

AlexNet模型的平均精度最高，达到85.37%。模型的分类型混沌矩阵如表3.6所示。

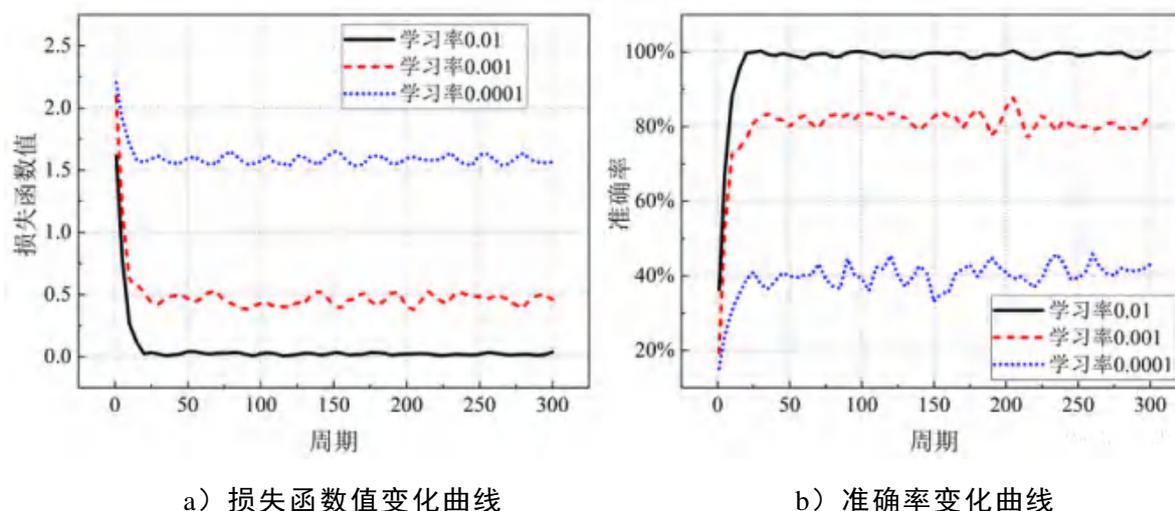


图 3.13 全训练 AlexNet 训练周期损失值和准确率变化

表 3.5 全训练 AlexNet 模型训练情况

序号	模型学习率	损失值变化情况	准确率变化情况	模型训练耗时	模型平均精度
1	0.0001	40个训练周期后 在1.6左右波动	40个训练周期后 在40%左右波动	2:19:00	25.81%
2	0.001	50个训练周期后 在0.5左右波动	40个训练周期后 在80%左右波动	2:11:19	72.29%
3	0.01	40个训练周期后 在0.03左右波动	40个训练周期后 在99%左右波动	2:20:41	85.37%

表 3.6 测试集的分类混沌矩阵

车辆类型	小轿车	面包车	中型客车	大型客车	公交车	2轴4轮货车	2轴6轮货车	3轴货车	4轴货车	精度
小轿车	207	0	0	0	0	0	0	0	0	100%
面包车	3	178	0	0	0	24	2	2	0	85.17%
中型客车	0	2	39	0	2	0	1	0	0	88.64%
大型客车	0	0	1	134	11	0	1	0	0	91.16%
公交车	0	0	1	4	142	0	0	0	0	96.60%
2轴4轮货车	0	6	0	0	0	178	28	0	0	93.96%
2轴6轮货车	0	1	0	1	0	7	296	74	21	74.00%
3轴货车	0	0	0	0	0	0	21	321	91	74.13%
4轴货车	0	0	0	0	0	0	0	89	269	75.14%

3.5.2.2 微调模型参数分析

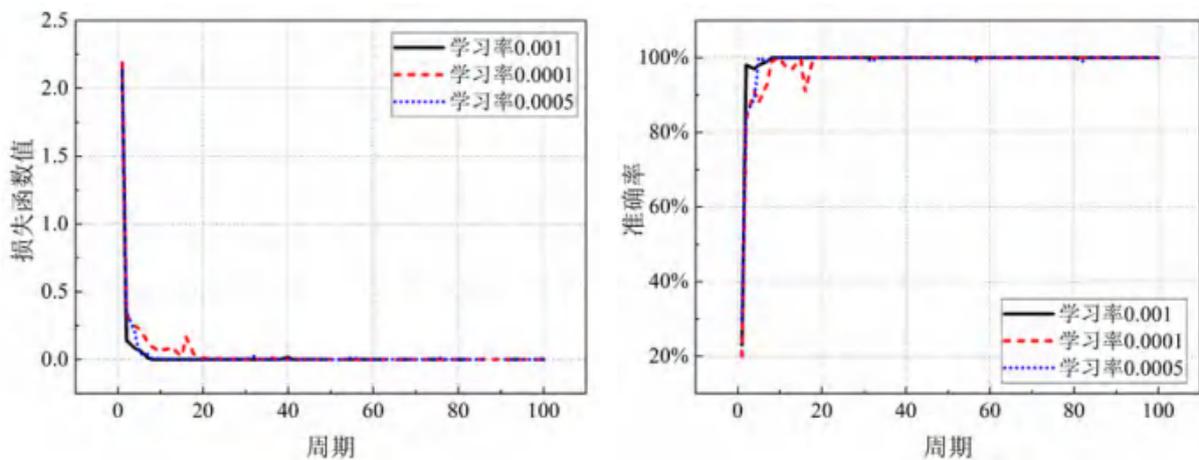
由于模型参数量达到6000万个，而训练数据仅有6000多张，这样的数据量远远不足以训练一个好的模型。迁移学习能够改善数据不足的情况，本文选用迁移学习中的微调（Fine-tuning）方法来训练模型。

微调是将在大型数据集上训练过的模型稍作修改后在自己的小数据集上训练，其实就是保留在大型数据集上训练过的模型的部分参数，其他参数重置来训练模型。训练的方式可以是微调所有层参数，或者固定网络前面几层参数，只微调网络的后面几层。这样一方面能够避免因数据量小造成过拟合现象，另一方面，卷积神经网络的前几层特征中多为浅层特征，如边缘、线条等，这些特征对于大部分任务都是共通的，后几层特征即结构特征等则根据数据集的不同而不同。这里使用的微调方式是固定前面几层参数不变，只重置后几层的参数。

首先加载在大型公开数据集ImageNet上训练过的AlexNet模型，然后对其进行微调。即保留前面所有卷积层的参数不变，只将最后一层全连接层的参数重置，叠加Softmax分类器并将输出类别数设置为车辆类别数9。

为了找到学习率的最优值，训练参数设置为具有动量的随机梯度下降算法，每次训练迭代的最小批次为100，最大训练周期为100，学习率分别为0.0005、0.0001、0.001和0.01。将数据存储对象、配置的训练参数以及预设的网络结构作为输入，分别对模型进行训练，训练过程中训练周期的损失函数值曲线及准确率曲线如图3.14所示，对应的训练情况如表3.7所示。

由图3.14和表3.7可知，对于微调AlexNet模型，当学习率设置为0.001时模型的平均精度最高。图3.14中未画出学习率为0.01时的变化曲线，是由于模型从第2个训练周期开始，周期损失函数值骤增，模型无法收敛。说明学习率设为0.01对于微调模型来说过大，参数更新幅度太快导致模型无法收敛。



a) 损失函数值变化曲线

b) 准确率变化曲线

图 3.14 微调 AlexNet 训练周期损失值和准确率变化

表 3.7 不同学习率对应的微调 AlexNet 模型训练情况

序号	模型学习率	损失值变化情况	准确率变化情况	模型训练耗时	模型平均精度
1	0.0005	60个训练周期后 在0.0002左右波动	60个训练周期后 达100%	0:32:25	83.82%
2	0.0001	70个训练周期后 在0.0009左右波动	70个训练周期后 达100%	0:32:46	82.82%
3	0.001	45个训练周期后 在0.00009左右波动	8个训练周期后达 100%	0:32:25	85.30%
4	0.01	骤增, 无法收敛	/	/	/

为了对比训练周期对模型精度的影响, 将训练参数设置为随机梯度下降算法, 每次训练迭代的最小批次为100, 学习率为0.001, 最大训练周期分别为50、100和150。三个模型训练周期对应的损失函数值和准确率区别不大, 训练周期为50的模型平均精度为84.81%, 训练周期为100的模型平均精度为85.30%, 训练周期为150的模型平均精度为84.57%。由上述结果可知, 训练周期设置为100时, 模型的精度较高。

最后, 为了对比最小批次数量对模型精度的影响, 将训练参数设置为具有动量的随机梯度下降算法, 学习率为0.001, 最大训练周期为100, 每次训练迭代的最小批次分别为32、64、100、128。四个模型的训练情况及其平均精度如表3.8所示。

结果表明, 当模型训练参数设置为具有动量的随机梯度下降算法, 学习率为0.001, 最大训练周期为100, 每次训练迭代的最小批次为32时, 微调的AlexNet模型性能最好, 平均精度为87.55%。

表 3.8 不同最小批次数量对应的微调 AlexNet 模型训练情况

序号	模型最小批次数量	损失值变化情况	准确率变化情况	模型训练耗时	模型平均精度
1	32	17个训练周期后 在0.00005左右波动	5个训练周期后达 100%	0:36:55	87.55%
2	64	25个训练周期后 在0.0001左右波动	10个训练周期后 达100%	0:33:50	85.25%
3	100	46个训练周期后 在0.0001左右波动	8个训练周期后达 100%	0:32:25	85.30%
4	128	50个训练周期后	12个训练周期后	0:40:46	84.91%

在0.0002左右波动 达100%

上述参数设置下训练的模型在测试集上车辆识别的混沌矩阵如表3.9所示，可见模型在各种类型上的精度有较大程度差别，对外观差别显著的小轿车、面包车、中型客车、大型客车、公交车几种类型的识别效果较好，精确度均在90%以上。而2轴6轮客车、3轴客车、4轴客车的准确度却很低，有近一半4轴货车被识别成3轴货车，30%的3轴货车识别成4轴货车，其原因可能是因为两者之间的区别仅仅在于轴数，在图片即训练集上表现为车轮数量的区别，数据集数量的不足使得模型没有学习到3轴货车与4轴货车之间的精确特征并进行区分。

表 3.9 测试集的分类混沌矩阵

车辆类型	小轿车	面包车	中型客车	大型客车	公交车	2轴4轮货车	2轴6轮货车	3轴货车	4轴货车	精度
小轿车	207	0	0	0	0	0	0	0	0	100%
面包车	0	203	0	0	0	6	0	0	0	97.3%
中型客车	0	1	42	1	0	0	0	0	0	95.45%
大型客车	0	0	2	139	5	0	0	1	0	94.56%
公交车	0	0	0	4	143	0	0	0	0	97.82%
2轴4轮货车	1	1	0	0	0	202	6	2	0	95.28%
2轴6轮货车	0	0	0	0	0	0	317	78	5	79.25%
3轴货车	0	0	0	0	0	1	14	297	121	68.59%
4轴货车	0	0	0	0	0	0	0	140	218	60.89%

3.5.2.3 数据增强训练方法

从训练结果来看，尽管训练中最小批次精度达到100%，损失达到0.0001，然其在测试集上的平均精度为86.52%，因此模型的泛化能力较差。为改善这个问题进一步提升模型性能，在不增加数据库样本量和图像处理工作量的基础上，使用数据增强的手段对训练数据进行处理。

数据增强是减轻网络过拟合，提高模型的泛化能力的有效手段。数据增强常用的方法有：旋转变换、翻转变换、裁剪变换、缩放变换、平移变换、对比度变换、尺度变换、颜色变换和噪声扰动等。

首先，将训练数据集所有图片做横向翻转变换，保存横向翻转后的所有图片，并按前文方法进行处理，得到的新的训练数据集数量翻倍，共有12934张车辆图。测试数据集不做变换，仍为2157张车辆图片。

模型的训练参数学习率为0.001，最大训练周期为100，最小训练批次为32。模型训练耗时01:15:39，在测试集上的平均精度为87.49%。与相同参数未进行翻转变

换的模型结果相差不大，因而在现有数据集上只增加一两倍的图像对结果影响不大。

对此，本文使用数据增强函数来实现数据增强，该函数每个训练周期对原始图像进行随机处理，得到的图像作为当前训练周期的输入，下一训练周期又在原数据库基础上产生新的输入图像。使得有限的训练数据集提供了更多信息，同时既不增加计算损耗又能有效减轻数据不足的影响。数据增强函数能够实现随机横向翻转、随机横向平移、随机纵向平移、随机旋转、随机横向缩放和随机纵向缩放的随机组合，以图3.15a)为例，各种变换如图3.15所示。



图 3.15 数据增强各种变换示例

为了测试数据增强函数各项参数设置的最佳取值范围，进行了一系列对比实验，模型训练参数均设置为：学习率为0.001，每次训练迭代的最小批次为32，最大训练周期为100，每个模型的数据增强函数参数的设置以及模型对应的平均精度如表3.10所示。

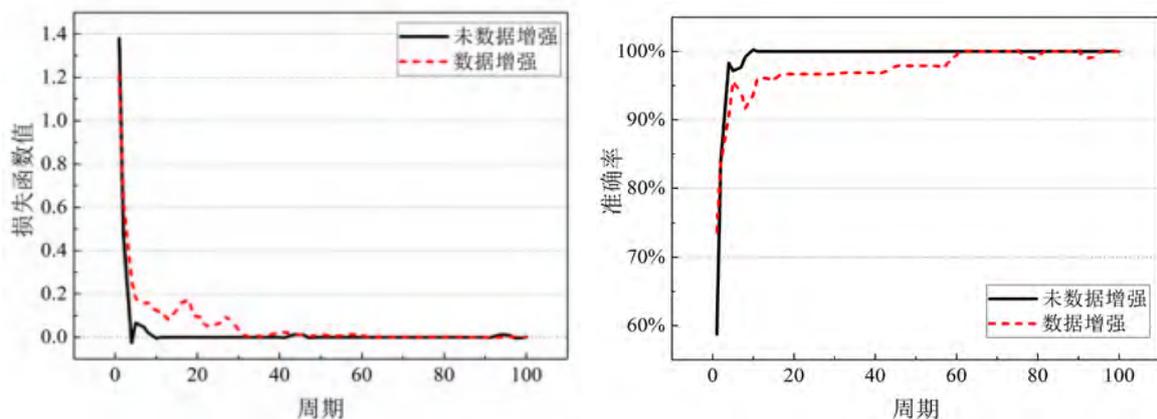
由表3.10可知，当数据增强设置为随机翻转、随机旋转范围 $[-20^{\circ}, 20^{\circ}]$ ，随机横向、纵向平移 $[-10, 10]$ ，随机横向、纵向缩放比例 $[0.9, 1.1]$ ，同时学习率为0.001，每次训练迭代的最小批次为32，最大训练周期为100时，模型精度较好，平均精度在92.59%到93.40%范围内。

其中平均精度达到93.4%的模型训练过程中的损失函数值、准确率变化如图3.16所示，其对应的模型在测试集上的分类混沌矩阵见表3.11。

由图3.16可知，未进行数据增强操作的模型损失值和准确率在训练前20个周期内迅速变化然后稳定，而进行数据增强的模型的损失值和准确率变化更缓慢然后趋于稳定，可见数据增强操作使得模型在训练集上得到了更充分的训练。

表 3.10 数据增强设置的实验结果

序号	随机横向翻转	随机旋转角度	随机横向平移	随机纵向平移	随机横向缩放	随机纵向缩放	平均精度
1	是	[-20°,20°]	[-10,10]	[-10,10]	[0.8,1.2]	[0.8,1.2]	92.55%
2	是	[-20°,20°]	[-10,10]	[-10,10]	[0.9,1.1]	[0.9,1.1]	93.20%
3	是	[-20°,20°]	[-5,5]	[-5,5]	[0.8,1.2]	[0.8,1.2]	91.76%
4	是	[-20°,20°]	[-5,5]	[-5,5]	[0.9,1.1]	[0.9,1.1]	92.14%
5	是	[-15°,15°]	[-10,10]	[-10,10]	[0.8,1.2]	[0.8,1.2]	91.05%
6	是	[-15°,15°]	[-10,10]	[-10,10]	[0.9,1.1]	[0.9,1.1]	91.55%
7	是	[-15°,15°]	[-5,5]	[-5,5]	[0.8,1.2]	[0.8,1.2]	90.83%
8	是	[-15°,15°]	[-5,5]	[-5,5]	[0.9,1.1]	[0.9,1.1]	91.22%



a) 损失函数数值变化曲线

b) 准确率变化曲线

图 3.16 微调、数据增强 AlexNet 周期损失值和准确率变化

表 3.11 分类混沌矩阵

车辆类型	小轿	面包	中型	大型	公交	2轴4	2轴6	3轴	4轴	精度
------	----	----	----	----	----	-----	-----	----	----	----

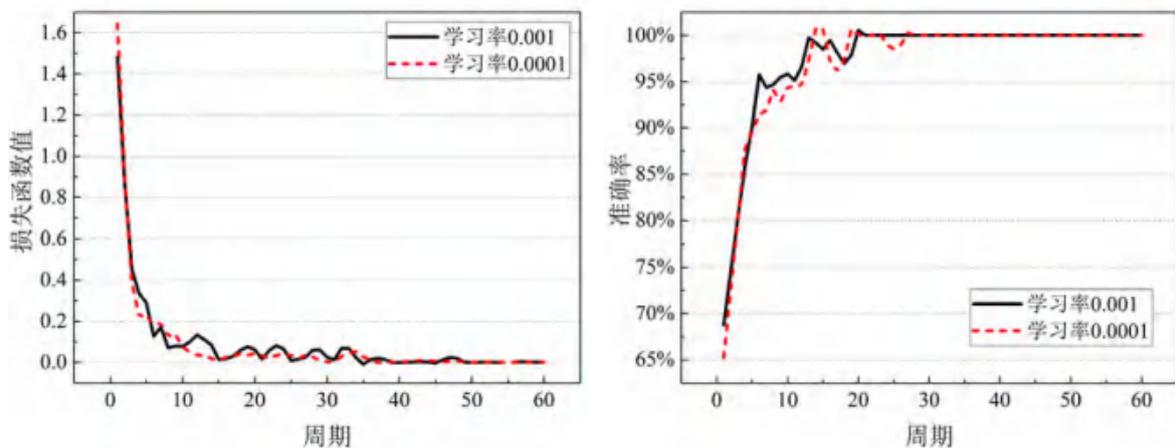
	车	车	客车	客车	车	轮货	轮货	货车	货车	
						车	车			
小轿车	204	3	0	0	0	0	0	0	0	98.6%
面包车	1	204	0	0	0	2	2	0	0	97.6%
中型客车	0	0	43	1	0	0	0	0	0	97.7%
大型客车	0	0	2	138	6	0	0	1	0	93.9%
公交车	0	0	0	3	144	0	0	0	0	98.0%
2轴4轮货车	0	3	0	0	0	199	10	0	0	93.9%
2轴6轮货车	0	0	0	0	0	2	384	12	2	96.0%
3轴货车	0	0	0	0	0	1	18	365	49	84.3%
4轴货车	0	0	0	0	0	0	0	69	289	80.7%

由表3.11可知，数据增强操作后的模型在测试集上各类型的分类精度都达到了80%，可见数据增强使得模型在小样本上训练更充分，有效提高了模型泛化能力和分类精度。

3.5.3 VGG-16 模型训练

对于更深的卷积神经网络VGG-16，由于其输入层为 $224 \times 224 \times 3$ 的彩色图片，首先将数据集所有图片转换成 224×224 大小，然后结合AlexNet的训练经验，对预训练的VGG-16进行微调，仍旧保留前面所有结构层参数不变，重置最后一层全连接层的参数，其后加入Softmax和分类层并将类别数目设置为9。

为了找到此情况下对应的最优学习率，将初始学习率分别设置为0.001、0.0001，最小批次为32，最大训练周期为60，数据增强配置为随机翻转，随机旋转范围 $[-20^\circ, 20^\circ]$ ，随机横向、纵向平移 $[-10, 10]$ ，随机横向、纵向缩放比例 $[0.9, 1.1]$ 。分别训练模型，训练过程中损失函数值和准确率变化曲线如图3.17所示。



a) 损失函数值变化曲线

b) 准确率变化曲线

图 3.17 微调、数据增强 VGG-16 损失值、准确率变化曲线

由图3.17可知，当学习率为0.001时，在45个训练周期后损失函数值在0.0008上下波动，准确率从第38个周期开始变为100%，模型训练耗时03:55:25，在测试集上的平均精度为98.17%；当学习率为0.0001时，在43个训练周期后损失函数值在0.001上下波动，准确率从第36个周期开始变为100%，模型训练耗时03:51:38，在测试集上的平均精度为96.36%。

