Novel Traffic Sensing Using Multi-Camera Car Tracking and Re-Identification (MCCTRI)

Hao Yang

A thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Transportation Engineering

University of Washington

2020

Reading Committee:

Yinhai Wang
Xuegang (Jeff) Ban
Edward McCormack

Program Authorized to Offer Degree:
Department of Civil and Environmental Engineering

Abstract

Novel Traffic Sensing Using Multi-Camera Car Tracking and Re-Identification (MCCTRI)

Hao Yang

Chair of the Supervisory Committee:

Yinhai Wang

Department of Civil and Environmental Engineering

Traffic sensing devices are the eyes of the Intelligent Transportation Systems (ITS) nowadays. Among all the traffic sensors, the surveillance camera system is one of the most widely deployed system due to the easy installation, valuable data, and the intuitive information format. However, it’s a great pity that these cameras collect data isolated. One camera can only monitor a fixed of view and there is no bridge to share the monitoring information with each other. Tremendous labor work is necessary if the traffic managers try to find the same target in different cameras. Recently, the development of computer vision technology brings light to traffic information extraction based on the multi-camera scenario. Different from the previous single-camera based traffic information estimation, the multi-camera work is much more challenging. Since in the real-world scenarios, different camera views, orientations and lighting conditions make the video features in a huge difference. Moreover, the more rigorous thing is that only the top-one candidate can be used in the traffic information estimation procedure. Thus, how to link each single camera into a multi-camera system and estimate the traffic information from the whole surveillance system becomes the main problem in the research.

To address the challenges, four kinds of information are designed to capture and integrate, including vision information, vehicle attributes information, road network graph information and spatial-temporal information. These four kinds of information are summarized and decomposed into four levels of features, including frame-level, clip-level, identity-level and network-level of features. A cutting-edge multi-camera car tracking and Re-ID framework based on temporal-attention model and deep neural networks is improved to capture the frame-level, clip-level and identity-level of features. A Spatial-temporal Camera Graph Inference Model (StCGIM) are designed to integrate the network level of features into the MCCTRI framework. After obtained the multi-camera tracking result, the tracking accuracy levels of different cameras are various from each other. An Adaptative Accuracy Model (AAM) is designed to eliminate and unify errors and prepare the input for the traffic information estimation algorithms. Furthermore, different levels of traffic-related information can be estimated properly.

The author evaluated the framework based on five cameras video data on captured on the Interstate 5, including different views, orientations, lighting conditions and color settings in various challenging scenarios. Based on MCCTRI, not only including the traffic information value, such as link average speed, average travel time and volume, but also a more particular data format – the distribution of each parameter can be estimated precisely. All the value information estimation error is less than 8% through the dataset evaluation including five camera views. The KL distance of the estimated distribution and real distribution is less than 3.42. Based on the experiment, the MCCTRI gives the surveillance camera system a brain and more precise and valuable information can be extracted through the method.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I owe a lot thanks to so many nice people who helped and encouraged me. I would never have the thesis completed without the love, support and help from them. This long journey now comes to an end and it’s a great time to make a summary and express my appreciations.

First and foremost, a deep appreciation to my graduate adviser, Professor Yinhai Wang for his profound guidance and advice on both my research and life. I really appreciate the great research opportunities he provided to me. In addition, Professor Wang is a great guy that every word and action from him is professional and visionary.

I am sincerely grateful to Professor Jeff Ban and Professor Edward McCormack for serving on my thesis committee. They gave me insightful and valuable advice on improving my thesis work. Additionally, Also, the authors would like to thank the Pacific Northwest Transportation Consortium (PacTrans) at Regional University Transportation Center (UTC) for Federal Region 10, for support and funding this research.

I would like to express appreciation to my wife Jiarui Cai and her advisor professor Jenq-Neng Hwang for their help in developing the framework of this research and support me to join the CVPR AI city challenge 2019.

Finally, I would also like to thank my colleagues at Smart Transportation Application and Research Lab (STAR Lab) at the University of Washington. As the name, here is full of stars. I also want to say thanks to my fellows loudly, including Chenxi Liu, Kristian Henrickson, John Ash, Ruimin Ke, Zhiyong Cui, Meixin Zhu, Yifan Zhuang, Dr. Xin Fu, Dr. Chunsheng Liu, Muzhi Han, Yanlong Chen, Min Zhang. Wish you guys have a colorful and wonderful future!

I dedicate this thesis to everyone who I love and love me.

The end is also a new beginning!

Chapter 1. INTRODUCTION

1.1 GENERAL BACKGROUND

Traffic sensing is a crucial part of the Intelligent Transportation Systems (ITS) nowadays. The sensing results are the necessary input of various ITS related services, such as travel time estimation [1, 2], route guidance and dynamic traffic management [3, 4]. Generally, there are three primary sensors and data types for traffic managers to obtain the traffic information: loop detector data, vehicle trajectory data and traffic surveillance camera data [5].

The loop detector is an electromagnetic device installed under the road surface to detect vehicles in the road network. Though the detection result, traffic engineers can estimate the traffic parameters such as the traffic flow, speed, occupancy rate and length of the vehicle. However, the loop detectors must be installed under the road surface. Traffic engineers need to close the road during the installation and maintenance process, which is very inconvenient. In contrast, the vehicle trajectory data provided the GPS location records of each single vehicle at every basic time slot. Through the continuous records, researchers can extract the vehicle trajectory, and then estimate the traffic information. However, the penetration rate of the trajectory data is always shallow. At the same time, the accuracy of the trajectory data will be decreased in the urban area because of the GPS signal drift.

With the rapid development of computer vision technology, cameras are widely used in the ITS system [6-9]. Cameras are not only used for recording and collecting the traffic data, but also for the high-accuracy and real-time traffic information exaction, traffic monitoring and security management. Through the surveillance cameras, each single-vehicle information i.e., vehicle type, color, location and time can be captured. By processing the traffic video, the traffic parameters, such as traffic flow, density and speed can be estimated precisely.

