Software Engineering and Applications 软件工程与应用, 2023, 12(1), 102-112
Published Online February 2023 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2023.121011

Hans汉斯

基于深度学习融合全景拍摄技术进行车辆识别

王子昇，范国伦，江　河，李昕悦

北京信息科技大学计算机学院，北京

收稿日期：2023年1月21日；录用日期：2023年2月17日；发布日期：2023年2月24日

摘　要

现如今，VR技术迅速发展，社会各方面对于VR技术的实践层出不穷，本论文意从拍摄地全景视频出发，再加上较为成熟的识别和追踪算法，对于车辆进行一个便利和精准的追踪。全景视频被广泛地应用在交通和安全领域，通过实现摄像头的移动使被拍摄物体(车辆)始终居于画面中央，减少了不必要的干扰，可以尽量保证主体的完整，通过全景视频可以使相关的数据分析更加便利和精准。本系统在全景视频中检查运动目标的轨迹，运用深度学习的框架来提高车辆追踪的精度和速度。

关键词

全景视频，车辆识别，车辆追踪

Vehicle Recognition Based on Deep Learning Fusion Panoramic Shooting Technology

Zisheng Wang, Guolun Fan, He Jiang, Xinyue Li

School of computing, Beijing Information Science & Technology University, Beijing

Received: Jan. 21st, 2023; accepted: Feb. 17th, 2023; published: Feb. 24th, 2023

Abstract

Nowadays, with the rapid development of VR technology, the practice of VR technology in all aspects of society is endless, this paper intends to start from the panoramic video of the shooting location, coupled with a more mature identification and tracking algorithm, for a convenient and accurate tracking of vehicles. Panoramic video is widely used in the field of traffic and safety, by realizing the movement of the camera so that the object (vehicle) to be photographed is always in the center of the picture, reducing unnecessary interference, can try to ensure the integrity of the subject, through panoramic video can make the relevant data analysis more convenient and accurate. This system examines the trajectory of moving targets in a panoramic video, and uses a deep learning framework to improve the accuracy and speed of vehicle tracking.

文章引用: 王子昇, 范国伦, 江河, 李昕悦. 基于深度学习融合全景拍摄技术进行车辆识别[J]. 软件工程与应用, 2023, 12(1): 102-112. DOI: 10.12677/sea.2023.121011

Keywords

Panoramic Video, Vehicle Recognition, Vehicle Tracking

1. 前言

国内外的研究现状及研究意义

随着深度神经网络在计算机视觉和多媒体领域的发展，大众对车辆再识别技术的关注度逐渐提高，车辆再识别技术也成为深度学习领域的研究热点问题。

与以往的近似重复图像检索(NDIR)问题不同，车辆再识别任务的巨大挑战之一来自于交通监控系统存在拍摄视角变化、天气影响、光照变化等导致车牌像素模糊、车牌信息不完整的问题。而且不同摄像头位置的照度、视角和分辨率的差异，也会造成同一辆车出现不同的视角类内差异或不同车辆之间因同一车型而相似的结果[1]。针对这些问题，同时为了克服现实交通环境中出现道路拥塞的情况所导致的车辆高度密集出现，以及拍摄距离远、设备老化等导致的图像分辨率低等复杂环境因素的负面影响，我们选择采用全景视频拍摄作为切入点。全景视频具有更好的完整性、沉浸性和呈现性，能够实现更加精准、实时、个性化的车辆追踪和识别。因此我们提出了应用全景拍摄技术并融合多种深度学习算法的方法，应用于车辆追踪问题，形成与其他传统研究所不同的车辆追踪框架，实现了范围更大，时间更长，覆盖面更广等优势。

在车辆再识别方法的选择上，深度学习算法选择是为了规避传统方法中的诸多弊端，例如基于传感器的方法，需要安装大量的硬件设备且受客观环境影响大；基于 3D 建模的方法，时间复杂度高、实现效率和识别精度较低；基于手工设计特征的车辆再识别方法，性能容易受到限制且受客观环境影响大；基于度量学习的方法需要抽象地理解车辆图像在特征空间中的关系，并在神经网络的训练过程中运用一些技巧；使用时间－地理信息的再识别方法对数据集的要求十分严格，需要大量额外的时间戳和地点标注工作。综合上述各种方法，可以看出深度学习对于车辆图像具有强大的特征表达能力，能够较好地化解光照变化、视角变化等客观因素带来的困难和挑战，在完成训练后，可以在极短的时间内完成对目标车辆的再识别过程，通过合理地设计网络结构和损失函数，深度学习方法可以准确、快速地在海量图像中找出目标车辆，是可以较好地完成车辆再识别任务的基础框架[2]。