其中学习率为0.001对应的模型在测试集上的分类混沌矩阵如表3.12所示，模型在测试集上各种类型的分类精度均达到96%以上，可见模型被充分训练能够准确地区分各类别车辆目标。

此外，对于同样参数配置的AlexNet和VGG-16两种模型，后者的训练时间是前者的七倍多，但VGG-16的精度较AlexNet有较大提升，这说明增加卷积网络结构层的深度能够有效提高模型的学习能力，提升精度。

表 3.12 分类混沌矩阵

车辆类型	小轿车	面包车	中型客车	大型客车	公交车	2轴4轮货车	2轴6轮货车	3轴货车	4轴货车	精度
小轿车	207	0	0	0	0	0	0	0	0	100%
面包车	0	209	0	0	0	0	0	0	0	100%
中型客车	0	0	44	0	0	0	0	0	0	100%
大型客车	0	0	2	142	3	0	0	0	0	96.6%
公交车	0	0	0	3	144	0	0	0	0	98.0%
2轴4轮货车	0	3	0	0	0	204	5	0	0	96.2%
2轴6轮货车	0	0	0	0	0	0	388	12	0	97.0%
3轴货车	0	0	0	0	0	0	11	417	5	96.3%
4轴货车	0	0	0	0	0	0	0	2	356	99.4%
类别总数	207	209	44	147	147	212	400	433	358	/

上述识别平均精度为98.17%的VGG-16模型的车型识别结果如图3.18所示，图片上的注释文字为该图片中的车辆类型以及车辆属于该类型的概率。





图 3.18 VGG-16 部分识别结果

3.6 本章小结

本章通过截取桥梁拍摄的视频帧图像和网络搜集车辆图片两种方式，建立了车辆图片数据集，包含小轿车、面包车、中型客车、大型客车、公交车、2轴4轮货车、2轴6轮货车、3轴货车、4轴货车共9种车辆类型。数据集共有8624车辆图片，其中75%为训练集，其余为测试集。

接着利用一系列方法来训练车辆识别模型。首先对AlexNet进行全训练，发现模型的平均精度为85.37%，由于数据集较小导致训练不足。然后对在大型公开数据集ImagNet训练好的AlexNet模型用微调的方法训练，微调的AlexNet模型的平均精度提高到87.55%。接着，用数据增强的方法进行训练，AlexNet模型的平均精度提高到93.4%。最后，利用微调模型和数据增强两种方法训练VGG-16模型，其平均精度达到了98.17%。研究发现，在相同训练参数下的VGG-16模型精度比AlexNet高，表明增加深度卷积层的数量能够提高网络的学习能力和精度。

第4章 视频图像中的车辆目标检测

4.1 引言

在本文的提出的车辆自动检测分类系统中，将输入的桥梁上拍摄的视频流转换为图像帧后，首先需要检测出车辆在图像中的位置。目标车辆的位置检测结果是车辆跟踪的基础，为后续跟踪器的初始化提供依据，同时也是车型识别模块中的输入信息。因此建立一个稳定的车辆目标检测模型对本文的研究十分重要。本章首先介绍了传统的目标检测方法和深度学习目标检测方法，然后用深度学习 Faster R-CNN 和基于高斯混合模型的背景差分法两种方法分别对图像中的车辆位置进行检测，结合本文的实际应用场景，发现深度学习 Faster R-CNN 更适用于复杂的桥梁上车辆检测问题。

4.2 目标检测方法

运动目标检测是一种图像分割技术，分离出背景中感兴趣的运动目标，是后续目标跟踪、目标分类的基础，其准确程度直接影响着后续的准确度和性能。在目前的研究中，基于计算机视觉的目标检测技术可以分为传统方法、机器学习方法和深度学习方法，其中传统方法指的是对视频帧中的运动目标进行检测的方法，要求视频中目标必须运动变化，而后两者则是对图像中的目标进行检测，目标在视频中可以是静止或者运动。

4.2.1 传统目标检测方法

传统的目标检测方法有边缘检测法^[100]、帧间差分法^[101]、背景差分法^[102]和光流法^[103]等，这些方法对视频帧中的运动目标进行检测，各种方法有不同的应用场景和特点，以下着重介绍帧间差分法和光流法的优缺点。

1. 帧间差分法^[101]

帧间差分法利用相邻两帧图像的相关性对运动物体进行检测，通过计算像素的时间差分在连续两帧图像中提取出运动目标所在区域。将连续两帧图像进行差分，然后将差分值与人工设定的阈值相比较，若超过阈值，则认为该像素点为前景，否则为背景。

三帧差分法是对两帧差分法的改进，将连续三帧图像中相邻两帧做差分处理，通过二值化去除噪声获得运动目标，从而获得更优的运动目标区域。

帧间差分法对外界环境和光线具有很好的适应性，计算简单，能够满足视频处理的实时性，但当目标移动速度缓慢或连续两帧图像差别不大时容易出现检测

错误和空洞现象，此外，其对噪声较敏感且抗噪能力较差。

2. 光流法^[103]

视频图像中每个像素点都在运动，区别在于运动前景灰度有明显变化而背景变化不大。根据灰度运动趋势将图像中每个像素点作为一个运动矢量而形成光流场，光流场反映图像像素取值的变化情况。光流法就是通过时间域变化和相关性大小确定固定间隔的两帧图像像素点间的对应关系，得到目标的运动轨迹^[104]。光流法的原理是将某时刻图像的像素点投影到三维物体上，分析每个点的矢量特征，若矢量变化连续则没有运动物体，反之，运动物体表现为矢量及其邻域矢量之间存在差异。

光流法不需预先了解场景信息即可检测出运动目标及其信息，因而不仅适用于静止背景下的运动目标检测，同时也适用于运动背景下的运动目标检测，具有良好的环境适应性。但该方法也存在计算复杂，运算时间长、错误率较高、抗噪能力较差、孔径问题等缺点。

4.2.2 深度学习方法

基于深度学习的目标检测算法领域大致可分为两类，基于区域建议的两步式目标检测算法，如 R-CNN^[56]、SPP-Net^[57]、Fast R-CNN^[58]、Faster R-CNN^[47]、R-FCN^[105]等，和基于回归方法的一步式方法，这类方法不需要提取候选区域，代表算法是 YOLO^[59]和 SSD^[60]。

R-CNN 目标检测算法使用选择性搜索算法(Selective Search)生成候选区域，对区域进行归一化处理，得到固定大小的区域输入预训练的 CNN 网络从而生成相同维度的特征向量，结合多个二分类器 SVM 生成每个候选区域的评分，最后借助边界回归对候选框进行修正以生成最终检测结果。该方法生成的候选区域包含大量重复计算，且其固定的尺寸输入会导致目标有一定形变或信息损失。

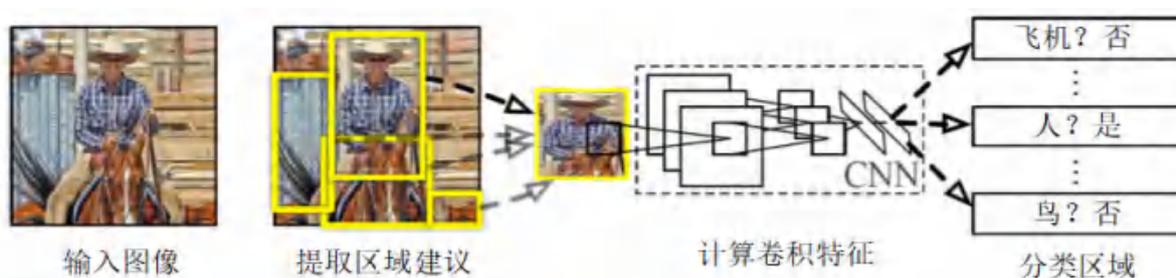


图 4.1 R-CNN 流程^[56]

Fast R-CNN 是对 R-CNN 算法的改进，使用基于金字塔原理的感兴趣区域池化层 (RoI 层) 来统一不同候选区域特征向量的尺寸，通过多任务学习将训练过程变成单阶段从而省去大量读写操作，通过映射关系计算不同候选区域卷积特征，由于整个原始图像直接送入卷积部分同时使用 softmax 代替 SVM 分类，避免了大

量重复计算，减少计算内存同时加快了训练速度。

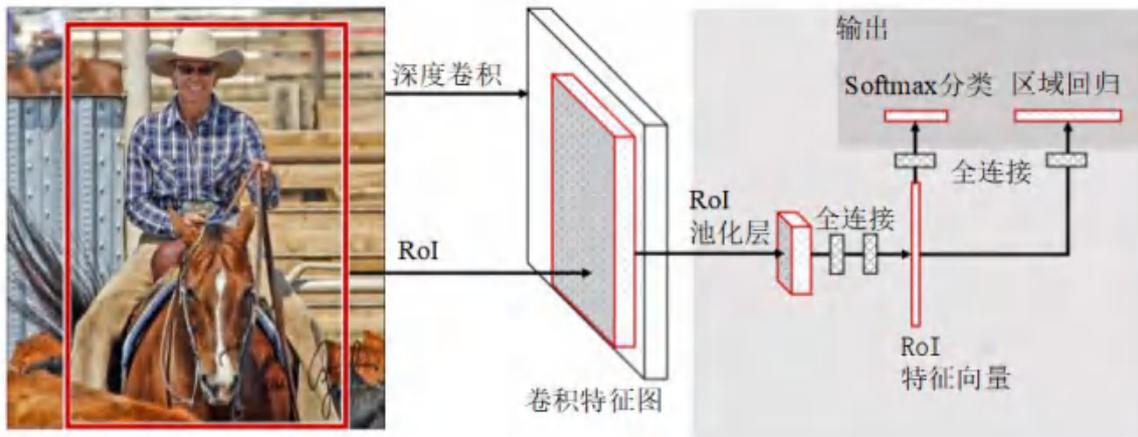


图 4.2 Fast R-CNN 流程^[58]

R-CNN 和 Fast R-CNN 算法都使用基于低级特征的选择性搜索或边缘检测算法生成区域建议，速度慢且候选区域质量较差，Faster R-CNN 对此提出了基于深度特征的区域建议网络（Region Proposal Networks, RPN），使用卷积神经网络来生成候选区域，通过卷积核在最后一层卷积特征上滑动以此回归出不同尺度和比例的候选区域，RPN 和分类器共享底层的卷积网络使得候选区域提取、卷积网络分类和边框回归合并到一起，极大提高了检测速度和精度。

YOLO 和 SSD 都是基于回归的目标检测算法，不需要区域建议，通过设置默认窗口完成在原始图像上的窗口截取，从深度特征中直接回归出目标的类别及位置偏移量来进行目标检测。主要流程是将输入图像归一化后分成若干个网格，待检测目标的中点会落入其中一个网格，一次性算出所有网格上可能对应的目标边框以及对应的类别概率。基于卷积特征直接产生位置和类别预测，通过非极大值抑制算法进行多余框的合并以生成检测结果。YOLO 使用全连接层来预测结果而 SSD 使用局部特征预测结果，由于取消区域建议，基于回归的算法实时性远超过基于区域建议的算法，然而这类算法由于网络结构的特性，其精度还远低于前者。

4.3 基于高斯混合模型的背景差分法

传统目标检测算法中，帧间差分法计算简单，但当目标移动速度缓慢或连续两帧图像差别不大时容易出现检测错误，同时其抗噪能力较差。光流法计算复杂，运算时间长、抗噪能力较差。背景差分法是目前主流的运动目标检测方法。

背景差分法^[102]是目前最常用的运动目标检测方法，通过对一张或多张图像的分析处理获得背景模型，将视频逐帧与背景模型的差分结果与阈值对比，以判断像素点是否为前景。其原理图如图 4.3 所示，其中 $f_k(x,y)$ 和 $b_k(x,y)$ 分别表示第 K 帧输入图像和背景模型，差分图像 $D_k(x,y)$ 的运算公式为：

$$D_k(x,y) = |f_k(x,y) - b_k(x,y)| \quad (4.1)$$

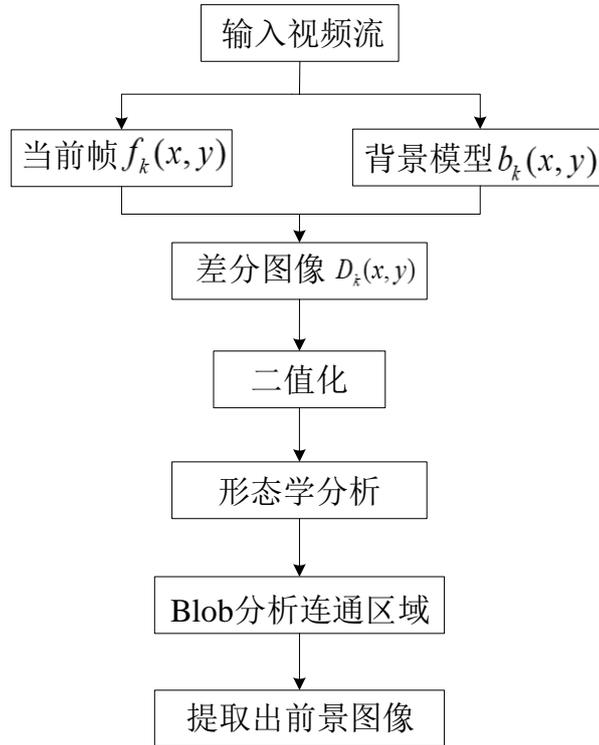


图 4.3 背景差分法基本原理

二值化后运动区域为：

$$R_k(x,y) = \begin{cases} 1 & \text{foreground} & D_k(x,y) > T \\ 0 & \text{background} & D_k(x,y) \leq T \end{cases} \quad (4.2)$$

背景差分法实现简单、速度快，能够满足视频处理的实时性要求，其核心在于如何建立一个优秀的背景模型。在实际情况中，若有摇摆的树枝、阴影、光线变化等将导致检测结果不准确。

4.3.1 高斯混合模型背景建模

背景差分法的关键在于背景模型的建立，其理想的背景是除了运动目标其他对象都是静止的，然而在实际情况中很难达到，常常导致检测结果不准确。由此引入了背景模型的更新机制，常用的方法有：统计平均法^[106]、核密度估计^[107]、单高斯模型^[108]、高斯混合模型^[109]等方法。

背景模型更新方法中最常用的是高斯混合模型法，高斯混合模型法一般用 3-5 个高斯函数表示复杂多变的场景，对环境变化、遮挡等问题有较好的处理效果。图像中像素点对应的概率密度函数为：

$$P(X_{j,t}) = \sum_{i=1}^K \omega_{j,t}^i \eta(X_{j,t}, \mu_{j,t}^i, \Sigma_{j,t}^i) \quad (4.3)$$

K 为高斯函数的个数，常取3-5， $X_{j,t}$ 为 t 时刻位置 j 的像素值， $\omega_{j,t}^i$ 、 $\mu_{j,t}^i$ 、 $\Sigma_{j,t}^i$

分别表示 t 时刻第 i 个模型位置 j 的权重、均值和方差, $\Sigma_{j,t}^i = (\sigma_{j,t}^i)^2 I$ (I 为单位矩阵), η 为高斯分布概率密度函数, 公式如下:

$$\eta(X_{j,t}, \mu_{j,t}^i, \Sigma_{j,t}^i) = \frac{1}{(2\pi)^{n/2} |\Sigma_{j,t}^i|^{1/2}} e^{-\frac{1}{2} (X_{j,t} - \mu_{j,t}^i)^T (\Sigma_{j,t}^i)^{-1} (X_{j,t} - \mu_{j,t}^i)} \quad (4.4)$$

高斯混合模型的建模步骤大致如下:

1) 初始化: 将均值设为图像任一像素点的值, ω 设为 $1/K$, 方差设一个较大的值。

2) 模型匹配: t 时刻的像素值 X_t 与高斯混合模型中第 i 个高斯分布的均值距离满足下式, 判定此像素和高斯混合模型匹配。

$$|X_t - \mu_{j,t}^i| < 2.5\sigma_{j,t}^i \quad (4.5)$$

3) 模型更新: 若匹配, 对模型权重、均值和方差更新;

$$\omega_{j,t+1}^i = (1-\alpha)\omega_{j,t}^i + \alpha M_{j,t}^i \quad (4.6)$$

若 X_t 不能与任何一个背景模型匹配, 则去除当前高斯分布中权值最小的一个, 引入一个新的高斯分布:

$$\rho = \alpha / \omega_{j,t}^i \quad (4.7)$$

$$\mu_{j,t+1}^i = (1-\rho)\mu_{j,t}^i + \rho X_{j,t} \quad (4.8)$$

$$(\sigma_{j,t+1}^i)^2 = (1-\rho)(\sigma_{j,t}^i)^2 + \rho(X_{j,t}^i - \mu_{j,t}^i)^T (X_{j,t}^i - \mu_{j,t}^i) \quad (4.9)$$

其中, α 和 ρ 分别为权值和均值的更新率, 若高斯分布 G_i 与 t 时刻像素 X_t 匹配, 则 $M_{j,t}^i = 1$, 否则 $M_{j,t}^i = 0$ 。

4) 背景提取: 提取描述背景的像素, 将 $\omega_{j,t} / \sigma_{j,t}$ 降序排列, 取前 B 个高斯分布作为背景模型,

$$B = \arg \min_b (\sum_{i=1}^b \omega_{j,t} > T) \quad (4.10)$$

其中, T 为预设背景阈值 ($0.5 < T < 1$)。当 t 时刻像素值 X_t 与前 B 个高斯分布中某个分布相匹配时, 判定此像素为背景, 否则为前景。将高斯混合模型背景提取出来后, 经背景差分法将运动前景提取出来。

高斯混合模型能够处理室内外复杂环境中的运动目标检测问题, 较好地模拟背景, 对背景描述的训练不需要大量样本且其计算时间较短。

4.3.2 数字形态学方法和 Blob 分析

数字形态学方法是图像处理中应用最为广泛的技术之一, 能够在保留信息的前提下抑制噪声。本章用在背景差分图像二值化之后, 对图像进行滤波处理, 从而达到更好的效果。

数字形态学的基本运算包括腐蚀和膨胀，其基本思想是用结构元素去提取图像中的对应形状从而达到图像识别和分析的目的，结构元素相当于信号处理中的滤波窗口。

对于二值化后的灰度图像，设 $f(x,y)$ 为输入图像， $b(x,y)$ 为对应的结构元素，用结构元素 $b(x,y)$ 对输入图像 $f(x,y)$ 进行膨胀运算表示为：

$$(f \oplus b)(s,t) = \max\{f(s-x,t-y) + b(x,y)\} \quad (4.11)$$

其中 $(f \oplus b)$ 表示结构元素 b 对输入图像 f 进行膨胀运算，可见膨胀运算可理解为将某一物体周围像素点与该物体合并，使物体边缘往外移动^[110]。

对应的腐蚀运算表示为：

$$(f \otimes b)(s,t) = \min\{f(s+x,t+y) + b(x,y)\} \quad (4.12)$$

其中 $(f \otimes b)$ 表示结构元素 b 对输入图像 f 进行腐蚀运算，可见腐蚀运算可理解为将某一物体边界点消除，使边界往内收缩^[110]。

Blob是指图像中具有相似特征（如颜色、纹理等）的连通区域，Blob分析是对这一块连通区域进行几何分析得到其特征，如区域面积、中心点坐标、最小外接矩形等。Blob分析首先将图形的前景和背景进行分割，然后根据目标的连通性对目标区域进行标记以将目标图像聚合为目标像素的集合体，最后通过计算获得连通区域的面积、中心点坐标、最小外接矩形等数据。

4.4 基于 Faster R-CNN 的车辆检测方法

Faster R-CNN 目标检测算法由提出区域建议的深度全卷积网络(RPN)和 Fast R-CNN 两部分组成，将提取候选区域、深度特征的提取、目标识别和位置信息四个步骤整合成一个整体，实现了卷积层的参数共享，大大降低了计算量的同时提高了运算速度。算法整体架构如图 4.4 所示，输入图像经过共享卷积层以获取共享特征图输入候选区域建议网络，经特定尺度的滑动窗口卷积得到特征向量，其中每个滑动窗口含有九个不同大小的候选框，判断每个候选框是否属于前景目标，RoI 池化层对取得的候选区域特征进行下采样并将新的特征输入分类器进行分类、回归和非极大值抑制处理从而得到目标位置。在这一架构中，候选区域建议网络和 Fast R-CNN 共享卷积层，用同一个卷积特征图进行计算使得两个模块在训练中同时运行，这部分卷积层常常采用 ZF、VGG-16 等卷积神经网络。本节中，使用上一章中使用过的 VGG-16 来提取特征。