Although many video-based methods have been proposed to estimate traffic information, they focused on single fixed surveillance camera analyses and cannot be applicable to multi-camera simultaneously [10, 11]. To extend studies with single fixed cameras and to facilitate cross-camera information and network-level traffic parameters extraction and estimation, Multi-Target Multi-Camera Tracking (MTMCT) and Vehicle Re-identification (Vehicle Re-ID) related research have emerged [12, 13]. In MTMCT, the system tracks multiple detected objects across multiple cameras of overlapping/non-overlapping views, which has rapidly increased in recent years. In general, the MTMCT technology includes the following parts: 1) Multi-Object Detection (MOD) and called single-camera tracking (SCT), 2) finding and associating the same objects detected by different cameras, also called Vehicle Re-ID, 3) linking all the tracks that belong to the same vehicle and restoring the spatial-temporal information of the object.

With the MTMCT technology, three levels of information can be extracted, including the single-vehicle level of information (vehicle type, color, brand etc.), point-based information (pointed based traffic volume, speed, and occupancy rate) and network-level of information. The network-level of interactive information such as point to point flow distributions, OD distribution, network-scale travel time distribution and speed distributions can be obtained based on tracking objects through different locations. The MTMCT technology enables the traffic cameras to work together based on an interactive and shared platform that each camera is not isolated anymore. It will be much more useful for traffic engineers to sense and control the traffic network.

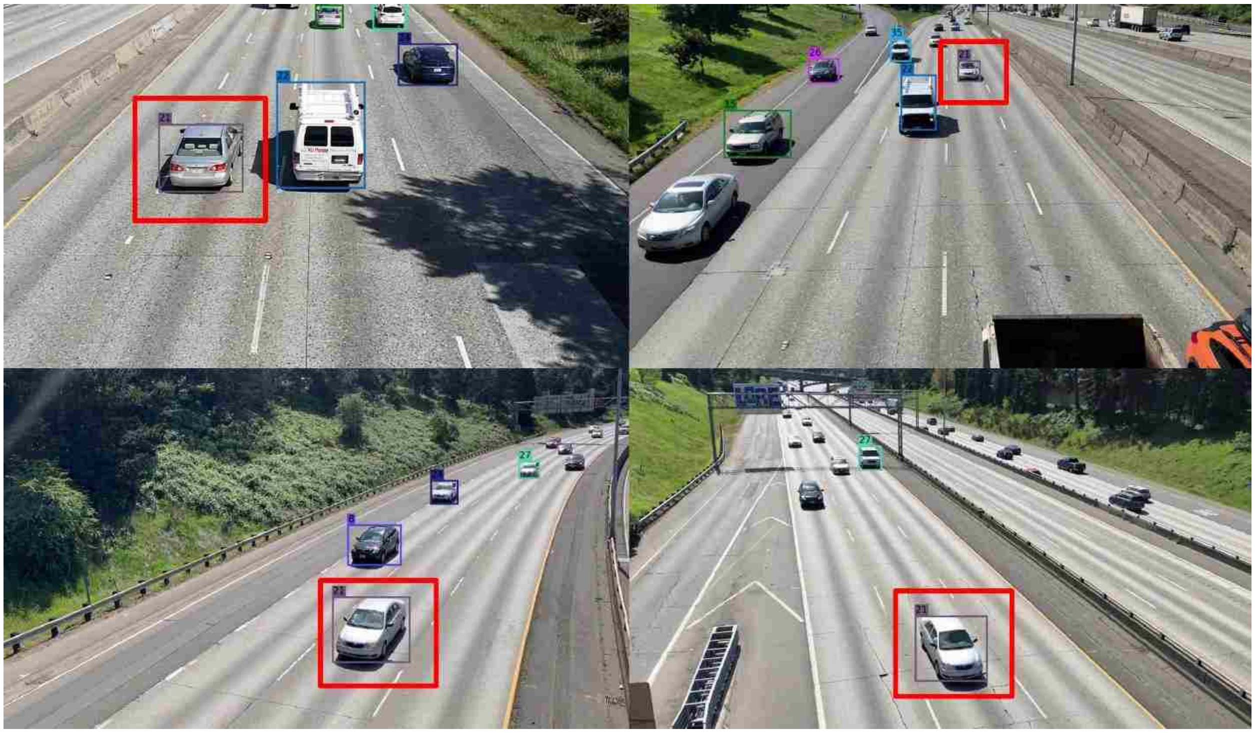


Figure 1-1 The illustration for vehicle MTMCT. Given a target vehicle (#21 in the figure), the aim of the MTMCT is to search and match the same vehicle from multiple cameras

1.2 PROBLEM STATEMENT

In the current surveillance camera system, hundreds of cameras have been installed in the road network. However, each of them is isolated. Every single camera extract information by itself. After the post possessing, the cross-camera information needs to be calculated and summarized and by human, which consumes much workforce. Since people's energy is limited, there will inevitably be omissions, especially while matching information through different cameras.

In this paper, the author aims to build a multi-camera traffic information estimation framework based on the cutting-edge single-camera traffic information estimation. Different from the traditional single camera-based traffic information estimation in the past two decades [14-19], traffic information estimation in a multi-camera system is much more complicated. To address the problems, the author divided the whole multi-camera information estimation framework into five sub-tasks in the following:

1. Single-camera multi-object detection and tracking;
2. Single-camera traffic information estimation;
3. Cross-camera multi-object Re-ID;
4. The multi-camera spatial-temporal graph inference model;
5. Adaptative accuracy for multi-camera traffic information estimation;

For a multi-camera system traffic information estimation methodology, the single-camera multi-object detection and tracking are the fundamental task of the traffic information estimation. The most challenging thing is how to link the detection and tracking results across several cameras. Not only a more reliable and accurate single-camera object detection, tracking and information estimation methods are necessary, but also a system-level of road network graph constraints, objects Re-ID and information estimation need to be set up.