本文致力于从拍摄地全景视频出发，再加上较为成熟的识别和追踪算法，对于车辆进行便利和精准的追踪，提高车辆再识的应用范围以及使用操作性。同时本文将按如下三步进行实验开发：1) 将采集到的数据能够在现有的大量数据集中(如 VeRi 数据集)运行起来并查看训练结果(精度)；2) 将自己拍摄的全景视频转换为 VeRi 数据集进行训练并使对车辆的识别精度达到一定的标准；3) 设计网页快捷使用本项目进行对车辆的识别。

2. 项目概述

2.1. 项目流程

本项目采用 Insta360 全景相机进行全景拍摄，采用 mmdetection 算法将拍摄的视频转化为 VeRi 数据

集，使用 reid baseline 对数据集进行检测识别，最后采用 mmtracking 对已识别的车辆进行追踪，具体项目流程图如图 1 所示。

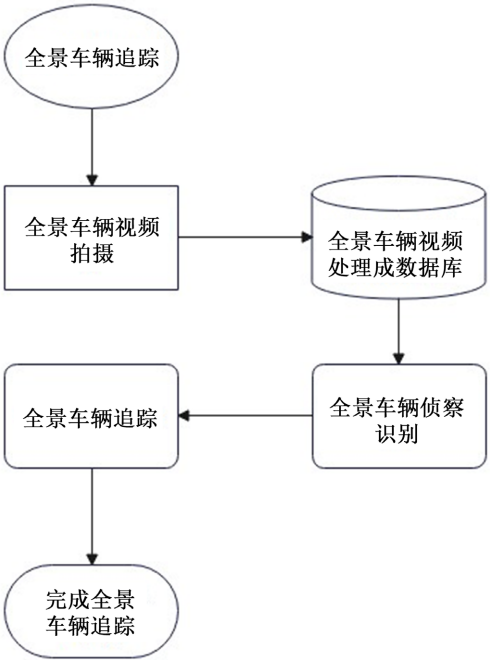

Figure 1. Project flow chart
图 1. 项目流程图

2.2. 相关设备介绍

Insta360 全景相机是 Insta360 研发的以全景技术为基点的相机，其具体介绍如图 2 所示，拍摄效果图如图 3 所示。

单镜头广角拍摄	支持
穿越小飞机模式	支持
270°极致广角	支持
裸机防水	支持：10米
画面矫正	支持
续航时间	5.7K@30fps. 续航80分钟
电池容量	1630mAh
收音	四麦全景声音
拍照	普通拍照 HDR拍照 Burst九连拍 间隔拍照 超级夜景拍照 PureShot纯净摄影 270°极致广角
录像	普通摄影 HDR视频 Timelapse延时摄影 子弹时间摄影 Timeshif移动延时摄影 超广角防抖摄影

Figure 2. Introduction to Insta360 panoramic camera
图 2. Insta360 全景相机介绍

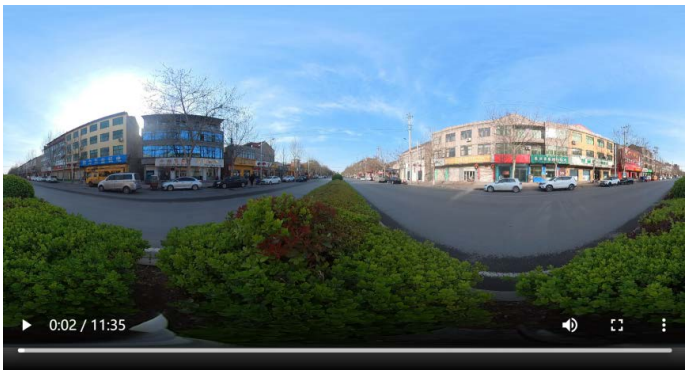

Figure 3. Insta360 panoramic camera video
图 3. Insta360 全景相机视频图

3. 系统运行环境

硬件环境：推荐 Intel Core 5，8 GB 内存，100 GB 硬盘空间。

开发语言：JavaScript、html。

开发工具：Flask、PyCharm。

运行环境：Mac 10.14.6、Win10 及以上版本。

系统的软硬件环境配置如表 1。

Table 1. Software and hardware environment of vehicle re-identification software in panoramic video
表 1. 全景视频中车辆再识别软件软硬件环境

