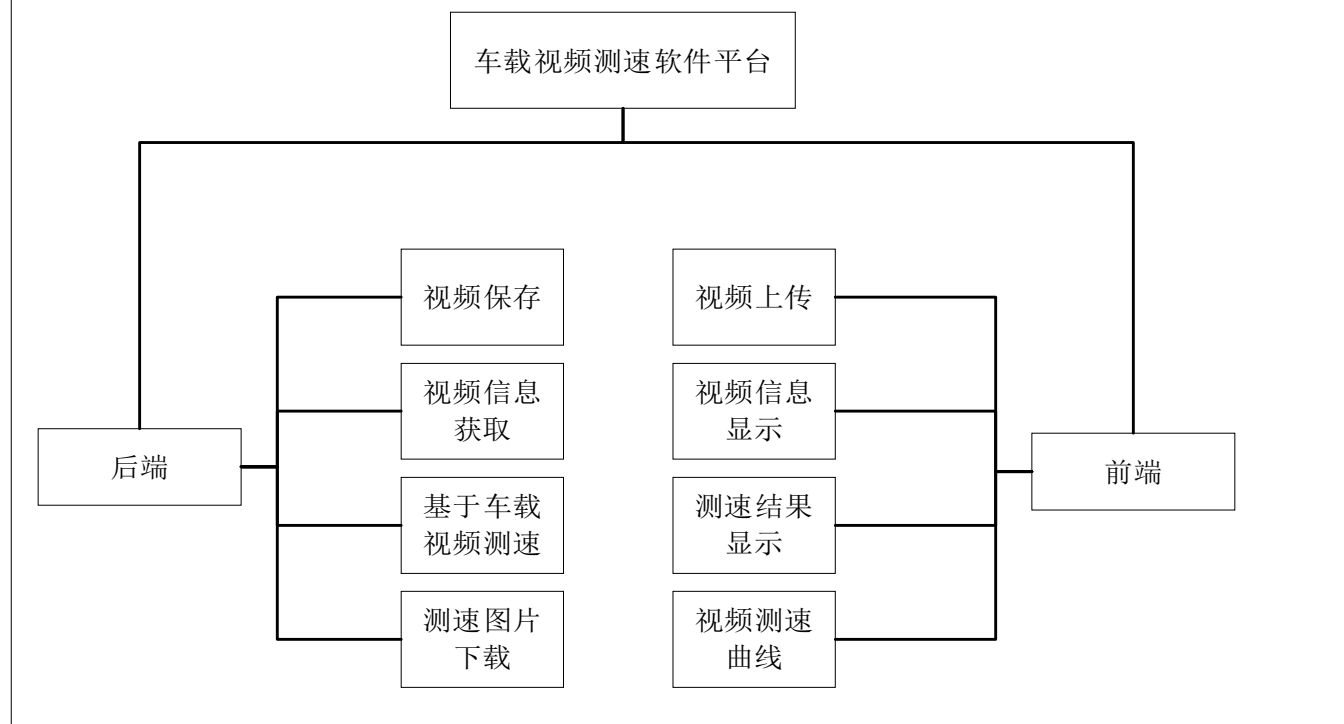


04

软件平台设计与实现

4.1 软件整体架构

本软件平台的开发过程在Linux系统下完成，主要分为前端界面设计和后端算法两个部分。前端界面部分使用HTML和JS语言进行开发，后端则使用Python语言结合Flask框架。开发平台为JetBrains PyCharm。



4.2 功能展示

视频上传：

基于车载视频测速平台

视频测速

视频上传

视频测速结果

视频测速曲线

文件信息

文件名: cut5.mp4

格式: mp4

分辨率: 1920X1080

帧率: 30fps

文件大小: 6.50MB

总时长: 00:00:05

比特率: 1089Kb/s

MD5码: 449e5aa8b...



测速结果：

基于车载视频测速平台

视频测速

视频上传

视频测速结果

视频测速曲线

文件信息

文件名: cut5.mp4

格式: mp4

分辨率: 1920X1080

帧率: 30fps

文件大小: 6.50MB

总时长: 00:00:05

比特率: 1089Kb/s

MD5码: 449e5aa8b...



4.2 功能展示

测速曲线：

基于车载视频测速平台

视频测速

视频上传

视频测速结果

视频测速曲线

文件信息

文件名: cut5.mp4

格式: mp4

分辨率: 1920X1080

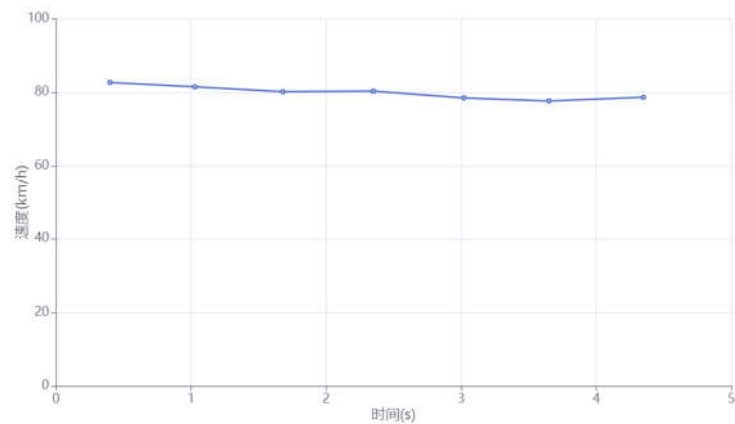
帧率: 30fps

文件大小: 6.50MB

总时长: 00:00:05

比特率: 10896kb/s

MD5码: 复制 446e5aaf8...



测速图片下载：



9.png



10.png



11.png



12.png



13.png



14.png



15.png



16.png



17.png



05

总结与展望

研究总结



研究总结

展望

- 1. 数据集方面：**后续可在更丰富的场景下，采集包含各种情况下的车载视频，进一步丰富道路标记数据集。
- 2. 道路标记分割方面：**一方面，可以优化网络结构，在保证分割精度的同时提高道路标记的分割速度。另一方面，可以采用视频语义分割算法。
- 3. 测速方面：**在后续研究的过程中，可以在本文研究工作的基础上，加入其他道路标记的测速，进一步完善自动车速测量。