In order to address the tasks, several problems are targeted to handle in this research. The first and foremost one is how to merge and well-use the existing road network features and information. Also, a proper way to link all the existing surveillance cameras installed in different locations based on a mathematics and computer language format is necessary. Then, a crossing camera vehicle Re-ID method is indispensable to link each single camera tracking result into a multi-camera system. The task is very challenging since in real-world transportation applications scenarios, the different camera orientations and lighting conditions are various in different locations. Different from the traditional human and vehicle Re-ID problem with several levels of accuracy evaluation, the most regions condition is that only the top-one candidate can be used to estimate the cross-camera traffic information automatically. Thus, how to link each single camera into a multi-camera system and even make use of the multi-target multi-camera Re-ID and tracking results to establish a novel framework to estimate the network-level traffic information becomes the main problem in the research.

1.3 RESEARCH DESIGN

Based on the five tasks in chapter 1.2, tasks one and two are single camera-related researches. Since the surveillance cameras are always installed at a fixed location with a specific orientation, choose a precise and fast a single camera multi-objects detector is the first step. With the detection output results, a well-applicable single-camera tracking method needs to be integrated into the framework [17]. With the single-camera multi-objects detection and tracking output, the traffic information can be estimated in different road network locations.

For the crossing-camera multi-target Re-ID task, a state-of-the-art image-based vehicle Re-ID framework [12] is targeted to adopt and improved by the author based on the video to video scenario. Since the surveillance cameras are installed in different locations, various orientations and diverse lighting conditions, merging the single-camera tracking results into clip level features is one of the efficient methods to expand and enlarge the features of the vehicle set. Also, not only including the deep CNN features for the vehicle appearance, but also the vehicle structure features, the orientation features and the original vehicle features (such as brand, type and model) are also needed to make good use in this challenging framework. The intuition and experience make the author apply two kinds of loss functions to train the model. The triplet loss is used to teach the model to distinguish different vehicle and the cross-entropy loss is trained the model to merge the feature belongs to the same vehicle captured by different cameras in various conditions. Finally, the vehicle identity level of features can be applied to re-ranking the results since the customized crossing-camera Re-ID method is designed to target the highest top-one accuracy.

The main challenges are in task four. Given a road network already installed with cameras, making good use of existing information is the first step. Node and link relationships, road network features and spatial-temporal information are all crucial information of the multi-camera system. Given the intuition that vehicles are only passing the cameras one by one in proper order no matter how complicated the target network. Thus, through the exhaustion of all potential route options, a camera link graph can be set up and summarized into a matrix format based on graph theory. With the underlying camera-link graph, many objective constraints can be merged into framework effectively, including the spatial-temporal features, road network features and travel time index. The searching window time slot and existent Re-ID candidates can be effectively narrow down. With these kinds of useful information, the author treated the whole process as a weighted optimization procedure and merged the information in a Spatial-temporal Camera Graph Inference Model (StCGIM). The model targets to provide with accurate and robust cross camera top1 vehicle re-ranking results and provide the input for traffic information estimation.

The last but not least part is task five. After obtained the multi-camera tracking and Re-ID results, how to calculate the precise traffic information based on different levels of MccTric accuracy becomes a challenge. Even the top multi-camera tracking method can only provide around 0.6-0.7 IDF1 accuracies [13]. Also, the accuracy levels are various in different camera viewpoints at different locations. Here, the author targets a framework to obtain precise traffic information based on various accurate levels, which is called Adaptative Accuracy Model (AAM).

1.4 Research Objectives

Inspired by the needs in exploring the novel traffic information estimation based on nowadays widely used multi-camera surveillance system, the author targets to propose a novel framework to achieve network-level traffic information estimation based on multi-camera sensing technology. Specifically, the research objectives are summarized in the following:

1. Detect and track the vehicles based on each surveillance camera;
2. Adopt and improve a cross-camera multi-object vehicle Re-ID method to find the same vehicles driving through different cameras with reliable top-one accuracy;
3. Propose a method to link the different cameras into a graph and merge other useful information, i.e., spatial-temporal information, connectivity information;
4. Propose a new method to narrow and optimize the vehicle candidates searching process;
5. Develop a framework for multi-camera traffic information based on various accuracy level of multi-camera tracking result;
6. Achieve a reliable and accurate result for road network traffic information estimation based on a multi-camera system.