	软硬件配置(或版本)	用途说明
硬件环境	服务器一台	系统运行
	外网端口开启，连接 Internet	连接外部设施(合作单位)
软件环境	操作系统：Win 或 Mac	运行应用程序
	开发环境：Flask、PyCharm	开发应用程序
	应用软件：无特殊要求	

4. 主要技术(三种主要应用框架)

4.1. MMDetection

MMDetection 是一个基于 PyTorch 的目标检测开源工具箱。它是 OpenMMLab 项目的一部分。它将检测框架解耦成不同的模块组件，通过组合不同的模块组件，可以便捷地构建自定义的检测模。同时它支持了众多主流的和最新的检测算法，例如 Faster R-CNN，Mask R-CNN，RetinaNet 等[3]。MMDetection 检测框架包括 MMDetection 和 mmcv，两者是不可分割的。起初 mmcv 是很小的一个存在，但是后来 mmdet 中很多的组件都被转移到了 mmcv 中去，其中包括 cnn 的构建相关组件和所有的 CV 相关 operators，比如 ROIAlign，ROIPooling，DCN 等。这些 ops 通常无法直接通过 Tensor 的操作来实现，所以都用了 CUDA 进行了实现。当把这些 CUDA 代码迁移到 mmcv 后，编译 mmdet 就方便了许多，只要安装了满足需求的 mmcv。mmcv 不仅是 mmdet 的基础，为检测框架服务，它同时也为其他 OpenMMLab 视觉框架服务。

配置过程如下：

在 GPU 服务器上选择合适的镜像创建容器，再通过以下代码安装 MMtracking：

conda create -n open-mmlab python=3.7 -y

conda activate open-mmlab

#创建 conda 虚拟环境并激活

conda install pytorch==1.6.0 torchvision==0.7.0 cudatoolkit=10.1 -c pytorch -y

#按照 PyTorch 官网安装 PyTorch 和 torchvision

pip install git + https://github.com/votchallenge/toolkit.git (可选)

#安装最新版本的 mmcv

pip install mmcv-full -f https://download.openmmlab.com/mmcv/dist/cu101/torch1.6.0/index.html