4.4.1 候选区域建议网络

传统的区域建议网络有边缘盒、选择搜索算法等，这些算法产生的候选区域存在大量重叠导致大量冗余运算。基于共享卷积网络的区域建议网络(RPN)的实现是在最后一层共享卷积层的特征图上用一个 $n \times n$ 的滑动窗口滑动扫描，每个

窗口都产生一个特征向量，利用 anchor 机制、边界回归等生成区域建议^[47]。候选区域建议网络实际上是一个全卷积神经网络，结构如图 4.5 所示。

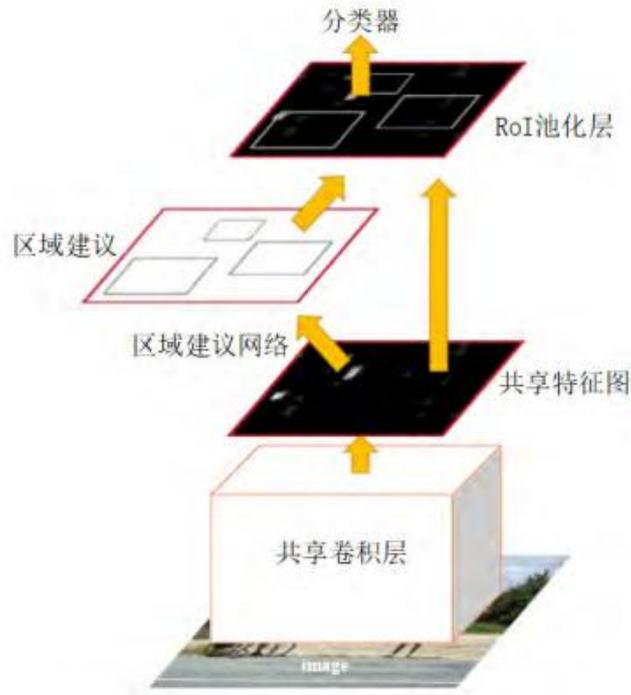


图 4.4 Faster R-CNN 结构^[47]

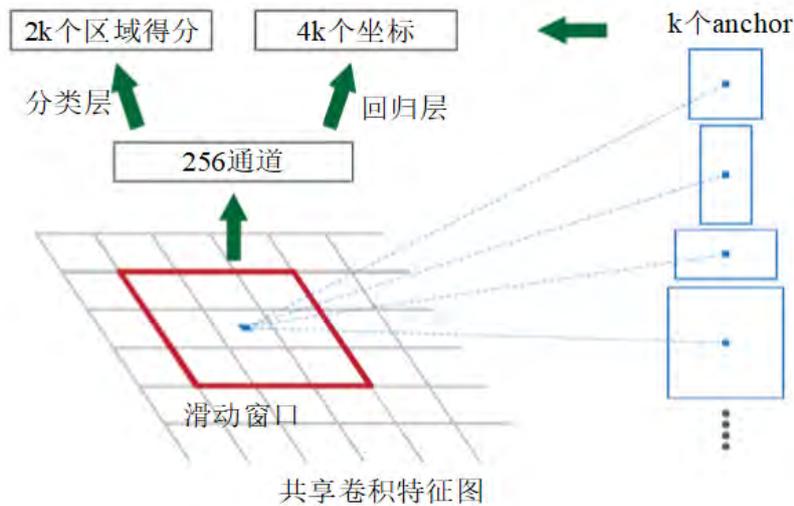


图 4.5 候选区域建议网络结构^[47]

在每个滑动窗口位置同时产生最多 k 个区域建议，对应于 k 个参考框，称为 Anchors。每个 Anchor 经过回归层输出 (x,y,w,h) 表示候选区域坐标，同时分类层有前景和背景之分，因而每个滑动窗口输出 $4k$ 个坐标和 $2k$ 个区域得分。

每个 Anchor 由滑动窗口中心点及纵横比共同决定，每个滑动窗口对应的候选区域为三种面积 $\{128^2, 256^2, 512^2\}$ 和三种纵横比 $\{1:1, 1:2, 2:1\}$ 组合产生的 9 个，即 $k=9$ 。对于大小为 $W \times H$ 的特征图对应 $W \times H \times k$ 个 Anchors，窗口每滑动一次对应的 9 个 Anchors 移动一次，同时对应原图像上的 16 个像素点移动一次，使得网

络具有平移不变性即平移图片中的目标图像仍能正确提取区域建议。

训练候选区域建议网络时，将 Anchors 与目标真实位置的标注之间的交并比作为衡量标准来判定区域建议是否准确包含目标，共有两种判定方式，一种为与真实标注区域重合度最高的为正确预测区域，另一种是当交并比大于 0.7 时为前景，否则为背景，通常用第二种方式足够检测出目标。交并比函数定义：

$$IOU = \frac{S_a \cap S_{gt}}{S_a \sqcup S_{gt}} \quad (4.13)$$

其中 S_a 表示 Anchors 区域， S_{gt} 表示目标真实标注的区域。

候选区域建议网络的训练目标是使多任务的损失函数最小，图像的损失函数表达式为：

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (4.14)$$

其中， i 为训练集中最小批次所包含的候选区域， p_i 是对应候选区域 i 属于检测目标的概率，当候选区域 i 为前景时， p_i^* 的值为 1，否则为 0。 $t_i = \{t_x^i, t_y^i, t_w^i, t_h^i\}$ 表示预测区域 i 的位置坐标向量， t_i^* 是目标真实区域的位置坐标向量。 L_{cls} 表示分类任务的损失函数， L_{reg} 表示候选区域边框回归任务的损失函数。 λ 用于平衡 L_{cls} 和 L_{reg} 之间的差距，通常取 10。

若使用均方误差法计算损失函数，可能会产生多个局部最优解导致梯度下降无法找到最优解，故使用逻辑回归法计算损失函数，表达式为：

$$L(y^*, y) = -(y \log y^* + (1-y) \log(1-y^*)) \quad (4.15)$$

其中， y 代表标记值， y^* 代表分类器的输出值。在候选区域建议网络中，对应为：

$$L_{cls}(p_i, p_i^*) = -(p_i^* \log p_i + (1-p_i^*) \log(1-p_i)) \quad (4.16)$$

只考虑 $p_i^* = 1$ 即候选区域为前景的情况，表达式变为：

$$L_{cls}(p_i, 1) = -\log p_i \quad (4.17)$$

此时， p_i 为前景的概率越接近 1，分类任务的损失函数越小。

使用 Smooth L1 函数来定义边框回归的损失函数：

$$L_{reg}(t_i, t_i^*) = \text{smooth}_{L1}(t_i - t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 & \text{if } |t_i - t_i^*| < 1 \\ |t_i - t_i^*| - 0.5 & \text{otherwise} \end{cases} \quad (4.18)$$

当预测区域与候选区域的相对位置和真实区域与候选区域的相对位置相近时，候选区域边框回归有效。

对于 t_i 用向量 $(t_x^i, t_y^i, t_w^i, t_h^i)$ 表示， t_i^* 用向量 $(t_x^{i*}, t_y^{i*}, t_w^{i*}, t_h^{i*})$ 表示，有：

$$\begin{aligned}
 t_x &= (x - x_a) / w_a & t_y &= (y - y_a) / h_a \\
 t_w &= \log(w / w_a) & t_h &= \log(h / h_a) \\
 t_x^* &= (x^* - x_a) / w_a & t_y^* &= (y^* - y_a) / h_a \\
 t_w^* &= \log(w^* / w_a) & t_h^* &= \log(h^* / h_a)
 \end{aligned}
 \tag{4.19}$$

其中, (x,y,w,h) 表示中心坐标及边框的宽和高, x,x_a,x^* 分别表示预测框、anchor框和真实标注框的中心横坐标, y,y_a,y^* 分别表示预测框、anchor框和真实标注框的中心纵坐标, w,h 同此理。

4.4.2 非极大值抑制算法

经过上述步骤后一个检测目标可能对应多个目标候选区域, 不全部代表正确区域, 使用非极大值抑制算法从这些区域中选取最优区域, 即与目标真实区域重合度最高的区域, 如图4.6所示。假设有6个候选区域, 根据检测为目标概率从小到大排序为A、B、C、D、E、F。从最大概率候选框F开始, 分别判断A-E与F的交并比是否大于设定的阈值(常设为0.7), 假设B、D与F的交并比超过阈值, 则删除B、D, 保留F, 从剩下的A、C、E中, 选择概率值最大的E, 判断E与A、C的交并比大于阈值则删除, 标记第二个保留下来的候选框。一直重复, 直至找到所有目标候选框。

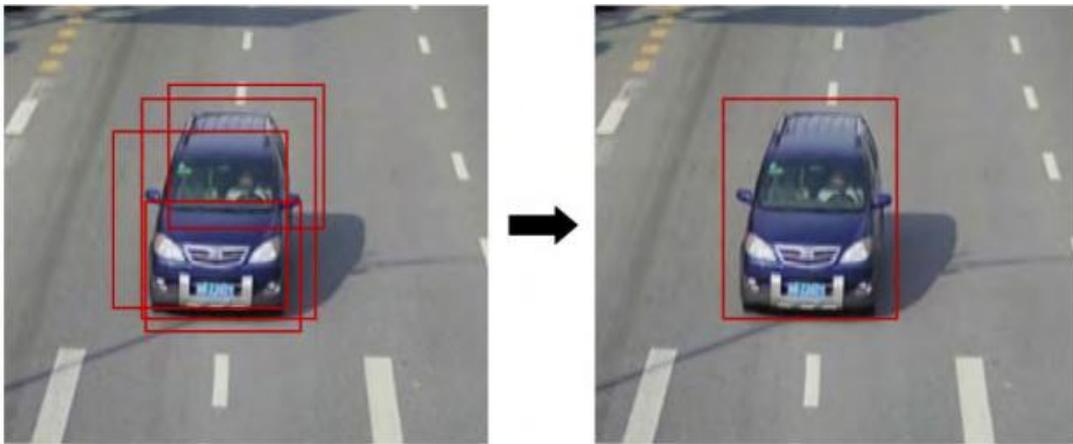


图 4.6 非极大值抑制

4.5 实验验证

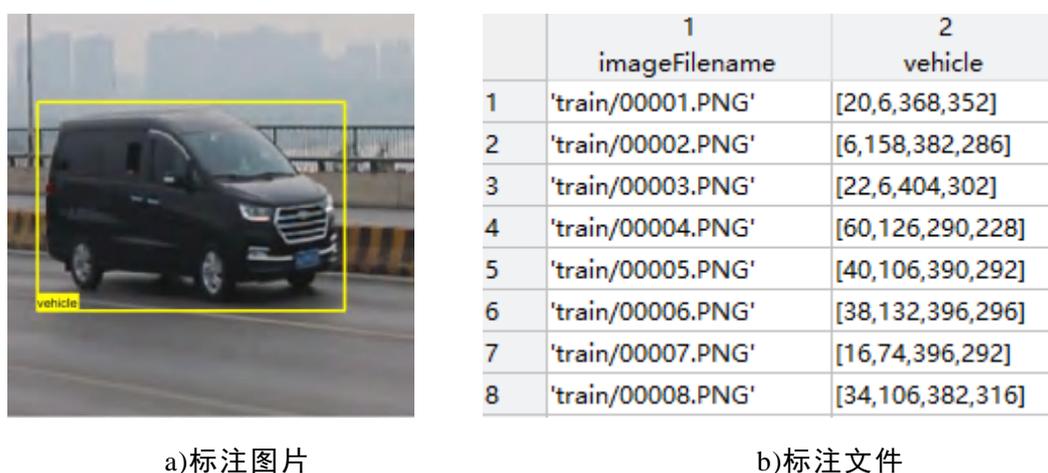
在桥梁上拍摄的视频中检测出车辆的位置是实现本文非接触式识别车辆的第一步, 本节拟使用基于高斯混合模型背景差分法和深度学习的 Faster R-CNN 两种方法进行车辆检测, 然后对比选出适合于复杂的实际场景下的车辆检测方法。

实验所用计算机配置为: 处理器 Intel(R) Core(TM) i7-8700 CPU @3.20GHz, 系统内存 16GB, 显卡 GTX 1060, 64 位操作系统 Windows10。实验软件为 Matlab R2018a。Faster R-CNN 模型的训练在 GPU 上进行。

4.5.1 Faster R-CNN 模型训练及检测结果

要使用Faster R-CNN方法检测出视频中车辆的位置，首先需要用大量的已标注车辆位置的图片对Faster R-CNN模型进行训练，然后将视频流转换成视频帧图像，将视频帧图像输入训练好的Faster R-CNN模型，即可输出每张图像中车辆所在位置，从而检测出视频中的车辆。

首先进行车辆数据集的标注工作，本章仍使用第三章中建立的车辆图片数据集作为训练数据集，采用标注工具Image Labeler标注出每张图片中车辆的位置坐标。标注的步骤为：将图片数据集导入标注工具，在每张图片中依次用矩形框画出车辆所在位置，矩形框应包含完整车辆同时尽可能少地包含背景，全部图片标注完成后输出标注数据文件。标注示例如图4.7所示，标注文件第一列为图片路径及图片名称，第二列为 1×4 矩阵，其中前两个元素为标注框左上角点坐标，后两个元素为标注框的宽和高。



a)标注图片

b)标注文件

图 4.7 数据集标注

然后对Faster R-CNN模型进行训练，模型训练的过程为：首先设置Faster R-CNN模型的结构以及各层结构参数的初始值，将车辆图片及其标注文件作为训练的输入，然后设置模型的训练方式和训练参数开始训练。

Faster R-CNN模型结构中用于学习和提取特征的共享卷积层通常可以选择AlexNet、ZF、VGG-16、ResNet等卷积神经网络，本文的共享卷积网络采用的是VGG-16网络，同时为了在有限数据集上获得更优的效果，本部分仍使用预训练过的卷积神经网络VGG-16来提取图片特征。候选区域建议网络的anchors个数设置为9个，候选区域与目标真实位置之间的交并比设置为0.6。

使用交替优化的训练方式来训练Faster R-CNN模型，交替优化的训练方式主要包括四个步骤：

- (1) 训练区域建议网络RPN；
- (2) 使用上一步提取出的候选区域训练Fast R-CNN；

(3) 固定共享卷积层的参数，重新训练RPN；

(4) 用上一步生成的新的候选区域训练Fast R-CNN。

训练参数设置为随机梯度下降算法，冲量为0.9，初始学习率为0.001，周期为20次，其余参数取默认值。对于包含8624张图片的训练数据集，模型训练共耗时约26小时。

用训练好的模型对各种类型的车辆图片进行位置检测，检测效果如图 4.8 所示。由结果可见，对于 9 种类型的车辆，Faster R-CNN 模型均可以标注出车辆在图片中的位置。



图 4.8 Faster R-CNN 部分检测结果

4.5.2 基于高斯混合模型的背景差分法检测结果

使用基于高斯混合模型的背景差分法检测车辆的位置，无需预先训练模型，只需要将参数调整到合适值即可进行检测。其原理是根据视频中的连续几帧图像建立静止的背景模型，然后将待检测图像与背景模型进行差分，从而得到视频中

运动的车辆目标的位置。本部分的核心程序如图4.9所示。

```

12     detector = vision.ForegroundDetector('NumGaussians', 5, ...
13         'NumTrainingFrames', 30, 'MinimumBackgroundRatio', 0.7);
14     blobAnalyser = vision.BlobAnalysis('BoundingBoxOutputPort', true, ...
15         'AreaOutputPort', true, 'CentroidOutputPort', true, ...
16         'MinimumBlobArea', 400);
17     while cont
18         currentStep = currentStep + 1;
19         frame = readFrame(videoReader);
20         mask = detector(frame);
21         mask = imopen(mask, strel('rectangle', [3,3]));
22         mask = imclose(mask, strel('rectangle', [25, 25]));
23         mask = imfill(mask, 'holes');
24
25         [~, centroids, bboxes] = blobAnalyser(mask);
26         mask = uint8( repmat(mask, [1, 1, 3])) .* 255;
27
28         imwrite(mask, strcat(num2str(currentStep), '.jpg'), 'jpg');
29         videoPlayer(mask);
30
31         cont = hasFrame(videoReader) && isOpen(videoPlayer);
32     end

```

图 4.9 基于高斯混合模型的背景差分法核心程序

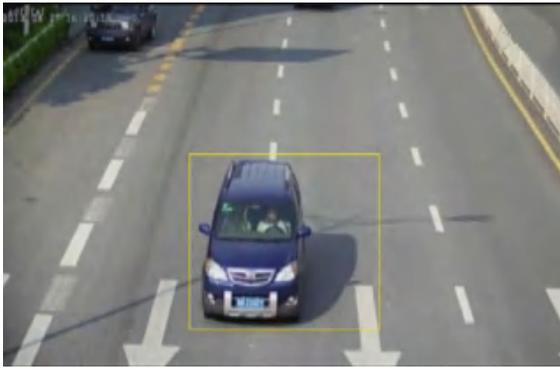
使用高斯混合模型来建立背景模型，然后将视频帧与背景模型进行比较以确定像素点属于前景或者背景，使用3个高斯分布函数进行建模，训练背景模型的初始视频帧数设置为40，其取值与视频总帧数有关，用于确定背景模型的阈值设为0.7，表示像素点被视为背景值的最小可能性为0.7。

使用形态学操作来消除结果中的噪声，先进行腐蚀操作以滤除残余背景和绝大部分噪声，减少前景图像中的毛刺，然后膨胀操作拟合裂开的细小缝隙，将原本不连通区域连接成一个区域，从而实现图像的重新恢复。本处使用腐蚀元素是 3×3 的正方形，膨胀元素是 15×15 的正方形。

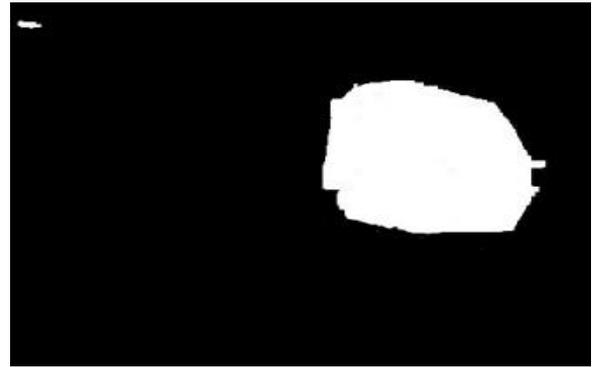
然后使用Blob分析计算图像中所有像素点的前景掩码，画出相连接的前景像素组的区域边界框，即为车辆的位置。

用基于高斯混合模型的背景差分法检测几组视频中的运动车辆的位置，其中两组结果如下图4.10所示，其中左侧图片为检测结果，右侧为对应的背景差分法检测出的前景图片。

图4.10检测结果显示，高斯混合模型方法能够大致检测出两种天气情况下的运动车辆。图a)晴天情况下，车辆区域标注框偏左，从前景图片可以看出，模型将车辆在太阳下产生的投影也检测成运动前景。图b)阴天情况下，则能够较准确检测出目标车辆位置，观察前景图像可知，虽然车辆轮廓并不清晰，但整体检测效果上无太大偏差。



a) 小轿车、晴天



b) 面包车、阴天

图 4.10 部分车辆检测效果

由此可知，基于高斯混合模型背景差分法能够检测出简单的场景下的运动车辆的位置，但对于稍复杂的自然场景，如车辆投影等，检测出的车辆位置存在一定偏差。基于高斯混合模型背景差分法是通过建立高斯混合背景模型，将当前帧与背景模型差分后得到运动车辆，其本质是检测出变化的像素点。因而对于类似于车辆投影和移动的阴影等因素产生的影响，无法通过调整算法参数等方法消除。

因此，基于高斯混合模型的目标检测方法优点在于计算简单同时能够快速检测出运动目标，但其缺点是对于复杂自然环境下的图像，其检测结果存在一定误差，因而不适用于本文中的实际桥梁上的车辆检测问题。相比之下，Faster R-CNN 在各种情况下都能实现准确检测，其缺点在于前期需要耗费大量资源去收集大量数据库以训练模型且对计算机要求高，但得益于模型训练时学习了大量车辆目标的特征，该方法不受自然环境变化的影响，在训练数据库足够大的情况下，在复杂的实际情况能够准确识别车辆目标。

4.6 本章小结

本章介绍了目标检测的常用方法，然后用深度学习Faster R-CNN和基于高斯混合模型背景差分法两种方法分别进行了车辆检测实验。首先训练Faster R-CNN模型，对第三章收集的车辆图片数据集进行标注并将其和数据集图片作为模