Chapter 2. STATE OF THE ART

2.1 TRAFFIC SENSING APPROACH SUMMARY

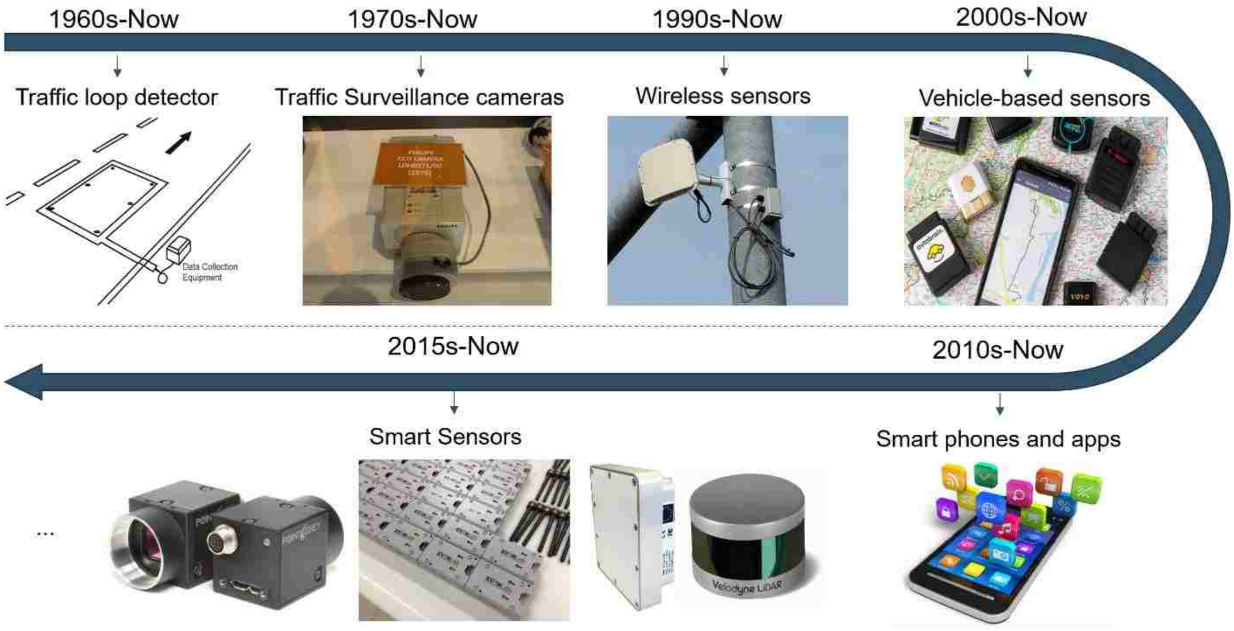


Figure 2-1 The development of different types of traffic sensors

Sensor-based traffic data collection has always been a hot topic for traffic researchers. The earliest traffic vehicle loop detector sensors date back to 60 years [18]. At that time, [18-19] proposed to a traffic sensing method by counting the vehicle by the loop detector. While the vehicle passing over the loop, it can cause the magnetic change and will trigger the counter. Through this method, people can obtain the traffic volume on the road section. After that, traffic monitoring cameras [20-21], wireless sensors [22], and other traffic sensors [23] are gradually installed on the roads. These sensors are usually installed and managed by the public authority, providing data and reference for decision-making by the traffic management department. The figure 2-1 listed the development of different types of traffic sensors from the 1960s.

Among all the sensors, the most widely used is the vehicle loop sensing systems and traffic surveillance cameras system. Vehicle loop sensor systems usually use magnetic loop detectors as the most basic sensing unit. Loop sensing systems can directly collect road data, including volume, speed, flow and vehicle type. This type of sensor has been widely used, and has formed a mature front-end data collection, back-end data storage and data display platform. Also, the traffic monitoring camera system often consists of one or more cameras installed in different road sections. The video cameras can provide real-time video at different locations, which is more intuitive. Of the two, the vehicle loop sensing system has a wider applicability and is less affected by weather factors, but the shortage is also apparent. The loop detector coverage is limited, which can only reach the lane level.

Furthermore, the installation and maintenance process are complicated. With the constraint of the system price, right now, the loop sensing system is often used on the essential road sections [24-26]. Compared with the vehicle loop, the cameras can provide more information, but it is more affected by the weather. A common disadvantage with both widely used sensors is that the data acquisition and processing of each sensor unit is independent and rarely to form a system work for users. For example, the information between the camera and the camera cannot be shared, not to mention tracking the travelers and vehicles. The loop detectors also collect and calculate data isolated based on each installed point. This information sharing challenges currently significantly limits the capabilities of the system of such widely deployed sensor networks.

In addition to road data, from the beginning of 2000, vehicle-based data and traveler-based sensors have gradually been adopted by researchers [27-32]. The data of trajectory sensors are often in the form of GPS coordinates. Researchers can extract the trajectory of each vehicle based on the data, and then obtain accurate vehicle position, point-to-point travel time, speed, acceleration and other information. The advantages of this type of information include 1) the road network with a wide range of information coverage; 2) it is not susceptible to external factors such as weather; 3) because the data is continuously collected, it has a strong correlation in space-time correlation, which can provide a reliable data source for data correlation mining. However, the shortcomings of such data are generally more obvious: 1) the penetration rate of the data is often shallow (usually less than 5% each road section at a particular time slot); 2) the accuracy is inconsistent, and there is a significant GPS position shift phenomenon in urban areas and so on. At present, this type of data collection and utilization has been becoming a hot research topic for several years.