#安装 mmdetection

pip install mmdet

1) 安装完成后进行采集数据

安装完成后进行数据采集，数据采集截图如图 4 所示。

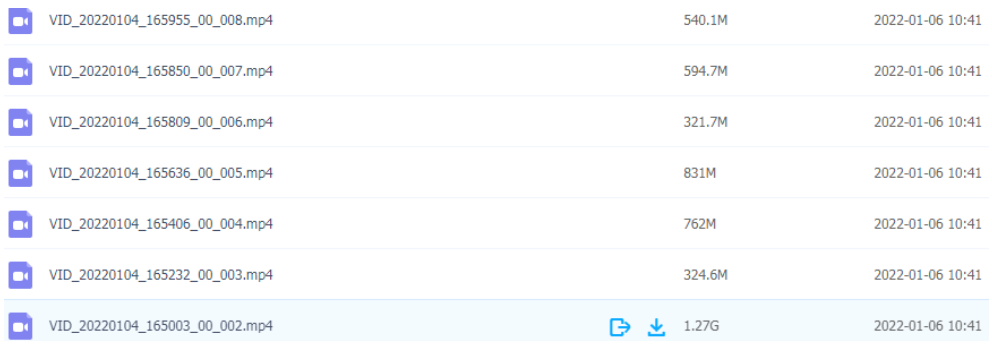

Figure 4. Capture screenshots of data
图 4. 采集数据截图

2) 数据处理

将数据进行处理，转化为需要的图片格式。其中每一个文件夹中都包含着一辆车的几张图片，如图 5、图 6 所示。这样一来方便进行模型训练并载入权重。

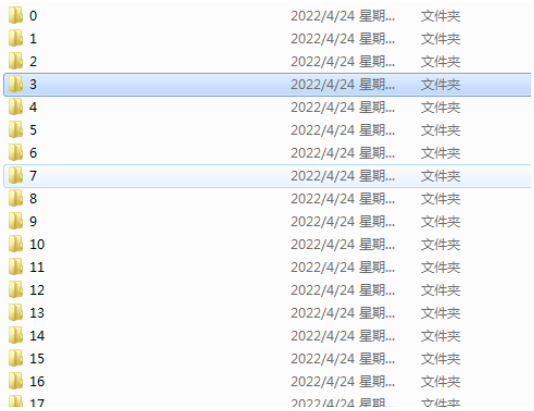

Figure 5. Screenshot of data in image format
图 5. 图片格式数据截图

Figure 6. Screenshot of image format data
图 6. 图片格式数据示意截图

3) 数据标注

采用 MMDetection 对数据集进行识别和标注，训练结果如图 7 所示，对整个场景中的车辆进行标注，以便于后续运用 mmtracking 算法进行筛选进而实现对单一车辆的追踪和监控。

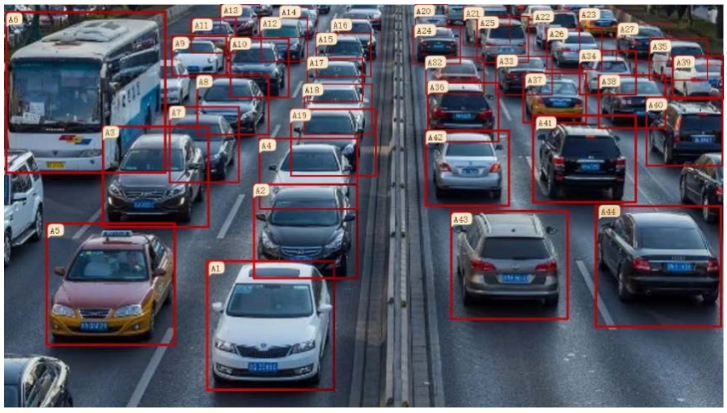

Figure 7. Graph of training results
图 7. 训练结果图

4.2. Reid Baseline

Reid Baseline 是一个基于 pytorchQ 实现的，小巧、友好并且强大的技术。它的性能媲美当前最好的公开方法，支持 fp16 精度用 2 GB 显存进行训练(小巧)，并且提供了一个 8 分钟快速教程入门 reid (新手友好)。

4.3. MMTracking

MMTracking 是一款基于 PyTorch 的视频目标感知开源工具箱，是 OpenMMLab 项目的一部分。同时，MMTracking 是首个开源一体化视频目标感知工具箱，同时支持视频目标检测，多目标跟踪，单目标跟踪和视频个例分割等多种任务和算法，它将统一的视频目标感知框架解耦成不同的模块组件，通过组合不同模块组件，用户可以便捷地构建自定义视频目标感知模型，具有简洁、高效、强大的特性[4]。利用 MMTracking 进行视频目标检测和单目标跟踪两个目的，将收集的视频和图像处理成符合格式的数据集，再通过 MMTracking 对车辆进行追踪和检测，从而实现车辆追踪。MMTracking 与其他 OpenMMLab 平台充分交互。MMTracking 充分复用 MMDetection 中的已有模块，只需要修改配置文件就可以使用任何检测器，并且 MMTracking 所有操作都在 GPU 上运行。相比其他开源库的实现，MMTracking 的训练和推理更加高效，它复现了 SOTA 性能的模型。受益于 MMDetection 的持续推进，部分实现精度超出官方版本。