型输入，使用预训练过的VGG-16作为模型结构中的共享卷积层，用交替化训练方式训练Faster R-CNN模型，对训练好的模型的检测效果进行实验，结果表明Faster R-CNN能够检测出不同环境下不同车型图片中的车辆目标。然后用基于高斯混合模型的目标检测方法检测视频中的车辆，使用了3个高斯分布函数来建立背景模型，结合形态学操作来去除噪声，将原始图像与背景模型进行差分得到运动车辆，最后使用Blob分析输出检测目标的边界框等信息。结果显示，该方法对视频中的运动车辆的检测随着环境的变化存在不同程度的误差。

基于高斯混合模型的检测方法虽然实现简单快速，但对于复杂环境下的问题处理能力不足，而Faster R-CNN虽然前期耗费了大量资源，但其检测效果很稳定，能够准确检测出各种环境下不同类别车辆在图片中的位置。

第5章 车辆跟踪及车辆自动检测跟踪分类系统设计

5.1 引言

车辆检测算法能够检测出视频图像帧中车辆目标的位置，然而，进一步辨识视频流中连续图像帧检测出的图像是否为同一车辆目标则是车辆的跟踪问题。车辆跟踪是在当前视频图像帧检测出车辆在图片中的位置的基础上，通过跟踪算法提取车辆的特征信息，预测车辆在下一视频图像帧的位置，同时建立同一车辆在不同视频图像帧之间的关联。本章将利用上一章训练好的Faster R-CNN车辆检测器的检测结果作为车辆目标跟踪的初始化信息，用卡尔曼滤波跟踪器来实现车辆的跟踪，然后用基于空间位置信息的最近距离关联方法实现多个车辆目标之间的数据关联，最终实现对视频中出现的车辆自动检测跟踪。最后结合第二章建立的车辆分类方法和车辆信息库、第三章训练的VGG-16车辆分类模型、第四章的车辆检测模型以及本章的车辆跟踪算法，建立了一个车辆自动检测跟踪分类系统。

5.2 目标跟踪常用算法

目标跟踪问题是在第一帧自动检测目标或者手动选取目标，提取目标的特征信息，然后对后续帧中目标位置进行预测，也即在连续图像帧之间，构建基于目标位置、形状、速度、色彩、纹理等特征的匹配问题。

目标跟踪包括目标初始状态的确立、跟踪模型的建立、后续运动估计和确定目标位置等主要步骤。常采用手动标定或检测的方法确定目标初始状态，锁定兴趣目标所在区域，提取目标特征信息建立目标模型。跟踪模型的建立包括对目标特征（边缘、颜色、纹理等）的描述以及特征相似性的度量。接着通过搜索算法来快速预估目标的运动轨迹和目标在下一帧的区域。最后，在跟踪模型和后续运动估计的基础上，获取目标最可能的位置，实现目标位置的确定。

按有无检测过程的参与，目标跟踪可分为生成式跟踪和判别式跟踪两种。生成式跟踪方法是在目标检测的基础上对前景目标按制定的方法进行建模，在跟踪帧中搜索与目标模型相似度最高的区域并进行跟踪，包括均值漂移算法、粒子滤波算法和卡尔曼滤波算法等。判别式跟踪方法是通过每一帧视频图像进行检测来获取目标信息进而实现跟踪，也称为基于检测的跟踪方法，把目标跟踪问题当成前景和背景的二分类问题，通过分类器在当前帧中搜索与背景区域区别最大的前景区域。

传统的跟踪算法主要是生成式跟踪方法，包括均值漂移、卡尔曼滤波和粒子滤波等，近年兴起的判别式跟踪方法主要包括相关滤波方法、深度学习等方法。

5.2.1 均值漂移算法

均值漂移算法是一种无参概率估计方法，通过计算均值偏移量，按相对于核函数的均值偏移量向当前点移动，不断迭代直至满足条件^[61]。

均值漂移算法的主要流程为，在第一帧视频画面中人工标注出跟踪目标区域，提取该区域上的灰度图特征后用核函数对其进行平滑处理，将处理结果归一化后即输出跟踪目标模型的概率密度函数。用相似性函数计算目标模型特征与候选区域特征的相似程度，从中选出相似值最大的候选区域模型，然后计算出其与目标模型之间的均值偏移量。经过迭代，目标沿梯度方向由当前帧位置逐渐向目标真实位置移动，直至移动的距离小于阈值或迭代周期达到最大值，迭代结束，得到目标位置。

均值漂移原理如图5.1所示，图中圆形区域表示所有样本区域，中间实心黑点表示目标模型基准点，空心点表示样本，箭头表示由目标模型到样本点的偏移向量，可见偏移向量的均值指向梯度方向即样本分布最密集的区域。

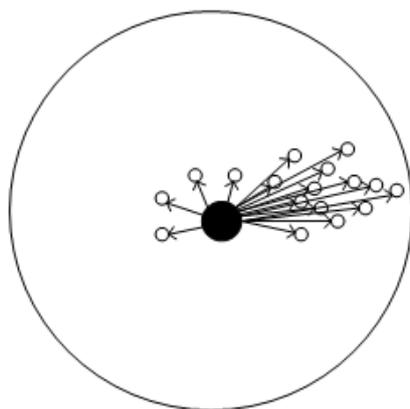


图 5.1 均值漂移向量原理图

均值漂移跟踪算法简单易于实现，有一定的旋转不变性、抗形变能力，能够达到实时跟踪的要求，但其缺点有：对目标特征的描述不完整，易受环境和背景的干扰；由于核函数的大小固定，跟踪过程中目标尺度变化时不能调整跟踪窗口大小；当目标出现遮挡和形变时跟踪效果不好；对于目标运动速度快的情况，容易出现跟踪失败等。

5.2.2 粒子滤波跟踪算法

粒子滤波跟踪算法是一种非参数滤波方法，能够免受噪声的干扰估计目标的运动状态。粒子是指在状态空间中带有权值的离散变量，粒子滤波对状态空间中的粒子进行随机采样来求解其近似概率密度函数并实现后验估计。

求解粒子滤波首先要进行粒子采样，从建议分布中提取一组新粒子，然后根据概率分布与贝叶斯公式计算权值，最后获取估值，得到系统的状态。

基于粒子滤波的跟踪算法根据目标特征建立粒子分布，接着通过迭代计算出

粒子状态的均值从而算出后验概率，然后估算出样本状态空间的最小方差概率。由于其无参数估计的特性，粒子滤波适用于非线性动态问题。目前该方法仍有检测效果易受背景和环境的影响，实时性不足，容易出现粒子退化等问题未解决。

5.3 基于卡尔曼滤波的车辆跟踪

传统的目标跟踪算法中，均值漂移算法跟踪过程中跟踪窗口大小固定，由于车辆跟踪问题中车辆的大小通常变化较大，因而不适用于车辆跟踪问题。而粒子滤波跟踪算法计算过程复杂，实时性不足。桥梁上的车辆跟踪问题通常是多车辆跟踪问题，本文用基于卡尔曼滤波的目标跟踪算法来进行单一车辆跟踪，用基于空间位置信息的最近距离关联方法进行多个目标之间的数据关联，从而实现多车辆跟踪。

5.3.1 卡尔曼滤波

卡尔曼滤波是一种递归滤波，以线性最小方差估计为基础，在贝叶斯估计理论的基础上，引入状态方程建立状态模型，利用物体运动轨迹平滑的先验知识，对跟踪系统进行最小方差的最优估计。定义物体从上一状态到这一状态转换的过程为状态转移，对当前物体的检测为观测状态，用线性函数表示这两个过程，将所有噪声干扰看作相互独立的高斯白噪声。对状态转移后的新预测状态和观测状态做加权平均，获得目标位置。

卡尔曼滤波的主要步骤为，将上一时刻检测到的目标作为初始条件（上一状态），预测当前时刻的状态变量，然后在预测点附近区域进行特征提取，获取目标的观测向量，最后利用观测向量对状态向量不断迭代更新，获得当前时刻的最佳状态估计。

卡尔曼滤波包括借助状态转移函数预测当前状态的预测阶段和结合观测值优化预测部分的更新阶段，目标状态转移方程为：

$$K_i = AK_{i-1} + \xi_{i-1} \quad (5.1)$$

目标观测方程为：

$$L_i = H_i K_i + \eta_i \quad (5.2)$$

其中， K_i 、 K_{i-1} 分别为当前时刻状态向量和上一时刻状态向量， A 表示上一时刻到当前时刻的状态转移矩阵， L_i 为当前时刻的观测状态， H 为观测矩阵， ξ_{i-1} 为系统噪声， η_i 为观测噪声，其中 ξ_{i-1} 和 η_i 为相互独立的高斯白噪声。

使用卡尔曼滤波器对目标进行跟踪分为四个步骤，分别是滤波器初始化、预测阶段、目标匹配和状态更新阶段，卡尔曼滤波器的循环流程如图5.2所示。

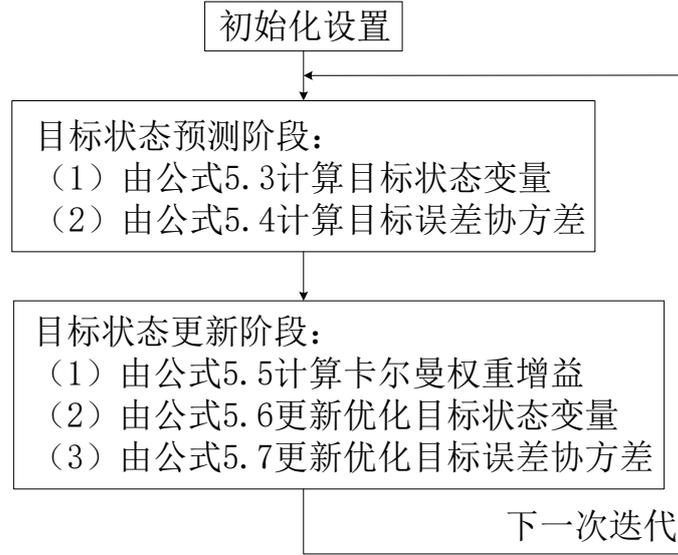


图 5.2 卡尔曼滤波迭代流程

滤波器初始化包括设置目标初始位置、初始速度、状态转移矩阵 A 的初始值、观测矩阵 H 的初始值、观测噪声协方差 R 的初始值和系统噪声协方差 Q 的初始值。

预测阶段是通过上一时刻目标运动状态估计当前时刻的状态，包括由初始值计算当前时刻的状态变量和误差协方差，状态变量预测方程为：

$$\hat{K}_i = AK_{i-1} + \zeta_{i-1} \quad (5.3)$$

误差协方差预测方程为：

$$\hat{P}_i = AP_{i-1}A^T + Q_i \quad (5.4)$$

状态更新阶段包括卡尔曼权重增益更新和状态变量及误差协方差的优化，卡尔曼权重增益方程为：

$$\omega_i = \hat{P}_i H^T (H \hat{P}_i H^T + R_i)^{-1} \quad (5.5)$$

状态变量更新优化公式为：

$$\hat{K}_i^+ = A\hat{K}_i + \omega_i(L_i - HA\hat{K}_i) \quad (5.6)$$

误差协方差更新优化公式为：

$$\hat{P}_i^+ = (1 - \omega_i H) \hat{P}_i \quad (5.7)$$

其中， \hat{K}_i 、 \hat{P}_i 表示当前时刻的预测状态和误差协方差预测， Q_i 为系统噪声的协方差， ω_i 为卡尔曼优化权重， R_i 为观测噪声的协方差， \hat{K}_i^+ 和 \hat{P}_i^+ 分别表示当前时刻的预测优化状态和优化误差协方差。

采用卡尔曼滤波器进行运动目标的跟踪预测，即使在参数初始化不准确的情况下，如初始速度估算误差，能够通过卡尔曼滤波迭代计算准确预测运动目标的真实位置。

5.3.2 多目标跟踪

卡尔曼滤波算法是对单目标进行跟踪的算法，若要实现多车辆跟踪则需对其

进一步改造。用检测器检测到的车辆位置来初始化卡尔曼滤波跟踪器，但要实现多目标跟踪，需要使用数据关联算法来实现不同目标之间的区分以及跟踪轨道的自动更新。多目标跟踪的关键在于当前检测目标与已经跟踪的目标之间建立一一对应的关系。数据关联算法联系同一目标检测结果的同时能够区分出不同目标，得到各个目标的运动轨迹。

本文使用的数据关联算法是基于空间位置信息的最近距离关联方法，即建立当前检测到的目标合集链表 $\{O_d\}$ 与已经跟踪的目标合集链表 $\{O_t\}$ 各目标之间的对应关系，从而实现跟踪轨迹的关联。

设检测链表 $\{O_d\}$ 中第 k 个目标的中心坐标为 (x_{dk}, y_{dk}) ，跟踪链表 $\{O_t\}$ 中第 l 个目标的中心坐标为 (x_{dl}, y_{dl}) ，二者的空间位置距离为：

$$D(k,l) = \left[(x_{dk} - x_{dl})^2 + (y_{dk} - y_{dl})^2 \right]^{\frac{1}{2}} \quad (5.8)$$

计算当前帧中每个检测目标与跟踪目标间的空间距离 $D(k,l)$ ，当检测目标有 K 个，跟踪目标 L 个时，其空间距离关联矩阵为：

$$D = \begin{pmatrix} D_{11} & D_{12} & \cdots & D_{1(L-1)} & D_{1L} \\ D_{21} & D_{22} & \cdots & D_{2(L-1)} & D_{2L} \\ \vdots & \vdots & D_{kl} & \vdots & \vdots \\ D_{(K-1)1} & D_{(K-1)2} & \cdots & D_{(K-1)(L-1)} & D_{(K-1)L} \\ D_{K1} & D_{K2} & \cdots & D_{K(L-1)} & D_{KL} \end{pmatrix} \quad (5.9)$$

设定关联矩阵 D 的最大阈值为 D_{max} ，当检测目标与跟踪目标间的距离 D_{kl} 大于阈值 D_{max} 时，则判定两个目标之间不存在关联，为不同目标；接着在关联矩阵 D 中寻找数值最小的元素 D_{kl} ，若 D_{kl} 小于阈值，则关联成功，为同一个目标；然后将矩阵 D 中的第 k 行和第 l 列删除，生成新的矩阵 D_i ，对矩阵 D_i 重复上述操作，直到所有检测目标与跟踪目标完成匹配，最后余下的矩阵 D_N 。

对于成功关联的目标，将新检测到的信息更新到对应物体的轨迹信息中，校正跟踪器的偏移。对于剩下的矩阵 D_N ，对应检测集合链表中的目标认为是新出现目标，为其创建新的卡尔曼滤波跟踪器并进行初始化；对应跟踪集合链表中的目标判定为离开画面的目标，将其删除即可。

5.3.3 实验结果

结合上述理论，在Matlab2018a软件中编写基于卡尔曼滤波跟踪的多车辆自动跟踪程序，该程序的主要流程如图5.3所示。

首先加载第四章中训练好的Faster R-CNN模型和第三章中训练好的VGG-16模型分别作为车辆检测器和车辆分类器，然后设置多车辆目标跟踪器的预设值和卡尔曼滤波器的初始参数值。接着用车辆检测器检测图像帧中的车辆位置，分别

剪切出检测到的车辆图片并输入车辆分类器进行分类。然后用检测结果更新多目标跟踪器，包括为满足条件的目标创建一个卡尔曼滤波跟踪器，卡尔曼滤波器通过计算每个检测出的车辆的状态向量和状态向量协方差矩阵来预测该目标在下一帧图像中的位置，利用数据关联算法建立已经跟踪的目标集和新检测的目标集之间的对应关系，更新每个目标的跟踪数据并为满足条件的目标创建新的跟踪轨道，同时删除满足条件的消失的目标。

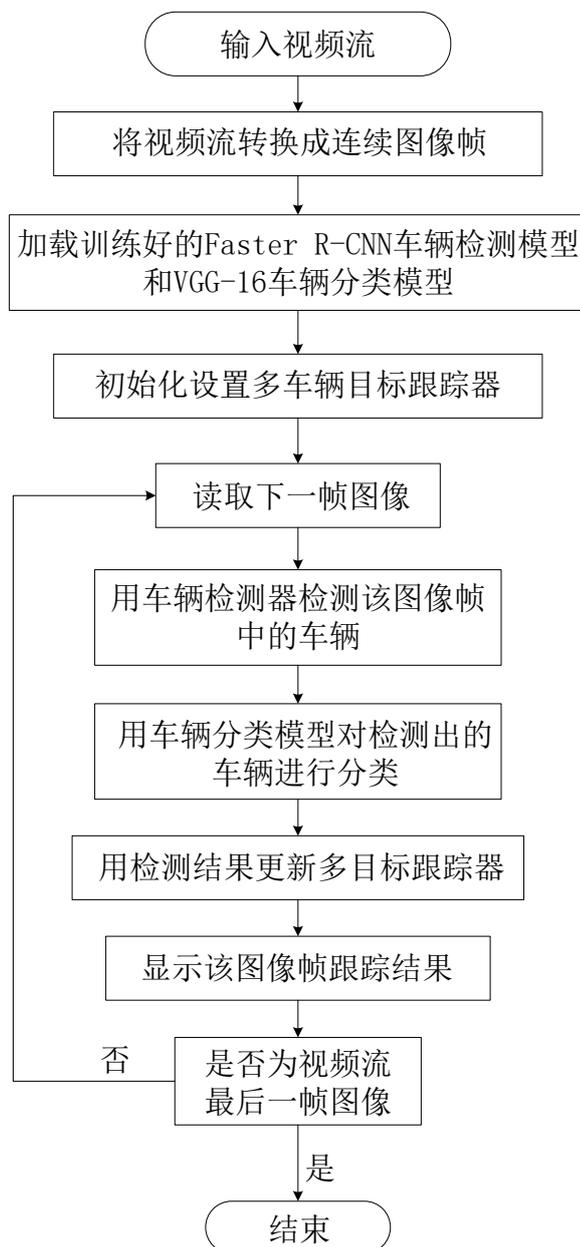


图 5.3 车辆跟踪程序流程

本文将多目标跟踪器的参数设置为：空间距离关联矩阵的最大阈值 D_{max} 为50，创建新轨道的确认参数为[3,5]（表示多目标跟踪器的数据连续更新5次中同一物体被检测到3次，则为该目标创建新的跟踪对象轨道），删除轨道的确认参数为5（表示多目标跟踪器连续更新5次且均无对应检测信息输入即认为该目标离开画面，则删除该跟踪轨道）。

同时卡尔曼滤波器的初始参数设置为：将检测结果作为目标位置的初始值，目标的初始速度设置为0，将连续图像帧间的目标运动近似看作匀速运动，则设置初始状态转移矩阵 A 为：

$$A = \begin{pmatrix} B & 0 & 0 & 0 \\ 0 & B & 0 & 0 \\ 0 & 0 & B & 0 \\ 0 & 0 & 0 & B \end{pmatrix}, \text{其中 } B = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad (5.10)$$

相应地，设置观测矩阵 H 的初始值为：

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (5.11)$$

为了去除每次预测结果中的噪声，设置观测噪声协方差矩阵设 R 为：

$$R = \begin{pmatrix} 50 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{pmatrix} \quad (5.12)$$

为了去除模型未考虑部分因素的影响，设置系统噪声协方差矩阵 Q 为：

$$Q = \begin{pmatrix} C & 0 & 0 & 0 \\ 0 & C & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & C \end{pmatrix}, \text{其中 } C = \begin{pmatrix} 0.25 & 0.5 \\ 0.5 & 1 \end{pmatrix} \quad (5.13)$$

设置好以上参数后，就完成了卡尔曼滤波算法的准备工作。接下来载入待识别视频即可对视频中出现的车辆进行自动跟踪。

为了对本文算法的实用性进行验证，笔者用佳能5D单反相机在长沙市岳麓区观音港大桥的一侧拍摄了桥面车辆行驶视频，视频分辨率为1920×1080，帧率为25帧/秒。相机固定在三脚架上，离地面高度为1.6m。拍摄现场如图5.4所示。