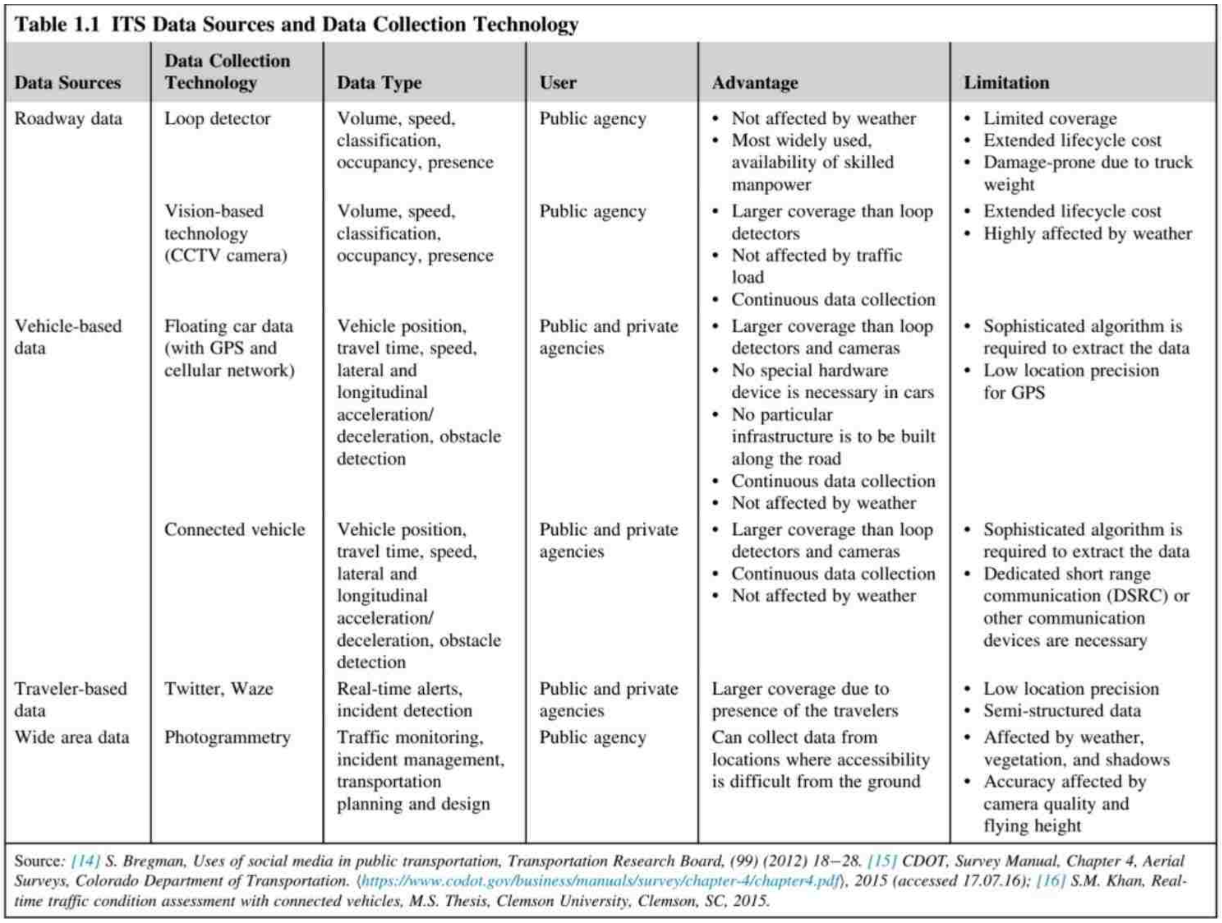


Table 1.1 ITS Data Sources and Data Collection Technology

Data Sources	Data Collection Technology	Data Type	User	Advantage	Limitation
Roadway data	Loop detector	Volume, speed, classification, occupancy, presence	Public agency	• Not affected by weather • Most widely used, availability of skilled manpower	• Limited coverage • Extended lifecycle cost • Damage-prone due to truck weight
	Vision-based technology (CCTV camera)	Volume, speed, classification, occupancy, presence	Public agency	• Larger coverage than loop detectors • Not affected by traffic load • Continuous data collection	• Extended lifecycle cost • Highly affected by weather
Vehicle-based data	Floating car data (with GPS and cellular network)	Vehicle position, travel time, speed, lateral and longitudinal acceleration/ deceleration, obstacle detection	Public and private agencies	• Larger coverage than loop detectors and cameras • No special hardware device is necessary in cars • No particular infrastructure is to be built along the road • Continuous data collection • Not affected by weather	• Sophisticated algorithm is required to extract the data • Low location precision for GPS
	Connected vehicle	Vehicle position, travel time, speed, lateral and longitudinal acceleration/ deceleration, obstacle detection	Public and private agencies	• Larger coverage than loop detectors and cameras • Continuous data collection • Not affected by weather	• Sophisticated algorithm is required to extract the data • Dedicated short range communication (DSRC) or other communication devices are necessary
Traveler-based data	Twitter, Waze	Real-time alerts, incident detection	Public and private agencies	Larger coverage due to presence of the travelers	• Low location precision • Semi-structured data
Wide area data	Photogrammetry	Traffic monitoring, incident management, transportation planning and design	Public agency	Can collect data from locations where accessibility is difficult from the ground	• Affected by weather, vegetation, and shadows • Accuracy affected by camera quality and flying height

Source: [14] S. Bregman, Uses of social media in public transportation, Transportation Research Board, (99) (2012) 18–28. [15] CDOT, Survey Manual, Chapter 4, Aerial Surveys, Colorado Department of Transportation. ⟨https://www.codot.gov/business/manuals/survey/chapter-4/chapter4.pdf⟩, 2015 (accessed 17.07.16); [16] S.M. Khan, Real-time traffic condition assessment with connected vehicles, M.S. Thesis, Clemson University, Clemson, SC, 2015.

Figure 2-2 Sensor Data Summary and Comparison [5]

Based on the research topic, video information is the primary source. The author summarizes the advantages and disadvantages of video data according to the current development of ITS and the characteristics of the data and showing in table 2-1 current surveillance video system advantages and disadvantages. Therefore, because of the problems existing in the current traffic monitoring system, the main research focus of this article will be to use cross-camera vehicle tracking to solve the problem of data sharing and isolation between cameras.

Table 2-1 Current surveillance video system advantages and disadvantages

Advantages	**Disadvantages**
Valuable data – Comparing with other data types, the information of the photos/videos are much richer. **Widely deployed** – Visible surveillance cameras have been already installed in our road network since 1970s. **Ease of use and installation** – Cameras only need to be installed somewhere over the roads, which is easy and reliable. **Aid commuters** – Intuitive, clear and easy-to-understand video data can be better shared with travelers.	**Limited coverage** – Each camera can only cover part of the road segment. **Limited data sharing and processing –** Each camera collects data separately. The information processing algorithms are still working for each camera independently. **Weather –** Traffic cameras are subject to damage caused by weather. Heat, wind, rain, snow and ice can all damage or ruin a traffic security camera.

2.2 SINGLE-CAMERA BASED METHODS

In nowadays traffic information estimation, single-camera information estimation is widely used by many traffic managers. The approaches used in the single-camera scenario are generally based on the **M**ulti-**O**bject **D**etection (MOD).