配置过程如下：

在 GPU 服务器上选择合适的镜像创建容器，再通过以下代码安装 MMTracking：

conda create -n open-mmlab python=3.7 -y
conda activate open-mmlab
#创建 conda 虚拟环境并激活
conda install pytorch==1.6.0 torchvision==0.7.0 cudatoolkit=10.1 -c pytorch -y
#按照 PyTorch 官网安装 PyTorch 和 torchvision
pip install git + https://github.com/votchallenge/toolkit.git (可选)
#安装最新版本的 mmcv
pip install mmcv-full -f https://download.openmmlab.com/mmcv/dist/cu101/torch1.6.0/index.html
#安装 mmdetection
pip install mmdet
#安装 mmtracking
git clone https://github.com/open-mmlab/mmtracking.git
cd mmtracking
#将 mmtracking 仓库克隆到本地
pip install -r requirements/build.txt
pip install -v -e.
#首先安装依赖，再安装 mmtracking
pip install git + https://github.com/JonathonLuiten/TrackEval.git
#为 MOTChallenge 评估
pip install git + https://github.com/lvis-dataset/lvis-api.git
#为 LVIS 评估
pip install git + https://github.com/TAO-Dataset/tao.git
#为 TAO 评估
最后对使用标注的车辆的轨迹进行全方位追踪。单一车辆追踪的效果图如图 8 所示。

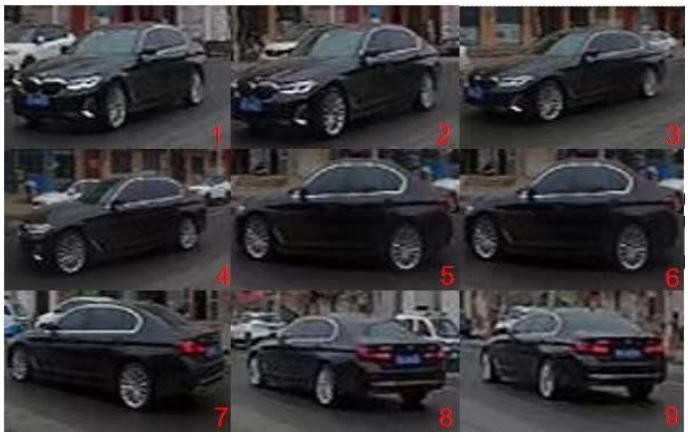

Figure 8. Single vehicle tracking
图 8. 单一车辆追踪

搭建平台：
制作网页界面，使得用户可以通过网页进行车辆识别，网页界面如图 9 所示。

Figure 9. Vehicle identification web page
图 9. 车辆识别网页

这样一来通过全景相机拍摄的视频和图片，经过简单的处理后交给 MMDetection 框架进行训练，可以很快的将我们需要观测的车辆划分出来，再通过 MMTracking 算法的单一目标追踪功能，可以对某一特定车辆实施追踪并记录其轨迹，进而解决了当今监控设备只能监测大范围所有车辆，而无法筛选进行大范围单一车辆的检测问题。

数据集准备：

对于视频目标检测任务的训练和测试，只需要 ILSVRC 数据集。对于多目标跟踪任务的训练和测试，需要 MOT Challenge 中的任意一个数据集(比如 MOT17, TAO 和 DanceTrack)，CrowdHuman 和 LVIS 可以作为补充数据。对于单目标跟踪任务的训练和测试，需要 MSCOCO，ILSVRC，LaSOT，UAV123，TrackingNet，OTB100 和 GOT10k 数据集。对于视频实例分割任务的训练和测试，只需要 YouTube-VIS 中的任意一个数据集(比如 YouTube-VIS 2019)。

还需要转换格式，具体转换代码如下：

```
#ImageNet DET
python ./tools/convert_datasets/ilsvrc/imagenet2coco_det.py -i ./data/ILSVRC -o ./data/ILSVRC/annotations
#ImageNet VID
python ./tools/convert_datasets/ilsvrc/imagenet2coco_vid.py -i ./data/ILSVRC -o ./data/ILSVRC/annotations
# MOT17# MOT Challenge 中其余数据集的处理与 MOT17 相同
python ./tools/convert_datasets/mot/mot2coco.py -i ./data/MOT17/ -o ./data/MOT17/annotations
--split-train --convert-det
python ./tools/convert_datasets/mot/mot2reid.py -i ./data/MOT17/ -o ./data/MOT17/reid --val-split 0.2
--vis-threshold 0.3
# DanceTrack
python ./tools/convert_datasets/dancetrack/dancetrack2coco.py
-i ./data/DanceTrack ./data/DanceTrack/annotations
# CrowdHuman
python ./tools/convert_datasets/mot/crowdhuman2coco.py -i ./data/crowdhuman
-o ./data/crowdhuman/annotations
#LVIS#合并 LVIS 和 COCO 的标注来训练 QDTrack
python ./tools/convert_datasets/tao/merge_coco_with_lvis.py
--lvis ./data/lvis/annotations/lvis_v0.5_train.json --coco ./data/coco/annotations/instances_train2017.json
```