由于拍摄的视频分辨率过高将导致程序运行过于缓慢，故将拍摄得到的视频素材全部转换成分辨率720×560、帧率25帧/秒的视频。

对所拍摄的一个时长为57.6s，共有1440帧的视频进行跟踪识别，图5.5、5.6分别展示了其中两组车辆的跟踪结果。

上述两组结果表明，基于卡尔曼滤波的车辆跟踪算法能够稳定地在视频图像帧中跟踪车辆。



a) 观音港桥俯视图



b) 相机架设位置

图 5.4 视频拍摄现场



a) 第 447 帧



b) 第 448 帧



c) 第 449 帧



d) 第 450 帧

图 5.5 面包车跟踪示例



a) 第 665 帧



b) 第 666 帧



图 5.6 货车跟踪示例

5.4 车辆自动检测跟踪分类系统设计

5.4.1 系统框架组成

为实现通过桥梁上沿桥摄像机监控视频确定桥面上车辆荷载区间及其实时分布，本文建立了一个系统，由三个模块组成，其示意图如图5.7所示。

该系统主要由目标检测、目标跟踪和目标分类三个模块组成，目标检测模块使用车辆图片数据集和标注信息和预训练的VGG-16作为共享卷积层，交替训练Faster R-CNN的两个组成部分RPN和Fast R-CNN，最后得到Faster R-CNN模型用于目标检测。

目标跟踪模块以目标检测结果为初始信息，创建卡尔曼滤波器进行跟踪，同时使用数据关联算法来实现多目标的区分和跟踪，然后用跟踪结果更新每个目标轨道信息。目标分类模块用车辆图片数据集和类型标注信息作为输入，预训练的模型和数据增强手段来训练模型，得到小轿车、面包车、中型客车、大型客车、公交车、2轴4轮货车、2轴6轮货车、3轴货车、4轴货车共9种类别输出的VGG-16模型。

系统首先提取出输入的桥面监控视频中的视频帧，将视频帧图像输入Faster R-CNN以检测图像中是否有车辆及车辆所在位置，随后卡尔曼滤波跟踪器以检测结果作为初始信息，结合数据关联算法，给满足一定条件的检测结果分配轨道，卡尔曼滤波跟踪器输出的位置结果更新轨道信息并预测车辆下一帧的位置。然后将一个轨道中间帧的车辆位置边界框内的图片调整成 224×224 大小后用VGG-16模型识别，得到该轨道车辆目标的类别信息。

结合第二章收集的常见车辆数据库车辆类型对应的轴距、轴重区间，即可知道桥面监控视频中每一时刻车辆所属类型、轴距区间、轴重区间、所在位置和实时轨迹。最后结合特定场景下监控相机在桥梁上的安装位置以及安装高度信息，即可将车辆的像素位置转换为实际位置。从而使由桥梁上拍摄的视频获得桥梁上

的车辆荷载区间及其实时分布成为可能。

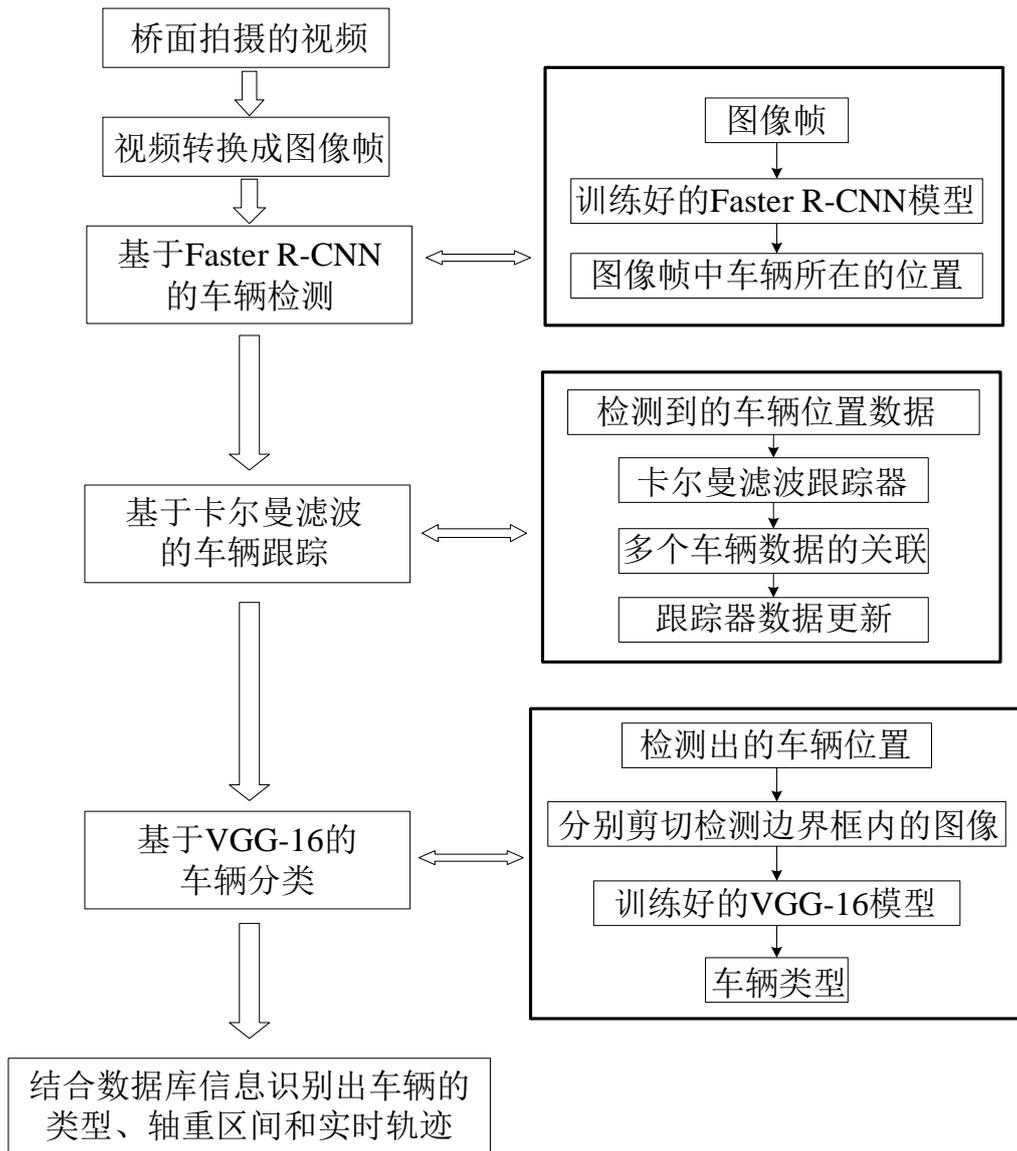


图 5.7 系统示意图

5.4.2 系统功能介绍

为了方便非专业人员的使用，本文创建了MATLAB的GUI图形用户界面来实现车辆自动检测跟踪分类系统。

GUI是指采用图形方式显示的计算机操作用户界面，用户能够通过点击屏幕上的图标或菜单选项来执行选择命令、调用系统文件或其他命令，使得用户无需学习编程语言即可完成特定任务。

本文建立的车辆自动检测跟踪分类系统的GUI界面如图5.8所示，界面由视频区域、视频信息、载入模型和控制面板四个板块组成。

视频区域用于展示处理的视频图像，下方滑动条及文本用来选择展示视频帧中的某一帧图像，播放键、暂停键、停止键和截图键用来控制视频展示区域实现

对应功能。



图 5.8 GUI 界面

视频信息区域用来显示所载入视频的信息，包括帧总数、帧宽度、帧高度、帧率、时长、视频格式和视频路径。

载入模型部分有目标检测模型和目标分类模型两个功能键，每次打开系统在识别前首先载入目标检测模型和目标分类模型，这一模块便于模型重新训练后的更新。

控制面板整合了所有操作，包括打开视频文件、获取视频信息、获取图像序列、车辆检测跟踪、运动轨迹显示、车辆类型识别和结果显示命令。

该系统共有12个功能键，其中各个功能键的介绍如下：

- (1) 按键A为载入模型模块目标检测模型功能键，在进行车辆检测前首先需要载入目标检测模型，便于及时更新性能更好的模型。
- (2) 按键B为载入模型模块目标分类模型，在进行车辆分类前首先需要载入目标分类模型，便于模型的更新和替换。
- (3) 按键C为打开视频文件，点击该按钮将弹出系统文件选择窗口，选择即将处理的文件即载入工作区，同时在视频信息区域显示视频路径信息。按键D为获取视频信息，点击该按钮后在视频信息区域将展示该视频的时长、格式、帧总数、帧宽度、帧高度和帧率等信息。
- (4) 按键E实现获取图像序列功能，将载入的视频逐帧转换为图像。
- (5) 按键F实现车辆检测跟踪功能，点击该按钮后在视频区域将逐帧显示视频中车辆的检测跟踪结果。

(6) 按键G为运动轨迹显示, 点击该按键后在视频区域将逐帧显示车辆的运动轨迹线。

(7) 按键H为车辆类型识别, 点击该按键后在视频区域将逐帧显示车辆的类型。

(8) 按键I为结果显示, 将同时展示车辆的运动轨迹线、车辆在视频帧中的位置和车辆对应类型标注等信息。

(9) 按键J为播放键, 在载入视频后, 任意时刻点击该键即可在视频区域播放原始视频帧。

(10) 按键K为暂停键, 在任意操作时点击该键可以暂停视频的播放和其他操作。

(11) 按键L为停止键, 在任意操作时点击该键即可停止当前操作并清空视频播放区域。

(12) 按键M为截图, 在任意操作时点击该键即可将当前视频区域显示画面截图并保存。

5.4.3 系统运行结果展示

对该系统的效果进行测试和展示, 首先以此载入目标检测模型和目标分类模型, 接着点击打开视频文件按钮, 载入一段处理后的观音港桥上的视频素材, 然后依次按下获取视频信息和获取图像序列按钮。

图 5.9 为点击车辆检测跟踪功能键后的第 257 帧图像截图, 从图中可以看出, 该段视频总时长为 42.6s, 分辨率为 720×576, 帧总数为 1065 帧, 帧率是 25.0 帧/秒。

图 5.10 为点击车辆类型识别功能键后, 视频的第 259 帧图像下车辆自动检测跟踪分类系统界面情况。由图可见, 图中车辆为 2 轴 6 轮货车的概率为 98.7%。

图 5.11 为点击结果显示功能键后, 视频的第 277 帧图像下车辆自动检测跟踪分类系统界面情况。由图可见, 该车为 2 轴 6 轮货车的概率是 93.5%, 图中红色曲线为该 2 轴 6 轮货车在连续图像帧中的位置的连线, 即该车辆的轨迹线。

图 5.12 为点击结果显示功能键后, 视频的第 813 帧图像下车辆自动检测跟踪分类系统界面情况。由图可见, 该车为面包车的概率是 100.0%, 图中红色曲线为该面包车在连续图像帧中的位置的连线, 即该车的轨迹线。



图 5.9 车辆检测跟踪



图 5.10 第 259 帧车辆类型识别结果



图 5.11 第 277 帧结果显示



图 5.12 第 813 帧结果显示

5.5 本章小结

本章介绍了几种目标跟踪算法及其原理，用基于卡尔曼滤波的目标跟踪算法来进行车辆目标的跟踪。将Faster R-CNN检测的车辆位置作为卡尔曼滤波算法的初始信息创建跟踪器，用基于空间位置信息的最近距离关联方法实现了多目标的数据关联。对一个视频进行实验，结果表明基于卡尔曼滤波的目标跟踪算法对于复杂实际场景下的各种类型车辆能够准确跟踪。

然后结合第二章建立的车辆信息库、第三章的车辆类型识别模型VGG-16、第四章的车辆检测模型Faster R-CNN和卡尔曼滤波跟踪器，建立了车辆自动检测跟踪分类系统。该系统输入桥梁上拍摄的视频，经过Faster R-CNN检测视频帧中的车辆位置并以此为基础进行跟踪，然后将车辆所在区域图片输入到VGG-16以识别车辆类型，得到桥面上车辆的类型、实时轨迹等信息。同时，建立了实现该车辆自动分类检测跟踪系统的GUI图形用户界面，用户无需懂得编程语言即可实现相关功能。

总结与展望

桥梁结构在国民经济和社会生活中发挥着重要作用，然而随着使用时间的增加，大量现役桥梁出现了不同程度的病害。随着交通运输重要性的增加，部分大型桥梁已经安装了桥梁结构健康检测系统对桥梁结构的实时状态进行监控。然而，我国仍有数量庞大的中小型桥梁由于成本原因无法实施监测。因而，对在役桥梁进行低成本、无接触、不需封闭交通的健康监测的问题亟待解决。本文提出了基于桥面拍摄的视频进行车辆自动检测跟踪分类的系统。该系统利用深度学习算法 Faster R-CNN 对车辆位置进行检测，通过卡尔曼滤波跟踪算法和数据关联技术对多个车辆目标进行跟踪，结合深度卷积神经网络 VGG-16 和建立的车辆数据信息库对车辆类型进行识别，从而得到被测桥梁上车辆的类型、轴重区间和实时运动轨迹。本文的主要成果如下：

(1) 车辆分类标准和车辆数据信息库的建立。通过收集和分析各国关于车辆分类的规定，结合我国相关国家规范、行业标准以及相关研究成果，建立了车辆细分类标准。根据车辆外型与轴重的对应关系、使用功能、载客数量、车轴数量等将车辆分为：小轿车、面包车、中型客车、大型客车、2轴4轮货车、2轴6轮货车、3轴货车、4轴货车、多轴货车、专项作业车和三轮汽车等共12类。在该细分类标准的基础上，通过收集汽车公告信息，建立了包含车辆轴距区间、轴重区间和总重区间等信息的车辆数据库。

(2) 车辆类型识别模型卷积神经网络的训练。结合本文车辆标准，通过截取桥梁拍摄的视频帧图像和网络搜集车辆图片两种方式，首先收集建立了车辆数据集，该数据集有9种类型车辆共8624张图片。然后利用一系列方法来对 AlexNet 和 VGG-16 两种模型进行训练，训练结果表明，在数据集较小的情况下用全训练的方式导致 AlexNet 模型训练不足，识别精度差。微调在大型公开数据集上训练好的 AlexNet 的方法能够将 AlexNet 模型精度提高到 87.55%。然后，用数据增强方法能够进一步解决数据不足的问题，使用数据增强方法的 AlexNet 模型的平均精度提高到了 93.4%。最后，微调预训练模型和数据增强两种方法下训练的 VGG-16 模型平均精度达到了 98.17%。同时，相同训练方式下的 VGG-16 精度比 AlexNet 高，表明增加深度卷积层的数量能够提高网络的学习能力和精度。

(3) 车辆位置的检测算法。本文对比了基于高斯混合模型背景差分法和深度学习 Faster R-CNN 两种方法在车辆位置监测中的效果。对比结果表明，基于高斯混合模型背景差分法虽然实现简单，但对于复杂环境下的问题处理能力不足。而 Faster R-CNN 方法虽然前期耗费了大量资源，但其检测效果很稳定，能够准确

检测出各种环境下不同类别车辆在图片中的位置。

(4) 车辆位置跟踪算法研究。本文将Faster R-CNN检测到的车辆位置作为卡尔曼滤波目标跟踪算法的初始信息创建跟踪器，用基于空间位置信息的最近距离关联方法实现了多目标自动跟踪和丢弃。对一个视频进行实验，结果表明基于卡尔曼滤波的目标跟踪算法对于复杂实际场景下的各种类型车辆能够准确跟踪。

(5) 车辆自动检测跟踪分类系统的设计。结合前文训练好的车辆分类模型VGG-16、车辆检测模型Faster R-CNN和卡尔曼滤波跟踪器，设计了基于GUI图形用户界面的车辆自动检测分类系统。结合建立的车辆数据信息库信息，输入桥梁拍摄的视频即可得到桥面上车辆的类型、轴重区间和实时轨迹等信息。

本文提出的方法是一种低成本、无接触、无需封闭交通的识别桥梁上车辆荷载的方法，为实现基于非接触式计算机视觉技术识别桥梁上车辆荷载的大小及其时空分布奠定了基础，是实现非接触式机器视觉监测技术的桥梁结构健康监测和状态快速评估系统的首要工作。

本文在建立车辆数据信息库和建立基于桥梁监控视频的车辆自动检测跟踪分类系统的过程中，由于受到实际条件的制约，研究中仍存在不足与有待进一步开展的问题：

(1) 本文根据汽车公告来收集车辆信息的过程中，只能建立部分车辆的数据信息库，数据库仍有待扩充。同时建立车辆轴重区间时考虑了车辆满载以及空载两种极限情况的荷载，对于货车的轴重区间范围仍有待进一步细化；

(2) 本文的车辆图片数据集有8624张车辆图片同时车辆大小比较统一，导致训练的Faster R-CNN模型对于视频中距离较远、体积较小的车辆检测能力不足，可以通过大量增加数据集图片数量和样本多样性来训练更好的车辆检测模型；

(3) Faster R-CNN模型由于其本身的结构特性，训练时计算量很大，用于实时视频的车辆位置检测时，检测速度仍有待进一步提高；

(4) 本文建立了车辆自动检测跟踪分类系统，实现了对视频内车辆的自动跟踪和分类，得到车辆的荷载区间和在视频中的实时位置。车辆在实际桥梁上的实时位置需要结合具体桥梁场景进一步分析。

参考文献

- [1] 2017年交通运输行业发展统计公报. 中国物流与采购, 2018(11).
- [2] 蔡唐涛, 贾彦武. 桥梁倒塌事故问题分析. 筑路机械与施工机械化, 2018, 35(5).
- [3] Richardson J, Jones S, Brown A, et al. On the use of bridge weigh-in-motion for overweight truck enforcement. *International Journal of Heavy Vehicle Systems*, 2014, 21(2): 83-104.
- [4] Yang Y, Cai C S, Lu D. State-of-the-art review on bridge weigh-in-motion technology. *Advances in Structural Engineering*, 2016, 19(9): 1514-1530.
- [5] 任伟新, 左小晗, 王宁波等. 非路面式桥梁动态称重研究综述. 中国公路学报, 2014, 27(7).
- [6] Ojio T, Carey C H, OBrien E J, et al. Contactless bridge weigh-in-motion. *Journal of Bridge Engineering*, 2016, 21(7): 04016032.
- [7] 基于深度学习的车辆型号识别:[华中科技大学博士论文]. 武汉: 华中科技大学, 2014.
- [8] 姜绍飞. 结构健康监测-智能信息处理及应用. 工程力学, 2009, 26(2).
- [9] 李惠. 结构健康监测数据科学与工程. 北京: 科学出版社: 2016.
- [10] Catbas F N, Zaurin R, Gul M, et al. Sensor Networks, Computer Imaging, and Unit Influence Lines for Structural Health Monitoring: Case Study for Bridge Load Rating. *Journal of Bridge Engineering*, 2012, 17(4): 662-670.
- [11] Min J-H, J. Gelo N, Jo H. Non-contact and Real-time Dynamic Displacement Monitoring using Smartphone Technologies. 2015: 40-51.
- [12] 董传智. 基于机器视觉的桥梁健康监测与状态评估:[浙江大学博士论文]. 杭州: 浙江大学, 2016.
- [13] Chen Z, Hui L, Bao Y, et al. Identification of spatio-temporal distribution of vehicle loads on long-span bridges using computer vision technology. *Structural Control & Health Monitoring*, 2016, 23(3): 517-534.
- [14] Rivera C J, Flores F W, Rivas L M, et al. Experimental image and range scanner datasets fusion in shm for displacement detection. *Structural Control and Health Monitoring*, 2017, 24(10): e1967.
- [15] Santos C A, Costa C O, Batista J. A vision-based system for measuring the displacements of large structures: Simultaneous adaptive calibration and full motion estimation. *Mechanical Systems and Signal Processing*, 2016, 72: 678-694.