The multi-object detection in computer vision has always been an important issue. With the rapid development of computer vision technology, object detection has been widely used in face recognition, pedestrian tracking, license plate recognition, and autonomous driving. Compared with image classification, object detection is more complicated. Object detection is to combine object localization and object classification and use multi-directional knowledge such as image processing and machine learning to locate objects of interest from images (videos). The object classification is responsible for judging whether the input image contains the required object, and the object localization is responsible for indicating the location of the target object, then localization with a bounding-box. The task requires the computer to accurately determine the object class while giving the precise relative location of each object.

Since the concept of object detection was proposed, scholars have made unremitting explorations on this issue. Traditional object detection algorithms are mostly based on sliding window frames or matching based on feature points [33]. Since 2012, AlexNet [34] was elected to the annual ImageNet large-scale visual recognition challenge, and the effect is far superior to traditional algorithms, bringing the public's vision back to deep neural networks. The proposal of R-CNN [35] in 2014 made the CNN based object detection algorithm gradually become mainstream [36]. The application of deep learning has improved detection accuracy and speed. Therefore, the author believes that based on whether deep learning is applied, object detection algorithms can be divided into traditional algorithms and deep learning-based object detection algorithms. In this chapter, the author will discuss the main algorithms of traditional algorithms and deep learning-based object detection algorithms, analyze the advantages and disadvantages of related algorithms, and combine existing problems to select the object detection algorithms that are suitable for this study.

2.2.1 *Traditional Methods*

Traditional algorithms can be roughly divided into object instance detection and traditional object class detection: (1) The object instance detection problem usually uses templates and image-stabilized feature points to obtain the correspondence between the template and the objects in the scene and detect the object instance. Object instance detection focuses on the specific object itself, and the rest of the objects in the image are irrelevant. (2) Traditional object class detection uses the AdaBoost [37] algorithm framework, HOG [38] feature and support vector machine [39] and other methods to detect a limited number of classes based on selected features and classifiers.

The SIFT [40] algorithm proposed by Lowe, which finds feature points that are not easily affected by illumination, noise, and affine transformations to match objects, is a widely used keypoint detection and description algorithm. This algorithm uses Gaussian blur to achieve scale space, Difference of Gaussian function for extreme value detection, and then determines the edge principal curvature, screens out unstable points of edge response, and obtains key points which have stable matching and robust noise immunity. Finally, used the direction histogram to calculate the gradient and direction of the neighborhood of key points to obtain descriptors. The SIFT algorithm guarantees that the extracted features have invariant features such as translation, scaling, and rotation through a series of methods. It is also robust to light, noise, and small changes in viewing angle. However, the SIFT algorithm has problems such as high complexity, slow detection speed, and difficult to extract valid feature points for blurred images and smooth edges.

AdaBoost is a machine learning algorithm based on Boosting [40]. Initially, it is assumed that n samples in the training set have the same weight. After each training, adjust the weight of the data in the training set and increase the weight of the wrong samples, so that the next classifier can focus on the wrong samples. After N rounds of training, N weak classifiers are integrated, and corresponding weights are assigned according to the performance of each classifier to form a strong classifier with high accuracy and low error rate. The Viola-Jones [41-42] algorithm is the first face detection algorithm that can be processed in real-time and has a good effect. The proposal of this algorithm marks that the face detection has entered the actual application stage. In a nutshell, the VJ algorithm uses Haar-like features to describe the common attributes of the object, and uses the integral graph to achieve fast feature calculation. The cascade classifier is used to reduce the amount of AdaBoost calculations and quickly detect the object. Rainer Lienhart and Jochen Maydt extended the Viola-Jones detector with diagonal features to form the Haar [43] classifier. In addition, other algorithms are proposed like change the Stump function to a decision tree or use classifiers such as RealBoost and GentleBoost.

In general, the purpose of these traditional algorithms is to quickly perform feature calculations and predictions on the premise of ensuring the extraction of rich and accurate features. However, the features extracted by traditional algorithms are basically low-level, artificially selected features. These features are relatively more intuitive, easy to understand, and more targeted to specific objects, but they cannot express a large number of multi-class objects well.

2.2.2 *Object Detection Algorithm based on Deep Learning – Two-stage Detectors*

Since AlexNet used convolutional neural networks in the competition to greatly improve the accuracy of image classification, some scholars have tried to apply deep learning to object class detection. Convolutional neural networks can not only extract higher-level, better-expressing features, but also complete feature extraction, selection, and classification in the same model. In this regard, there are two main types of algorithms: one is the R-CNN series of object detection frameworks (two stages) based on the classification proposal and the CNN network; the other is the conversion of object detection into a regression problem Algorithm (single-stage).

OverFeat [44] is one of the first algorithms to apply deep learning to object detection. Strictly speaking, OverFeat does not use a region proposal, but its ideas are followed and improved by the subsequent R-CNN series. The algorithm uses multi-scale sliding windows combined with AlexNet to extract image features and completes detection. The mean Average Precision (mAP) on the ILSVRC 2013 dataset is 24.3%. The detection effect is significantly improved compared to the traditional algorithm, but there is still a high error rate.

Ross Girshick et al. proposed the R-CNN model. R-CNN uses Selective Search to obtain candidate regions. The candidate region size is then normalized and used as the standard input for the CNN network. Then use AlexNet to obtain the features in the candidate area, and finally use multiple SVMs for classification and linear regression to fine-tune the Bounding-box. R-CNN greatly improved the detection effect to 31.4% (ILSVRC 2013 dataset) and obtained 58.5% accuracy on the VOC2007 dataset (unless otherwise specified below, all are the detection results on the VOC2007 dataset). However, R-CNN performs feature extraction on nearly 2,000 candidate regions, and there are many repetitive regions between candidate regions, resulting in many repeated operations, running slowly, and the average processing time of each picture is 34

s. At the same time, the data of each step is stored, which significantly consumes storage space. Besides, normalizing the candidate regions will affect the final result.