--mapping ./data/lvis/annotations/coco_to_lvis_synset.json
--output-json ./data/lvis/annotations/lvisv0.5+coco_train.json

#TAO#为 QDTrack 生成过滤后的 json 文件

python ./tools/convert_datasets/tao/tao2coco.py -i ./data/tao/annotations --filter-classes

#LaSOT

python ./tools/convert_datasets/lasot/gen_lasot_infos.py -i ./data/lasot/LaSOTBenchmark
-o ./data/lasot/annotations

#UAV123#下载标注#由于 UAV123 数据集的所有视频的标注信息不具有统一性，我们仅需下载提前生成的数据信息文件即可。

wget https://download.openmmlab.com/mmtracking/data/uav123_infos.txt -P data/uav123/annotations

TrackingNet#解压目录'data/trackingnet/'下的所有'*.zip'文件

bash ./tools/convert_datasets/trackingnet/unzip_trackingnet.sh ./data/trackingnet#生成标注

python ./tools/convert_datasets/trackingnet/gen_trackingnet_infos.py -i ./data/trackingnet
-o ./data/trackingnet/annotations

OTB100#解压目录'data/otb100/zips'下的所有'*.zip'文件

bash ./tools/convert_datasets/otb100/unzip_otb100.sh ./data/otb100#下载标注#由于 OTB100 数据集的所有视频的标注信息不具有统一性，我们仅需下载提前生成的数据信息文件即可。

wget https://download.openmmlab.com/mmtracking/data/otb100_infos.txt -P data/otb100/annotations

#GOT10k#解压'data/got10k/full_data/test_data.zip'，'data/got10k/full_data/val_data.zip'和目录'data/got10k/full_data/train_data/'下的所有'*.zip'文件

bash ./tools/convert_datasets/got10k/unzip_got10k.sh ./data/got10k# 生成标注

python ./tools/convert_datasets/got10k/gen_got10k_infos.py -i ./data/got10k -o ./data/got10k/annotations

VOT2018

python ./tools/convert_datasets/vot/gen_vot_infos.py -i ./data/vot2018 -o ./data/vot2018/annotations
--dataset_type vot2018

YouTube-VIS 2019

python ./tools/convert_datasets/youtubevis/youtubevis2coco.py -i ./data/youtube_vis_2019
-o ./data/youtube_vis_2019/annotations --version 2019

YouTube-VIS 2021

python ./tools/convert_datasets/youtubevis/youtubevis2coco.py -i ./data/youtube_vis_2021
-o ./data/youtube_vis_2021/annotations --version 2021

运行现有的数据集和模型：

1) 使用 VID 模型进行推理，代码如下：

```
python demo/demo_vid.py \
    ${CONFIG_FILE}\
    --input ${INPUT} \
    --checkpoint ${CHECKPOINT_FILE} \
    [--output ${OUTPUT}] \
    [--device ${DEVICE}] \
    [--show]
```

2) 使用 MOT/VIS 进行推理，代码如下：

```
python demo/demo_mot_vis.py \
    ${CONFIG_FILE} \
    --input ${INPUT} \
    [--output ${OUTPUT}] \
    [--checkpoint ${CHECKPOINT_FILE}] \
    [--score-thr ${SCORE_THR} \
    [--device ${DEVICE}] \
    [--backend ${BACKEND}] \
    [--show]
```

3) 使用 SOT 进行推理，代码如下：

```
python demo/demo_sot.py \
    ${CONFIG_FILE}\
    --input ${INPUT} \
    --checkpoint ${CHECKPOINT_FILE} \
    [--output ${OUTPUT}] \
    [--device ${DEVICE}] \
    [--show]
```