- [16] Khuc T, Catbas N. Completely contactless structural health monitoring of real-life structures using cameras and computer vision. *Structural Control & Health Monitoring*, 2017, 24(1): e1852.
- [17] Feng D, Feng M Q. Experimental validation of cost-effective vision-based structural health monitoring. *Mechanical Systems & Signal Processing*, 2017, 88: 199-211.
- [18] Dong C Z, Ye X W, Jin T, et al. Identification of Structural Dynamic Characteristics Based on Machine Vision Technology. *Measurement*, 2017: S0263224117306152.
- [19] Feng D, Scarangelo T, Feng M Q, et al. Cable tension force estimate using novel noncontact vision-based sensor. *Measurement*, 2017, 99: 44-52.
- [20] 裴耀东. 基于数字图像处理的桥梁结构裂缝与位移测量研究:[湖南大学博士学位论文]. 长沙: 湖南大学, 2017.
- [21] Khuc T, Catbas F N. Structural Identification Using Computer Vision-Based Bridge Health Monitoring. *Journal of Structural Engineering*, 2017, 144(2): 04017202.
- [22] Hinton G E, Osindero S, Yee W T. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 2006, 18(7): 1527-1554.
- [23] Lecun Y, Bottou L, Bengio Y, et al. Gradient-Based Learning Applied to Document Recognition. 1998: 2278-2324.
- [24] Hinton G E, Salakhutdinov R R. Reducing the Dimensionality of Data with Neural Networks. *Science*, 2006, 313(5786): 504-507.
- [25] Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. 2012: 82-97.
- [26] Le Q V, Ranzato M A, Monga R, et al. Building high-level features using large scale unsupervised learning. In: *IEEE International Conference on Acoustics*, 2013.
- [27] Turian J P, Ratinov L A, Bengio Y. Word Representations: A Simple and General Method for Semi-Supervised Learning. In: *Acl, Meeting of the Association for Computational Linguistics*, July, Uppsala, Sweden, 2010.
- [28] Socher R, Lin C C-Y, Ng A Y, et al. Parsing natural scenes and natural language with recursive neural networks. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011: 129-136.
- [29] Scholkopf B, Platt J, Hofmann T. Efficient sparse coding algorithms. In: *International Conference on Neural Information Processing Systems*, 2007: 801-

- 808.
- [30] Dan C C, Meier U, Masci J, et al. High Performance Convolutional Neural Networks for Image Classification. In: International Joint Conference on Ijcai, 2011.
- [31] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks. In: International Conference on Neural Information Processing Systems, 2012.
- [32] Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2015, 115(3): 211-252.
- [33] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science*, 2014.
- [34] Szegedy, Christian, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
- [35] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition. 2016: 770-778.
- [36] Guo J, Xie X, Bie R, et al. Structural health monitoring by using a sparse coding-based deep learning algorithm with wireless sensor networks. *Personal & Ubiquitous Computing*, 2014, 18(8): 1977-1987.
- [37] Gul M, Catbas F N, Hattori H. Image-Based Monitoring of Open Gears of Movable Bridges for Condition Assessment and Maintenance Decision Making. *Journal of Computing in Civil Engineering*, 2013, 29(2): 04014034.
- [38] Jiang X, Mahadevan S, Yuan Y. Fuzzy stochastic neural network model for structural system identification. *Mechanical Systems & Signal Processing*, 2017, 82: 394-411.
- [39] Yeum C M, Dyke S J, Basora Rovira R E, et al. Acceleration-Based Automated Vehicle Classification on Mobile Bridges. *Computer-Aided Civil & Infrastructure Engineering*, 2016, 31(11): 813-825.
- [40] Xu Y, Li S, Zhang D, et al. Identification framework for cracks on a steel structure surface by a restricted Boltzmann machines algorithm based on consumer - grade camera images. *Structural Control & Health Monitoring*, 2018, 25(2): 1-1.
- [41] Cha Y-J, Choi W, Büyüköztürk O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, 2017, 32(5): 361-378.
- [42] Cha Y J, Choi W, Suh G, et al. Autonomous Structural Visual Inspection Using Region - Based Deep Learning for Detecting Multiple Damage Types. *Computer-*

- Aided Civil & Infrastructure Engineering, 2018, 33(9): 731-747.
- [43] Morris B T, Trivedi M M. Learning, Modeling, and Classification of Vehicle Track Patterns from Live Video. 2008: 425-437.
- [44] Chen Z, Ellis T, Velastin S A. Vehicle detection, tracking and classification in urban traffic. In: International IEEE Conference on Intelligent Transportation Systems, 2012.
- [45] Hu B, Lai J H, Guo C C. Location-aware fine-grained vehicle type recognition using multi-task deep networks. Neurocomputing, 2017, 243: 60-68.
- [46] Yu S, Yun W, Wei L, et al. A model for fine-grained vehicle classification based on deep learning. Neurocomputing, 2017, 257: S092523121730156X.
- [47] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. 2015: 91-99.
- [48] Dubská M, Herout A, Sochor J. Automatic Camera Calibration for Traffic Understanding. In: BMVC. 2014, 4(6): 8.
- [49] Liang L, Hu R, Xiao J, et al. Exploiting effects of parts in fine-grained categorization of vehicles. In: IEEE International Conference on Image Processing, 2015.
- [50] Yang L, Luo P, Change Loy C, et al. A large-scale car dataset for fine-grained categorization and verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3973-3981.
- [51] Ha J E, Lee W H. Foreground objects detection using multiple difference images. Optical Engineering, 2010, 49(49): 258-258.
- [52] Huang C L, Liao W C. A vision-based vehicle identification system. In: International Conference on Pattern Recognition, 2004.
- [53] Stauffer C, Grimson W E L. Adaptive background mixture models for real-time tracking. In: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). IEEE, 1999, 2: 246-252.
- [54] Elgammal A, Harwood D, Davis L. Non-parametric model for background subtraction. In: European conference on computer vision. Springer, Berlin, Heidelberg, 2000: 751-767.
- [55] Yao B, Liu Z, Nie X, et al. Animated Pose Templates for Modelling and Detecting Human Actions. In: IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(3): 436-452.
- [56] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object

- detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
- [57] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2015, 37(9): 1904-1916.
- [58] Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
- [59] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
- [60] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector. In: European Conference on Computer Vision, 2016: 21-37.
- [61] Fukunaga K, Hostetler L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 1975, 21(1): 32-40.
- [62] Hare S, Golodetz S, Saffari A, et al. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 2016, 38(10): 2096-2109.
- [63] Grabner H, Leistner C, Bischof H. Semi-supervised on-line boosting for robust tracking. In: European conference on computer vision. Springer, Berlin, Heidelberg, 2008: 234-247.
- [64] Godec M, Leistner C, Saffari A, et al. On-line random naive bayes for tracking. In: 2010 20th International Conference on Pattern Recognition. IEEE, 2010: 3545-3548.
- [65] Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010: 2544-2550.
- [66] Reid D. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 1979, 24(6): 843-854.
- [67] Chen B H, Huang S C. Probabilistic neural networks based moving vehicles extraction algorithm for intelligent traffic surveillance systems. *Information Sciences*, 2015, 299(C): 283-295.
- [68] Bao Y, Li H, Zhang F, et al. Compressive sampling based approach for identification of moving loads distribution on cable-stayed bridges. In: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*

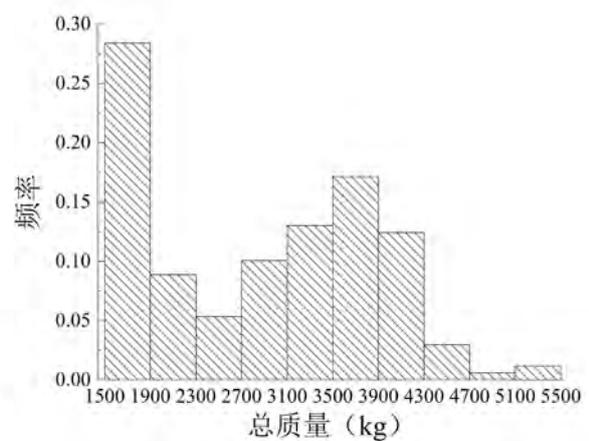
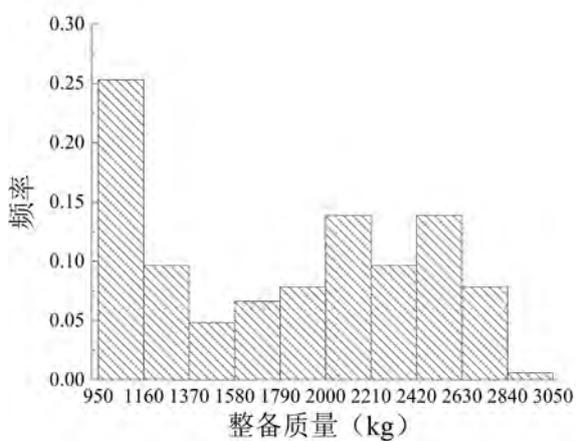
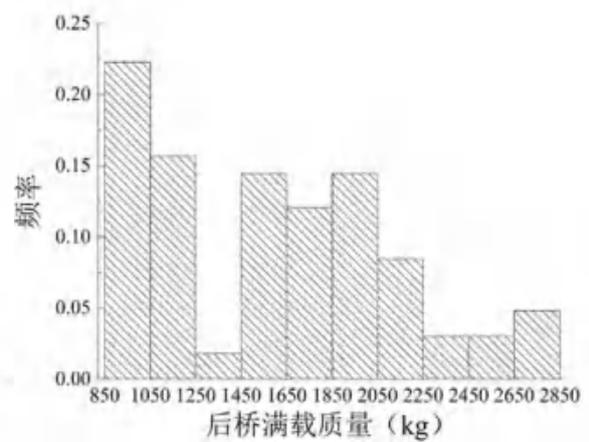
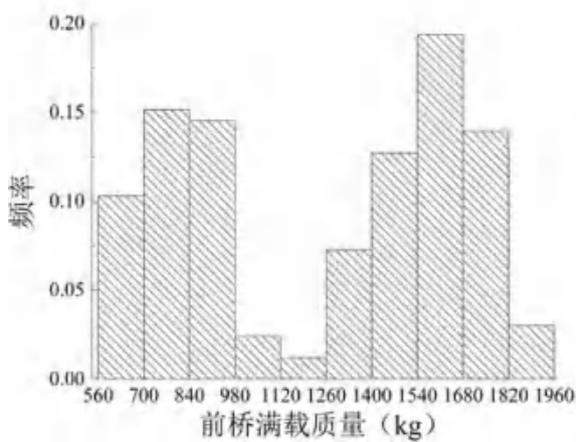
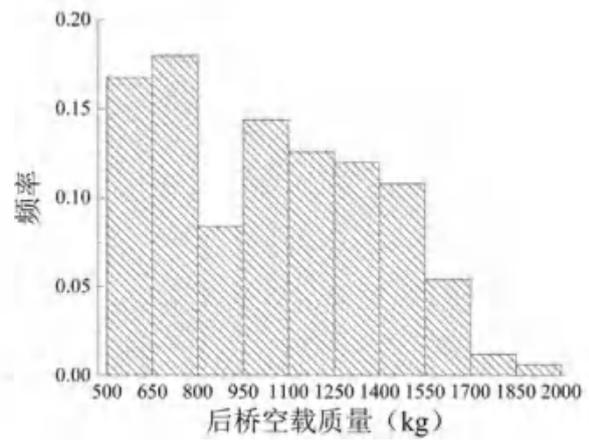
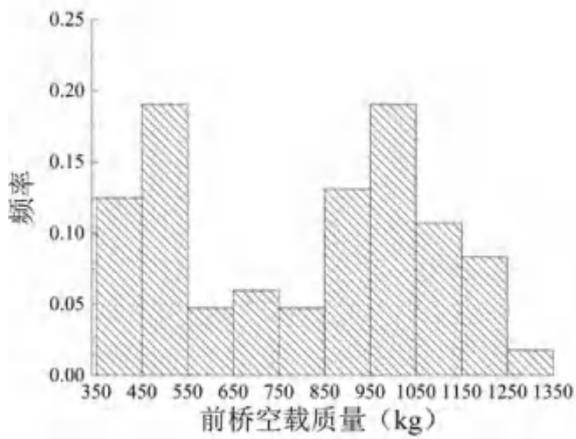
2013. *International Society for Optics and Photonics*, 2013, 8692: 86923B.
- [69] 李惠, 鲍跃全, 李顺龙等. 结构健康监测数据科学与工程. 工程力学, 2015, 32(8): 1-7.
- [70] 道路车辆类型术语和定义. ISO/WD 3833: 1999, 1999.
- [71] 闫志强. 我国车辆分类方法和标准研究与分析:[大连理工大学博士论文]. 大连:大连理工大学, 2014.
- [72] 张文斌. 公路车辆动态荷载测量及车型分类技术的研究:[哈尔滨工业大学博士论文]. 哈尔滨: 哈尔滨工业大学, 2009.
- [73] 中华人名共和国国家标准. 道路车辆分类与代码 机动车. GB 918.1-1989, 1989.
- [74] 中华人名共和国国家标准. 汽车和挂车类型的术语和定义. GB/T3730.1-2001, 2001.
- [75] 中华人名共和国国家标准. 机动车辆及挂车分类. GB/T 15089-2001, 2001.
- [76] 收费公路车辆通行费车型分类, 2003.
- [77] 机动车类型 术语及定义, 2014.
- [78] Wei C H, Chang C C, Wang S S. Vehicle classification using advanced technologies. *Transportation research record*, 1996, 1551(1): 45-50.
- [79] Zhen D, Wu Y, Pei M, et al. Vehicle Type Classification Using a Semisupervised Convolutional Neural Network. *IEEE Transactions on Intelligent Transportation Systems*, 2015, 16(4): 2247-2256.
- [80] Yuan X, Lu Y J, Sarraf S. Computer Vision System for Automatic Vehicle Classification. *Journal of Transportation Engineering*, 1994, 120(6): 861-876.
- [81] Sun C, Ritchie S. Heuristic vehicle classification using inductive signatures on freeways. *Transportation Research Record: Journal of the Transportation Research Board*, 2000 (1717): 130-136.
- [82] 周泳涛, 鲍卫刚, 翟辉等. 公路钢桥疲劳设计荷载标准研究. 土木工程学报, 2010, (11): 79-85.
- [83] Law S S, Lu Z R. Time domain responses of a prestressed beam and prestress identification. *Journal of Sound & Vibration*, 2005, 288(4): 1011-1025.
- [84] Chan T H T, Ashebo D B. Theoretical study of moving force identification on continuous bridges. *Journal of Sound and Vibration*, 2006, 295(3-5): 870-883.
- [85] 韩万水, 李彦伟, 乔磊等. 基于车-桥耦合振动理论的移动荷载识别. 中国公路学报, 2013, 26(1): 74-86.
- [86] 林诗枫, 黄侨, 任远等. 基于南京长江三桥的车辆荷载模型. 东南大学学报(自然科学版), 2016, 46(2): 365-370.

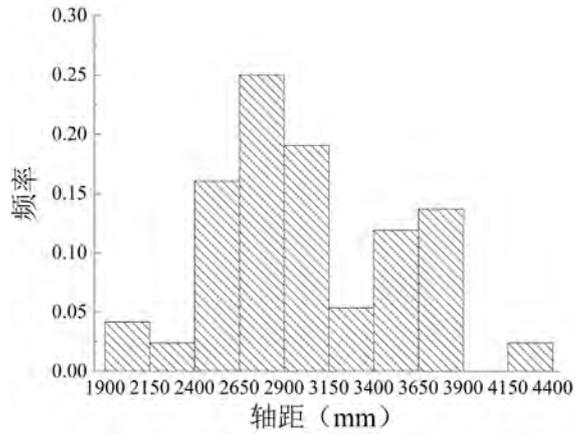
- [87] 汽车、挂车及汽车列车外廓尺寸、轴荷及质量限值, 2016.
- [88] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, 521(7553): 436.
- [89] Lawrence S, Giles C L, Tsoi A C, et al. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 1997, 8(1): 98-113.
- [90] Nebauer C. Evaluation of convolutional neural networks for visual recognition. *IEEE Transactions on Neural Networks*, 1998, 9(4): 685-696.
- [91] 余凯, 贾磊, 陈雨强等. 深度学习的昨天、今天和明天. *计算机研究与发展*, 2013, 50(9): 1799-1804.
- [92] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009: 248-255.
- [93] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks. In: *European conference on computer vision*. Springer, Cham, 2014: 818-833.
- [94] Taigman Y, Yang M, Ranzato M A, et al. Deepface: Closing the gap to human-level performance in face verification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014: 1701-1708.
- [95] Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10,000 classes. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014: 1891-1898.
- [96] Sun Y, Chen Y, Wang X, et al. Deep learning face representation by joint identification-verification. In: *Advances in neural information processing systems*. 2014: 1988-1996.
- [97] Liu S, Liu S, Cai W, et al. Early diagnosis of Alzheimer's disease with deep learning. In: *2014 IEEE 11th international symposium on biomedical imaging (ISBI)*. IEEE, 2014: 1015-1018.
- [98] 庄福振, 罗平, 何清等. 迁移学习研究进展. *软件学报*, 2015, 26(1): 26-39.
- [99] Pan S J, Yang Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 2010, 22(10): 1345-1359.
- [100] 李彦冬, 郝宗波, 雷航. 卷积神经网络研究综述. *计算机应用*, 2016, 36(9): 2508-2515.
- [101] Canny J F. A computation approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1986, 8(6): 670-700.
- [102] Zhan C, Duan X, Xu S, et al. An improved moving object detection algorithm based on frame difference and edge detection. In: *Fourth International Conference on Image and Graphics (ICIG 2007)*. IEEE, 2007: 519-523.

- [103] Piccardi M. Background subtraction techniques: a review. In: 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583). IEEE, 2004, 4: 3099-3104.
- [104] Kesrarat D, Patanavijit V. Experimental study efficiency of robust models of Lucas-Kanade optical flow algorithms in the present of Non-Gaussian Noise. In: Knowledge and Smart Technology (KST). IEEE, 2012: 43-48.
- [105] Gandhi T L. Image sequence analysis for object detection and segmentation[M]. 2000.
- [106] Dai J, Li Y, He K, et al. R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. 2016: 379-387.
- [107] Koller D, Daniilidis K, Nagel H H. Model-based object tracking in monocular image sequences of road traffic scenes. International Journal of Computer Vision, 1993, 10(3): 257-281.
- [108] Wachter S, Nagel H H. Tracking Persons in Monocular Image Sequences. Computer Vision & Image Understanding, 1997, 74(3): 174-192.
- [109] Wren C R. Pfunder : real-time tracking of the human body. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1997, 19(7): 780-785.
- [110] Huang X J, Zhou J M. Moving objects detection approach based on adaptive mixture Gaussian background model. Journal of Computer Applications, 2010, 30(1): 71-74.
- [111] 高展宏,徐文波. 基于 MATLAB 的图像处理案例教程. 北京: 清华大学出版社, 2011

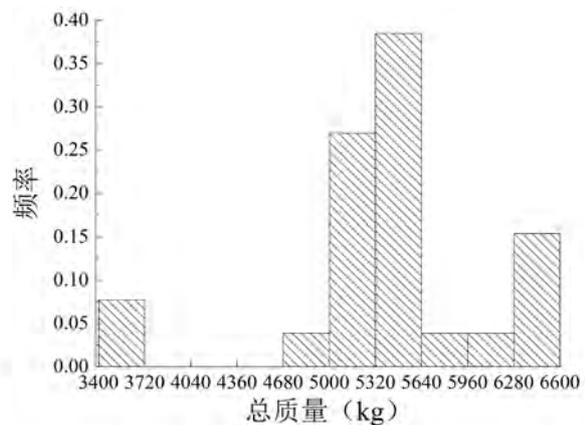
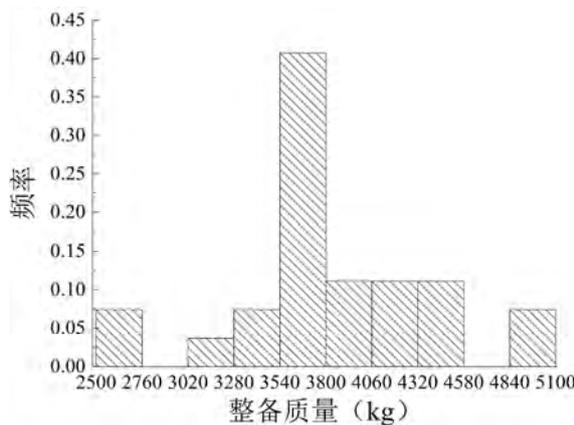
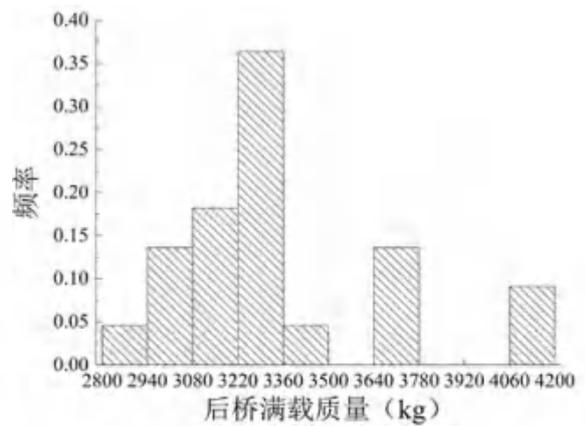
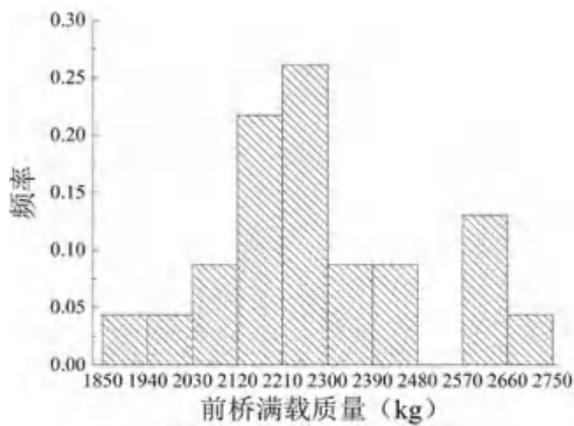
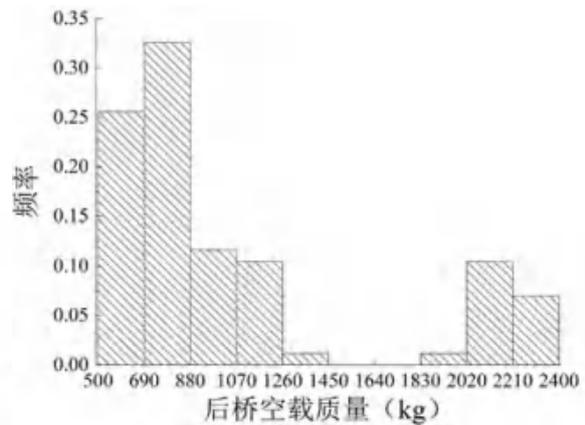
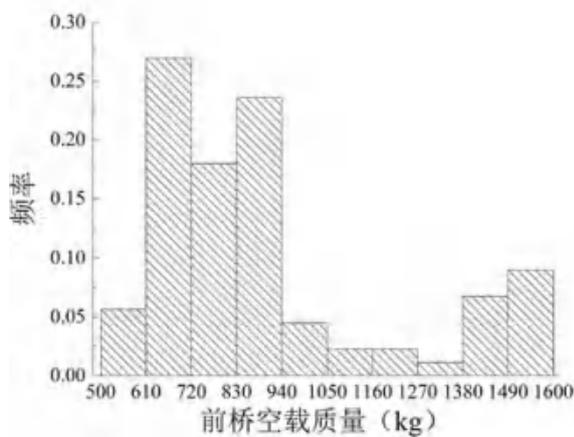
附录 A 车辆数据库信息统计图