For the shortcomings of R-CNN extracting features for all candidate regions separately, SPP-Net [45] performs a convolution operation on the entire picture to extract features at one time. The feature extraction has been changed from nearly 2,000 times of R-CNN to extracting the whole picture features once, which greatly reduces the workload. SPP-Net adds a spatial pyramid pooling layer (SPP layer) after the last convolution layer and before the fully connected layer to extract feature vectors of a fixed size to avoid the complex operation of normalizing the candidate region size. The above two improvements make SPP-Net's detection speed 38 ~ 102 times faster than R-CNN, and solve the problem of candidate region normalization. Although SPP-Net has replaced the convolutional network, the accuracy is almost the same. At the same time, SPPNet still does not solve the problem of R-CNN storage space consumption. The steps of determining candidate regions, feature extraction, object classification, and localization correction are still separate.

The Fast R-CNN [46] algorithm is based on the SPP-Net. The SPP layer is reduced to the ROI Pooling layer, and the output of the fully connected layer is decomposed by SVD to obtain two output vectors: the classification score of the softmax and the Bounding-box regression. This improvement merges the classification problem and the border regression problem, replaces the SVM with softmax, stores all features in video memory, reduces the occupation of disk space; and the SVD decomposition has almost no impact on accuracy, greatly Speed up detection. Fast R-CNN uses VGG16 instead of AlexNet. The average accuracy rate is 70.0%, and the training speed is 9 times faster than R-CNN. The detection speed reaches 0.3 s per image (excluding the region proposal stage). Fast R-CNN still uses the Selective Search method to select candidate regions. This step involves a lot of calculations. When running on the CPU, it takes on average 2 s to obtain the candidate area for each picture. It can be seen that improving Selective Search is the key to the speed improvement of Fast R-CNN.

From the perspective of feature extraction, SPP-Net and Fast R-CNN reduce the workload, but still do not solve the problem of slow selection of candidate regions by Selective Search. Faster R-CNN [47] uses RPN networks (Region Proposal Networks) instead of the Selective Search algorithm to enable object recognition to achieve exact end-to-end calculations. The RPN network performs a windowing operation on the feature map and uses a preset scale anchor box to map to the original map to obtain candidate regions. The RPN network input feature map shares calculations with the feature map in the fully connected layer. The use of RPN enables Faster R-CNN to complete candidate area, feature extraction, classification, and localization correction operations within a network framework. RPN makes Faster R-CNN only need 10 ms in the region proposal stage, the detection speed reaches 5 fps (including all steps), and the detection accuracy is also improved, reaching 73.2%. However, Faster R-CNN still uses ROI Pooling, which causes the subsequent network features to lose translation invariance, which affects the accuracy of final localization. After ROI Pooling, each region passes multiple fully connected layers and there are more repeated calculations. Faster R-CNN Using the anchor box on the feature map corresponds to the original image, and the anchor box undergoes multiple subsampling operations, corresponding to a large area of the original image, resulting in Faster R-CNN's poor detection of small objects.

Object detection should include two problems: classification problems and detection localization problems. The former has translation invariance and the latter has translation variance. R-FCN [48] uses full convolutional network ResNet [49] instead of VGG to improve the effect of feature extraction and classification; for the defect that full convolutional network does not adapt to translation variance, the algorithm uses specific convolutional layers to generate Position Sensitive Score Map which includes object spatial location information; the ROI Pooling layer is no longer connected to the fully connected layer to avoid duplicate calculations. R-FCN has an accuracy rate of 83.6%, and the average test time per image is 170 ms, which is 2.5 to 20 times faster than Faster-RCNN. However, R-FCN needs to generate a number of channels that increase linearly with the number of classes. This process improves the object detection accuracy, but slows down the detection speed, making it difficult to meet the real-time requirements.

Mask R-CNN [50] is an improved algorithm based on Faster R-CNN, increasing the focus on instance segmentation. In addition to classification and localization regression, the algorithm adds parallel branches on instance segmentation and jointly trains the three losses. Instance segmentation requires the accuracy of instance localization to be at the pixel level, while Faster R-CNN introduces errors in the equal scaling process of the ROI Pooling layer, resulting in coarse spatial quantization and inaccurate localization. Mask R-CNN proposes bilinear difference RoIAlign to obtain more accurate pixel information, which improves the mask accuracy by 10% to 50%; Mask R-CNN also uses the ResNeXt [51] basic network in the COCO dataset The detection speed is 5 fps, and the detection accuracy is improved from 19.7% to 39.8% of Fast R-CNN. Mask R-CNN has reached the current high level in terms of detection accuracy and instance segmentation. Since then, some algorithms have improved in performance, but basically maintained at the same level. However, the detection speed of the algorithm is still difficult to meet the real-time requirements, and instance segmentation is currently facing the problem of too expensive labeling.

Starting from R-CNN, researchers focused on the problem of object detection to classification, and adopted the idea of "region proposal + CNN feature + SVM", using the CNN network to greatly improve the accuracy of detection; later SPP-Net , Fast-RCNN, Faster-RCNN, etc. basically follow this idea and improve the detection efficiency; but FasterRCNN can only reach 5 fps, which is slightly insufficient in terms of real-time performance. Although the subsequent R-FCN has improved, the effect is still unsatisfactory. In this regard, the researchers proposed another new idea, directly transforming the object detection to regression, and using a picture to get the bounding box and class.