5. 采用的研究方案和解决的关键技术问题

1) 中小型数据集适用于利用多维信息的车辆重识别方法，结合附加标注信息或采用注意力机制，使网络能够更加准确地关注到车身所携带的识别特征，从而有效地简化网络的学习过程。不管是通过生成对抗模型来推断其他透视特征，还是使用预训练模型和聚类算法来获取透视信息，深度学习模型都可以通过训练来提高车辆身份的识别和灵敏度。

2) 目前车辆重识别领域的研究热点是基于度量学习的方法。其训练时间短，解释性强，一般采用三元损失函数和交叉熵损失函数。尽管其再识别性能较好，但有必要抽象地理解特征空间中车辆图像的关系，并在神经网络的训练过程中使用一些技能。

3) 我们通过强制卷积神经网络从域共享特征和特定领域特征重建来指导 CNN 学习域不变特征。通过分离不同域共享的信息和特定于每个域的信息，CNN 可以解开跨域共享的常识，以提高其再识别性能来减轻单个数据集的局限性。

4) 采用群体敏感三元嵌入方法来解决视角变化引起的类内差异、类间相似性问题。在训练中，将具有相同 ID 的车辆的图像划分为一个类，并找到类中心点：通过 K-mens 算法将图像聚集成数组，并找到组中心点。通过 inta-class 方差损失函数，将 ID 相同的车辆图像移动到特征空间中的类中心点，将同一组的图像移动到组内的组中心点，同时保持类间、组间一定的距离。完成移动后，使用 K 均值重新分组并重复上述步骤，直到损失函数最终收敛。

5) 解决问题的关键是挖掘车辆的识别特征。利用车身的特殊外观和数据集提供的属性标签来提高神经网络的灵敏度，比如车身贴图、保险杠、备胎、内饰或划痕都是非常明显的识别外观信息，而利用多维信息来监督神经网络的训练过程，也可以达到提高网络识别能力的目的。

6) 使用双分支自适应注意力网络(AAVER)进行车辆重新识别。网络的结构有两个分支，一个是用于捕捉车辆的全局外观特征的全局外观学习网络，；另一个分支包括车辆关键点检测和车辆视点估计下的

局部外观学习网络，通过学习将注意力集中在最容易识别的关键点上，从而捕捉到车辆最容易识别的区域以获取到有效的局部特征。通过全局和局部的融合，车辆再识别任务的准确度也可以得到很大的提升。

6. 结语

车辆追踪作为智能交通的重要组成部分，目前已经有很多较为成熟的技术研究。然而随着视频信息技术的不断发展，视频监控技术相较于其他技术更具直观性，含视觉信息、覆盖面广等特点。基于传统拍摄方法仍有一定的局限性，我组提出了全景视频拍摄作为切入点。全景视频具有更好的完整性、沉浸性和呈现性，能够实现更加精准、实时、个性化的车辆追踪和识别。因此我组提出了应用全景拍摄技术并融合多种深度学习算法的方法，应用于车辆追踪问题，形成与其他传统研究所不同的车辆追踪框架，实现了范围更大，时间更长，覆盖面更广等优势。

参考文献

[1] 刘凯, 李浥东, 林伟鹏. 车辆再识别技术综述[J]. 智能科学与技术学报, 2020, 2(1): 10-25.

[2] Zhu, X.Y., Luo, Z.B., Fu, P. and Ji, X. (2020) VOC-ReID: Vehicle Re-Identification Based on Vehicle-Orientation-Camera. 2020 *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (*CVPRW*), Seattle, WA, 14-19 June 2020, 2566-2573. https://doi.org/10.1109/CVPRW50498.2020.00309

[3] Song, W.F. and Li, S. (2020) Cross-View Contextual Relation Transferred Network for Unsupervised Vehicle Tracking in Drone Videos. 2020 *IEEE Winter Conference on Applications of Computer Vision* (*WACV*), Snowmass, CO, 1-5 March 2020, 1696-1705. https://doi.org/10.1109/WACV45572.2020.9093382

[4] Yao, Y. and Zheng, L. (2020) Simulating Content Consistent Vehicle Datasets with Attribute Descent. In: Vedaldi, A., Bischof, H., Brox, T. and Frahm, JM., Eds., *Computer Vision-ECCV* 2020. *ECCV* 2020. *Lecture Notes in Computer Science*, Vol. 12351, Springer, Cham. https://doi.org/10.1007/978-3-030-58539-6_46