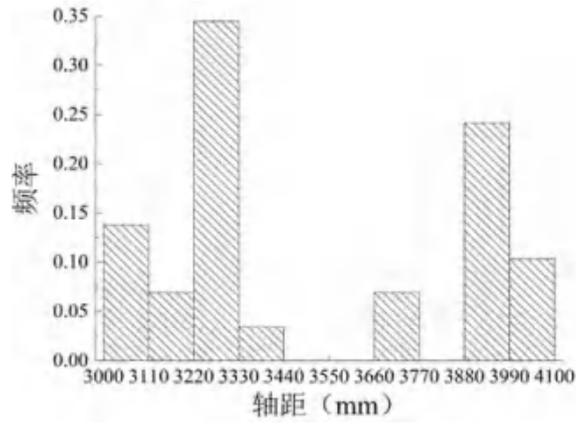
1、 面包车相关信息频率统计图



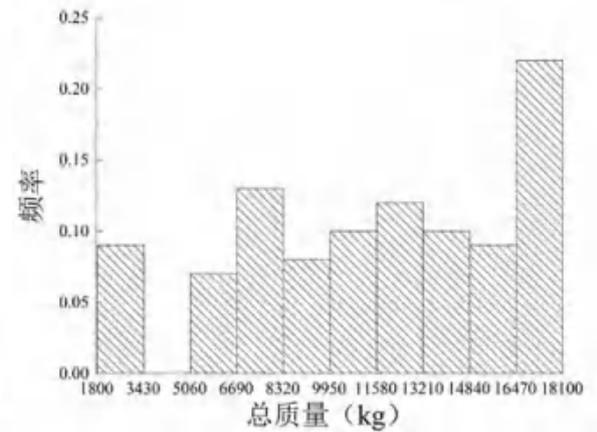
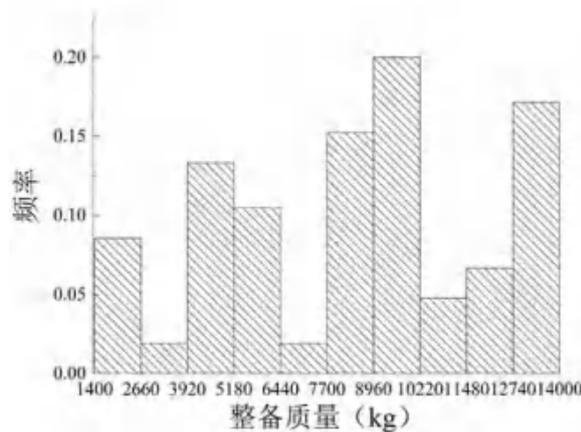
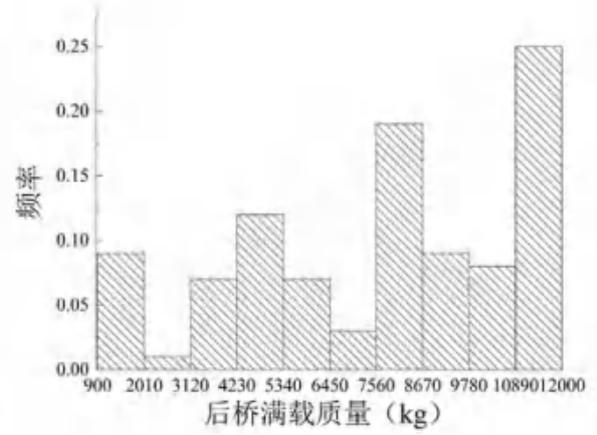
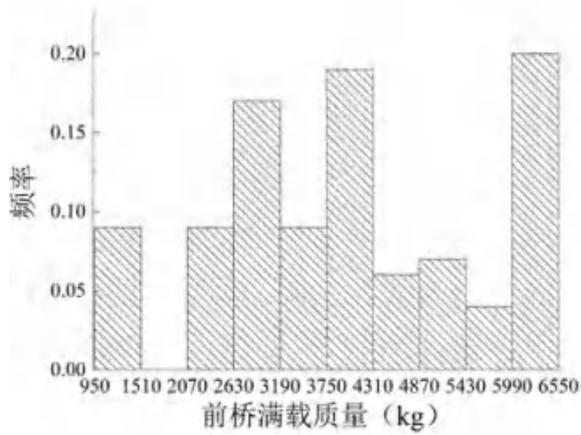
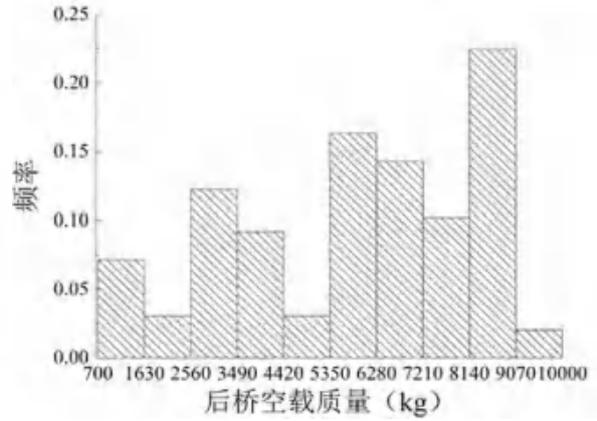
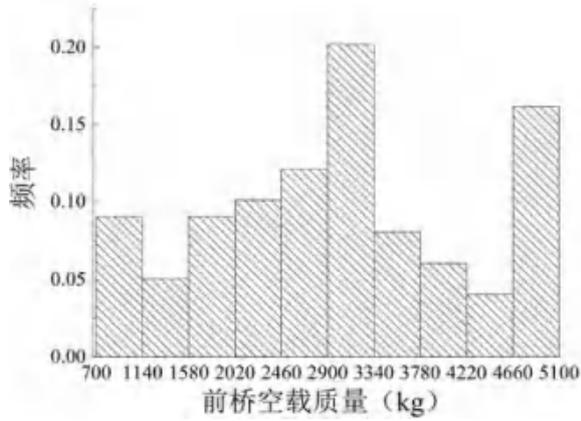


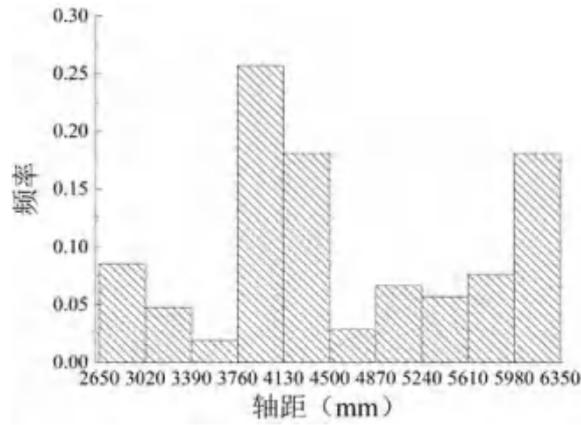
2、 中型客车相关信息频率统计图



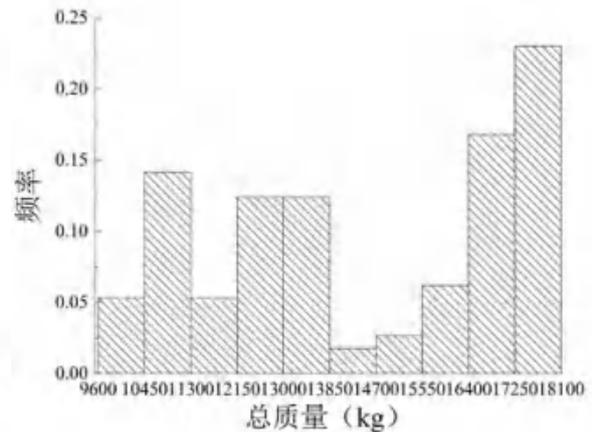
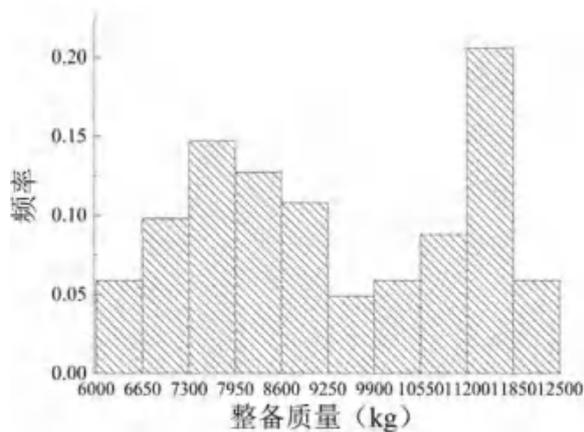
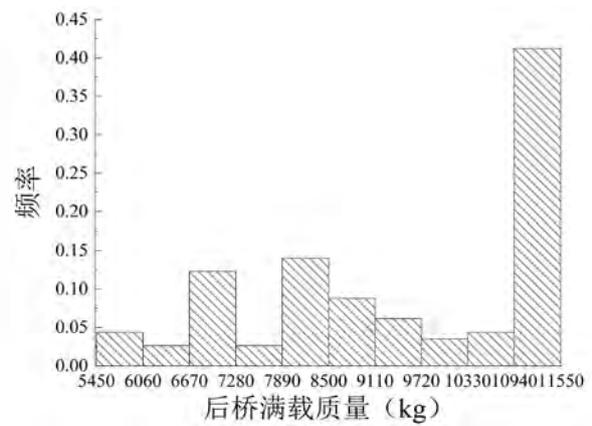
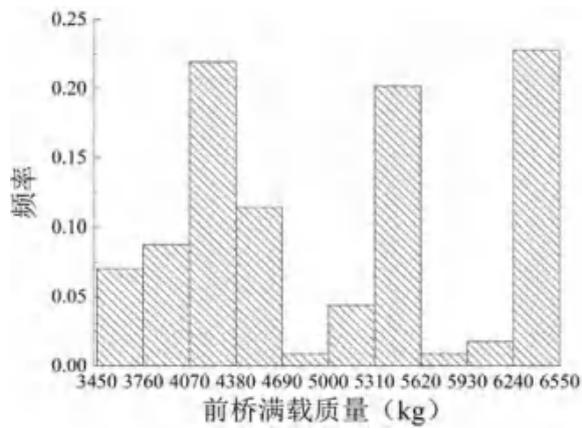
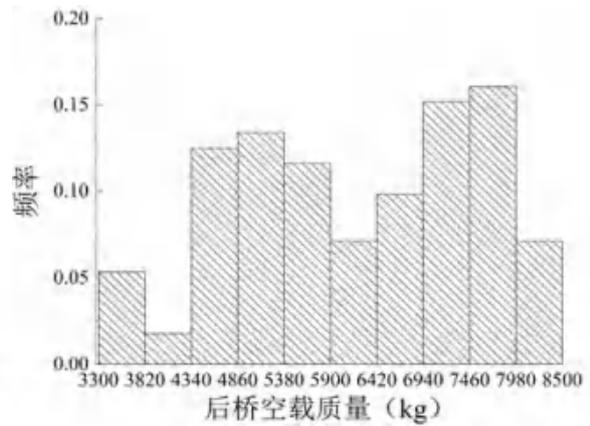
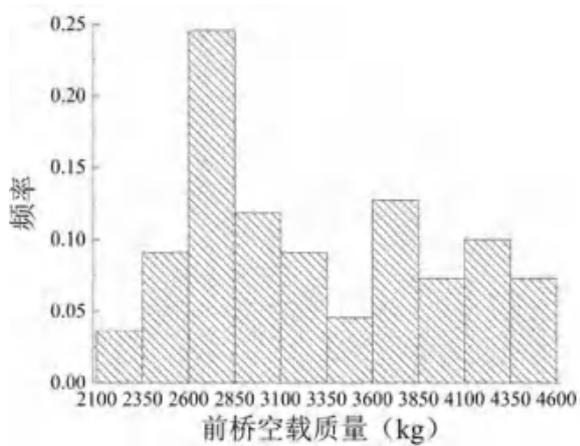


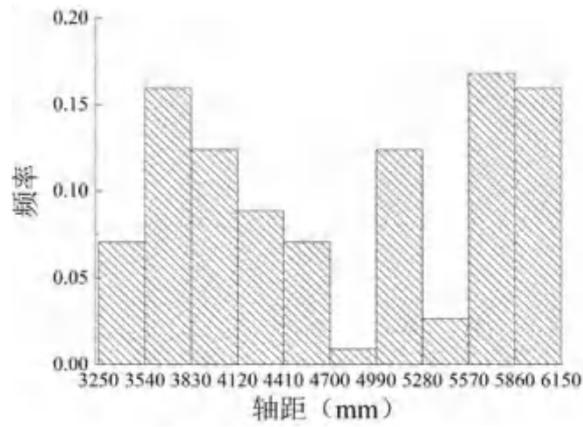
3、 大型客车相关信息频率统计图



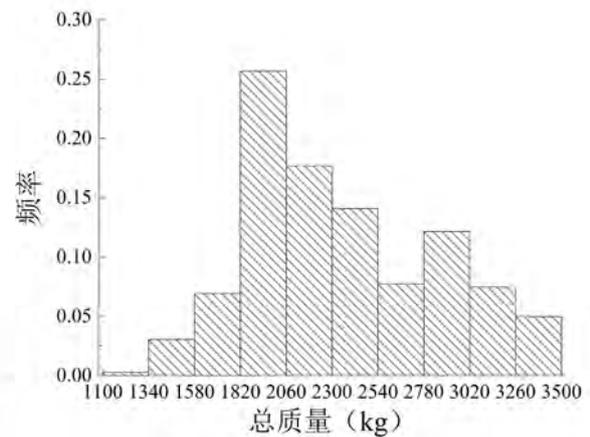
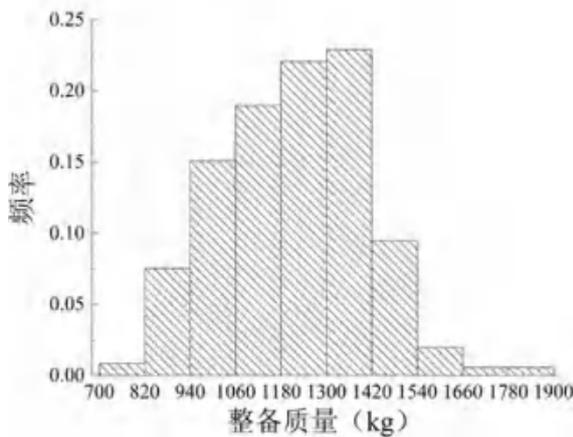
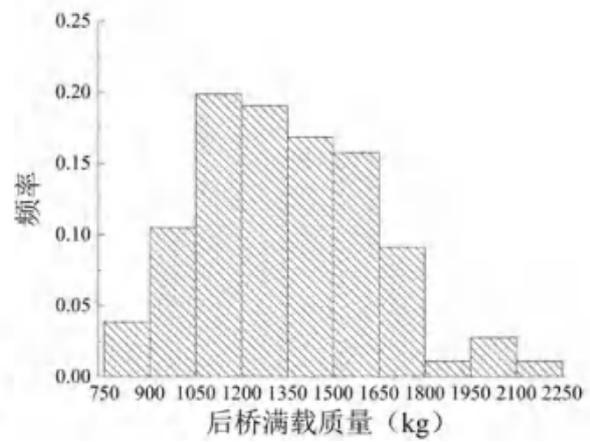
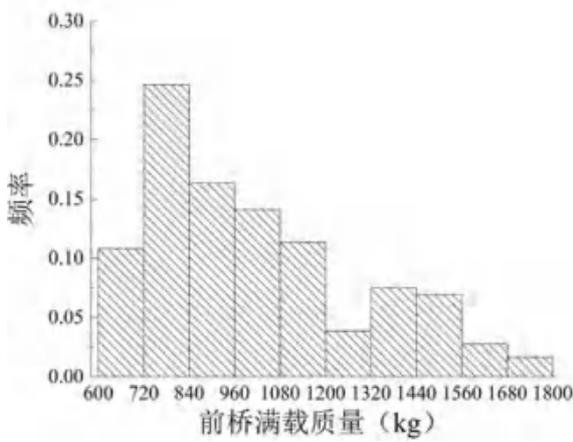
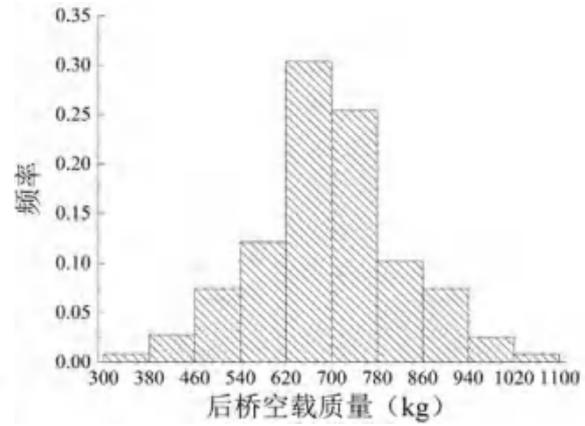
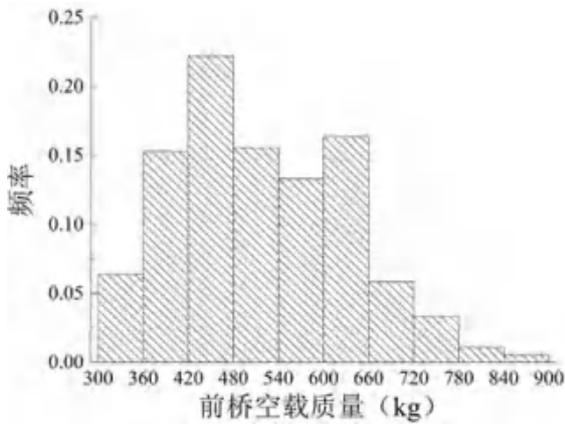


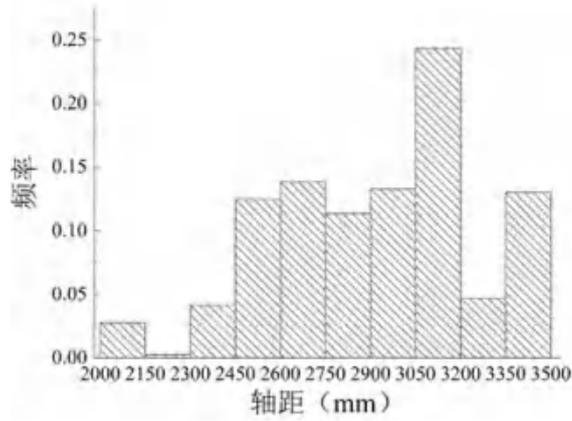
4、 公交车相关信息频率统计图



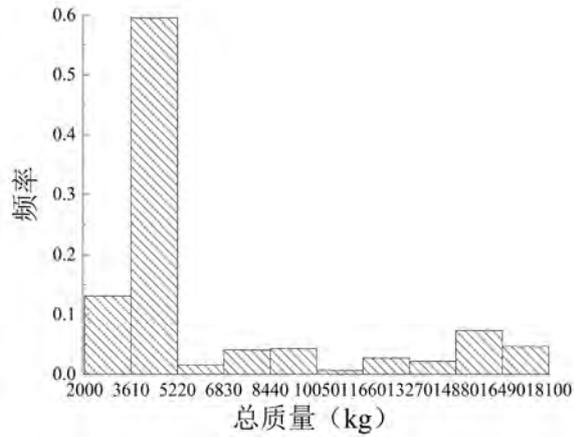
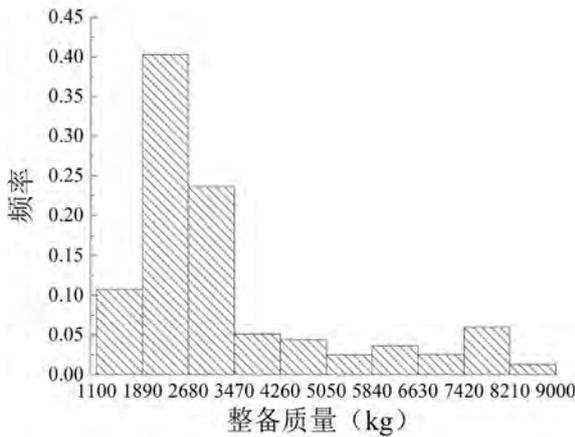
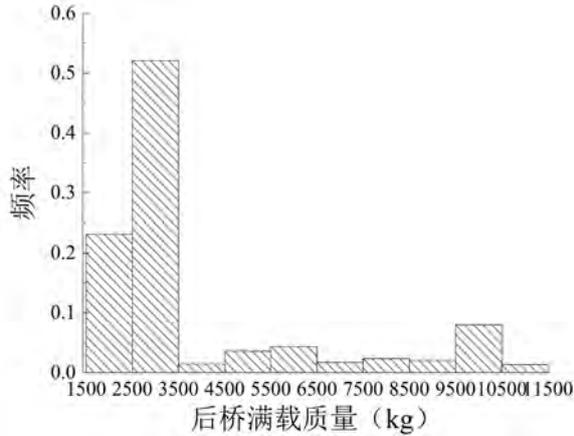
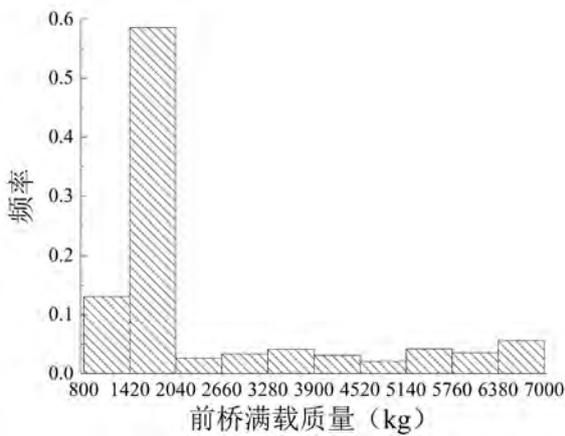
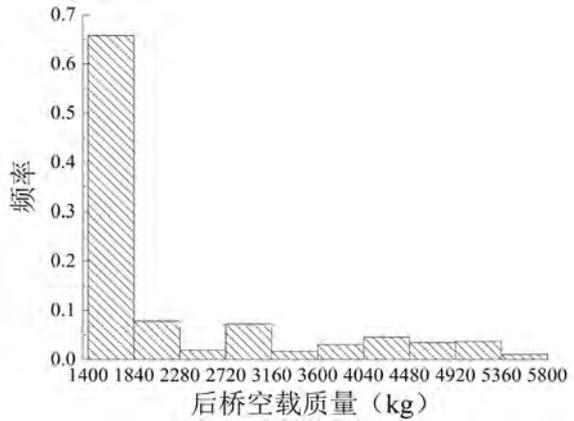
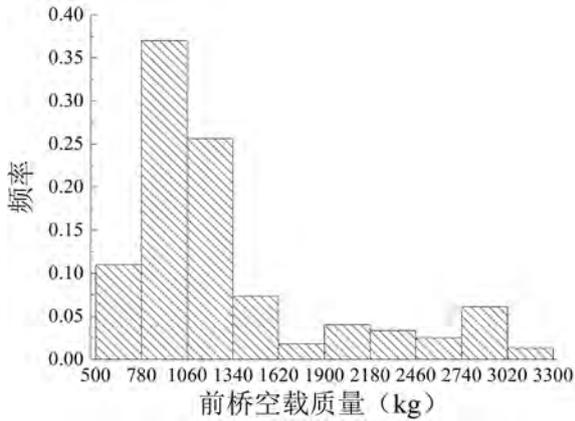


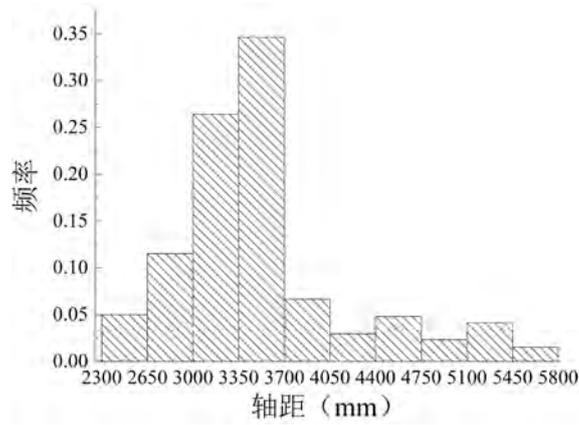
5、 2轴4轮货车相关信息频率统计图



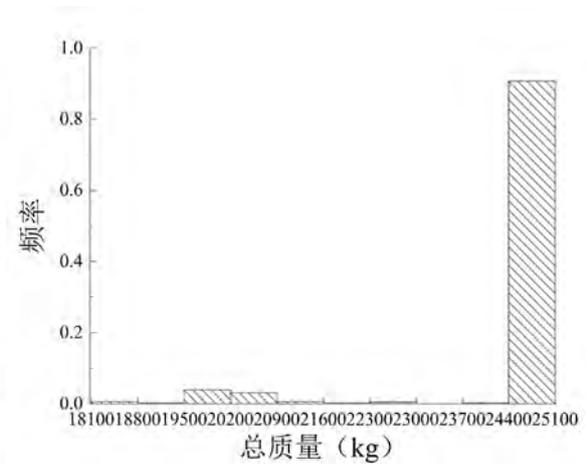
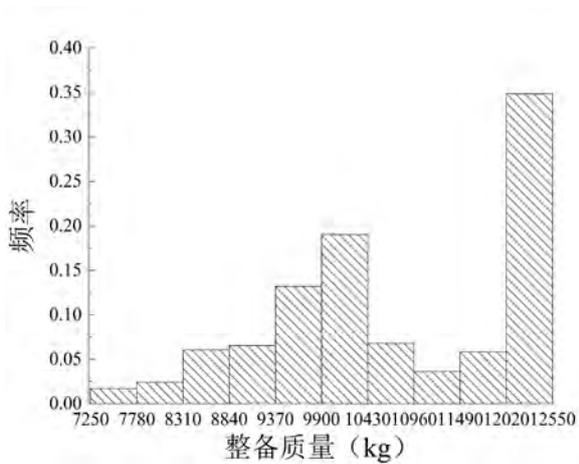
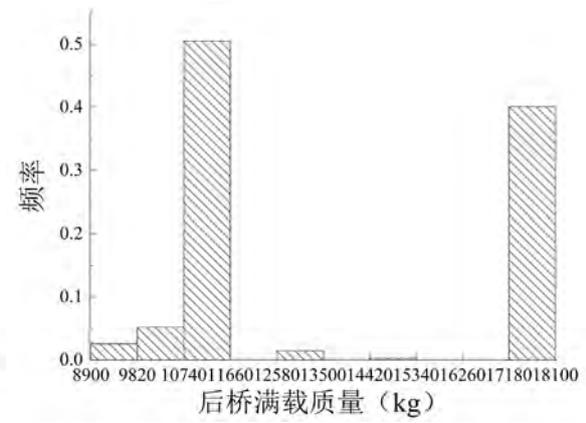
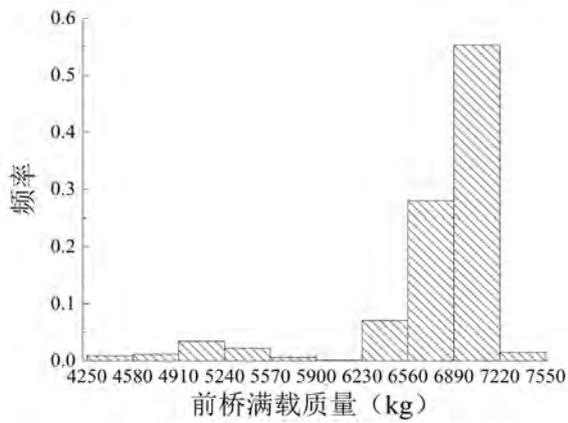
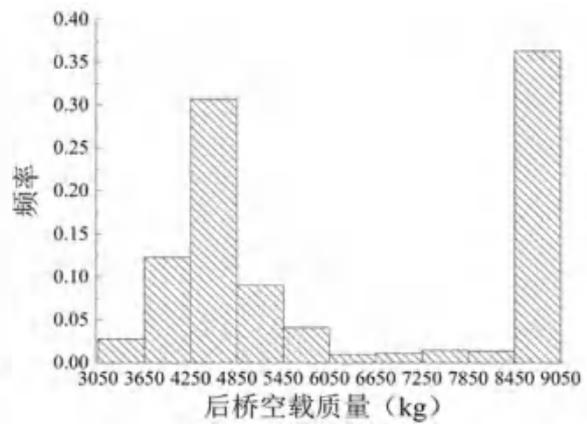
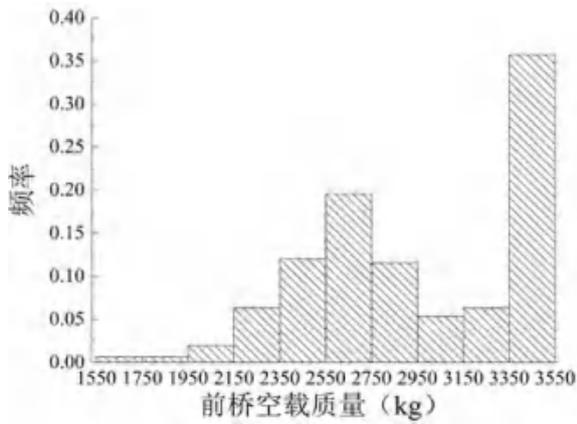


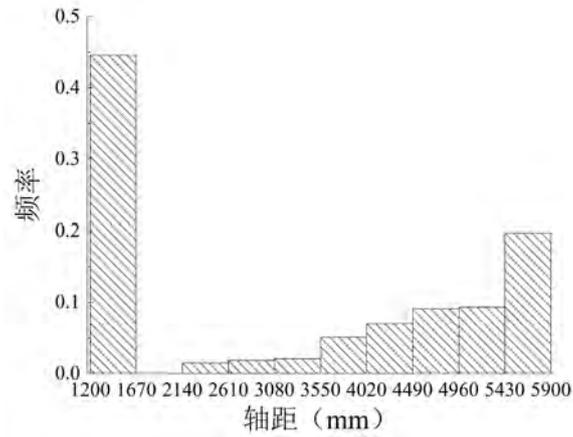
6、 2 轴 6 轮货车相关信息频率统计图



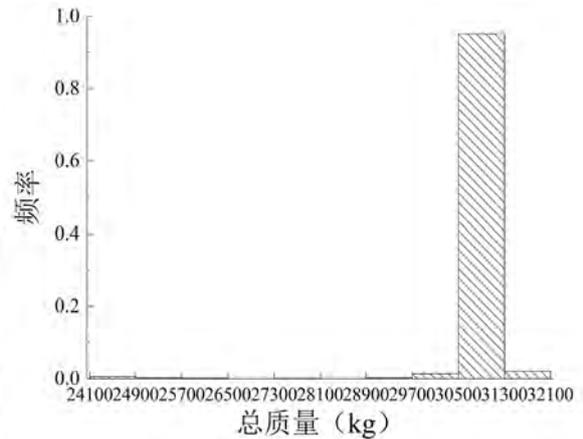
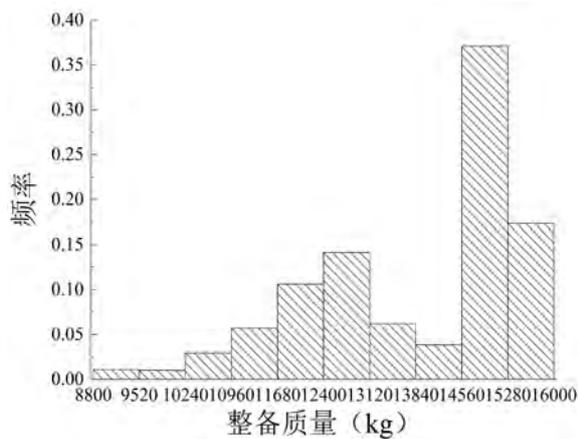
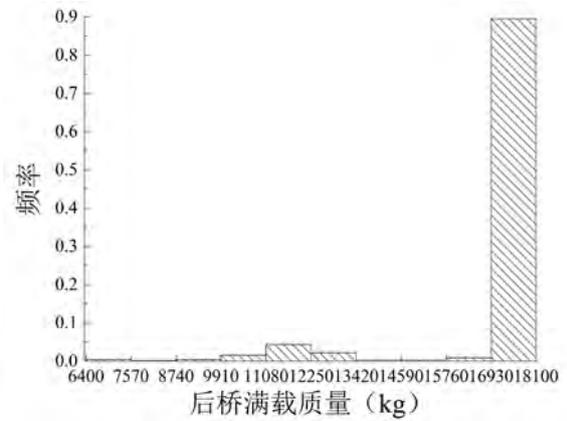
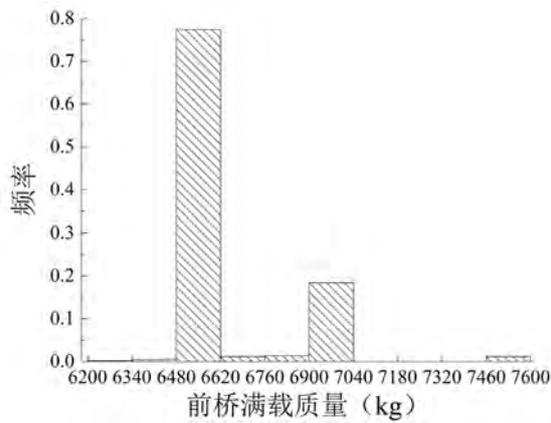
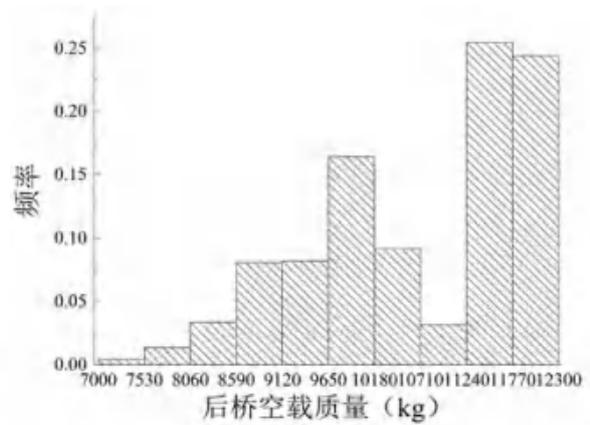
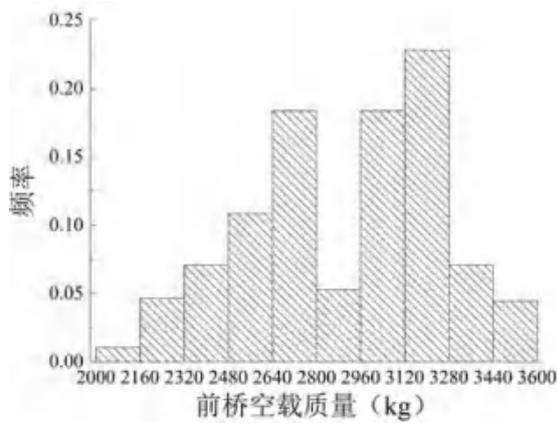


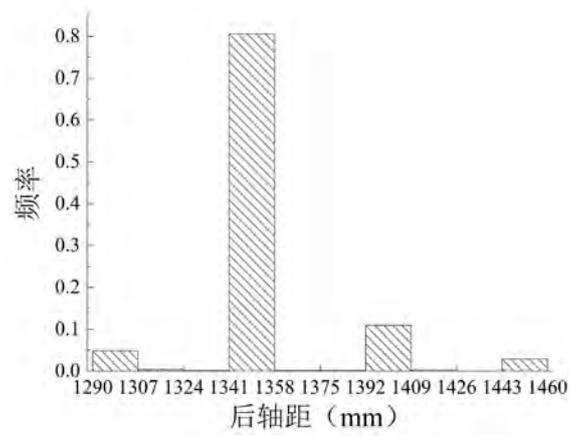
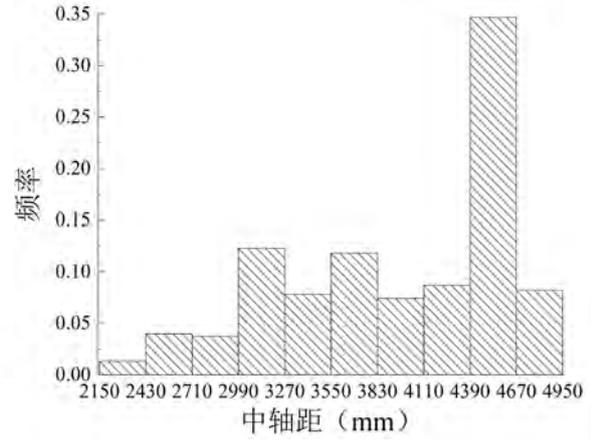
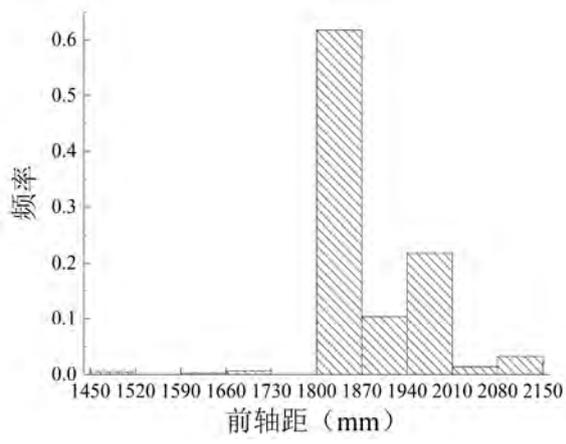
7、 3 轴货车相关信息频率统计图





8、 4 轴货车





致 谢

匆匆三年转眼即逝，充满了太多的艰辛与困苦，但更多的成长与幸福。回忆过去，已在这里度过了整整七年青春时光。三年的研究生生涯，既是一段学术的攀登，也是一次丰富的成长。而今毕业在即，心怀许多感慨与感激，唯有借着此次机会，将感激一一道来。

首先要感谢导师周云在学业上对我的指导和帮助，师恩之情，没齿难忘。在研究生的学习生涯中，从课题选择到论文撰写，周老师始终给予我细心的指导和不懈的支持。感谢师兄胡翔、陈太平、奚树杭，师姐周祎、刘蒙，同门彭涵钰、张琴、曾雅丽思、王振生，以及师弟师妹周赛、裴熠麟、程依婷、赵瑜、阳建波、胡建鑫、李自强、周易在三年之中对我的帮助与关心。感谢室友周亚萍对我生活的关心与帮助。

感谢我的父母，你们是我来时的路，也是我内心的归宿，感谢胡翔的陪伴、理解与支持。最后感谢与好友愉快相处的时光，在平淡的生活中为彼此增添了无穷的乐趣。感谢湖大的一草一木，于孤独的时光中为我带来春花秋月的好时节。

感谢各位专家、教授在百忙之中对本论文的审阅！

李紫玮

2019年3月30日于湖南大学