2.2.3 *Object Detection Algorithm based on Deep Learning – Single-stage Detectors*

You Only Look Once (YOLO) algorithm is a very famous single-stage detection algorithm. From R-CNN to Faster-RCNN, object detection always follows the idea of "region proposal + classification". Training two models will inevitably lead to an increase in parameters and training volume, affecting the speed of training and detection. Therefore, YOLO [52] proposed a "single-stage" idea. YOLO divides the picture into S × S cells. Each cell is only responsible for detecting the object whose center falls on the cell. Each cell needs to predict two scales bounding box and class information, and predict the bounding box, object confidence, and class probability of the objects contained in all regions at one time. YOLO replaces the region proposal with a multi-scale region centered on the cell, discarding some accuracy in exchange for a significant increase in detection speed. The detection speed can reach 45 fps, which is sufficient to meet real-time requirements; the detection accuracy is 63.4%, compared with 73.2% of Faster R-CNN, the gap is larger. In the case of greatly improving the detection speed, YOLO also has the following problems: (1) As each cell only predicts two bounding boxes, and the classes are the same, so the detection effect is poor for the objects whose center falls in a cell at the same time and small objects, and there are many missed detections in a multi-object environment; (2) Due to YOLO's determination of the localization box is slightly rough, the localization accuracy of object location is not as fast as Fast-RCNN; (3) Detection is not good for unconventional objects.

YOLOv2 by adding batch normalization, multi-scale training, and K-mean dimensional clustering after each convolutional layer, the detection speed and accuracy are improved again. The algorithm can achieve a detection speed of 67 fps at the same time with a 76.8% accuracy rate and 40 fps at a 78.6% accuracy rate. YOLO v2 is faster than other detection systems in a variety of monitoring data sets, and can be traded off in speed and accuracy. The performance of this algorithm basically represents the current advanced level in the industry. The same article also proposed YOLO9000 [53]. This algorithm uses WordTree hierarchical classification, mixes detection data and recognition data sets, and trains on both classification and detection data sets to achieve 9 418 types of detection. YOLO 9000's network structure allows real-time detection of more than 9,000 object classifications, thanks to its ability to optimize detection and classification functions simultaneously. Using WordTree to mix training data from different resources, and using joint optimization technology to train on ImageNet and COCO datasets at the same time, YOLO9000 further reduces the gap between the monitoring dataset and the recognition dataset.

Prior detection system of YOLOv3 [54] reuses the classifier or locator to perform the detection task. They applied the model to multiple locations and scales of the image. Those with higher scores can be regarded as the test results. In addition, compared with other object detection methods, YOLOv3 uses an entirely different method. It applies a single neural network to the entire image. The network divides the image into different regions, and thus predicts the bounding box and probability of each region. The predicted probability weights these bounding boxes. YOLOv3 has some advantages over classifier-based systems. It looks at the entire image during testing, so its predictions take advantage of global information in the image. Unlike R-CNN, which requires thousands of single object images, it makes predictions through a single network evaluation. This makes YOLOv3 very fast. Generally, it is 1000 times faster than R-CNN and 100 times faster than Fast R-CNN.

Faster-RCNN detection has high detection accuracy but slow detection speed. YOLO detection accuracy is not as fast as Faster-RCNN detection but fast detection speed. Single Shot MultiBox Detector (SSD) [55] combines the advantages of the two and borrows the idea of RPN on the basis of YOLO, and gives consideration to the detection speed while ensuring high precision detection. Because the feature maps of different layers have receptive fields of corresponding sizes, the feature maps of a specific layer only need to train object detection at corresponding scales. Therefore, SSD uses high-level and bottom-level feature maps to perform regression using multi-scale regional features. The mAP of SSD300 can reach 73.2%, which is basically the same as Faster R-CNN (VGG16), and the detection speed reaches 59 fps, which is 6.6 times faster than Faster R-CNN.

However, SSD has the following problems: (1) small objects correspond to small areas in the feature map and cannot be fully trained, so the detection effect of SSDs on small objects is still not ideal; (2) when there are no candidate regions, it is more difficult to return to the region. It is easy to cause problems such as difficulty in converging. (3) The feature maps of different layers of the SSD are used as independent inputs of the classification network, resulting in the same object being simultaneously detected by boxes of different sizes and repeated operations. The R-SSD [56] algorithm increases the association of feature maps between different layers on the basis of SSD to avoid the problem of duplicate boxes of the same object; at the same time, increases the number of feature maps in the feature pyramid to improve the detection effect of small objects. The mAP of the algorithm is 80.8%, which is slightly higher than the SSD. However, the increase of the feature map leads to an increase in calculation amount and a decrease in detection speed, which is only 16.6 fps.

Whether it is the YOLO series or the SSD algorithm, the R-CNN series algorithm is used to perform classification and pre-training on large data sets, and then fine-tune on small data sets. The application of deep learning has improved detection accuracy and speed.

2.3 MULTI-CAMERA BASED METHOD

In the traffic area, many surveillance cameras have been installed. It would be advantageous to use these surveillance cameras for traffic information extraction and estimation comparing with other specialized hardware. The data from these cameras have been used extensively to handle vehicle detection problems. Right now, if people want to collect information through different cameras, a large amount of brute-force human labor work is necessary. However, vehicle Re-ID researches have escalated in the past few years and now they are booming.

The object Re-ID process is to identify a particular object as the same one as the previous observations. As for vehicle Re-ID, the process is to identify and match the target vehicle in different cameras without overlapping views, as shown in Figure 2-3. When a target vehicle appears, vehicle Re-ID will show if the vehicle was obverse by other cameras somewhere else. So, the vehicle Re-ID technology breaks the ice that each camera installed at different locations works isolated. With the vehicle Re-ID, the surveillance cameras can be used together to detect and track the same object at different locations. The emergence and boom of vehicle Re-ID technology are because (1) the increasing public safety and video information extraction needs and (2) the extensive use of surveillance camera networks in the road network, university campuses, parking garages and streets. With the vehicle Re-ID technology, spot a query vehicle or track the vehicle cross multiple cameras in the surveillance networks that can be done accurately and efficiently.

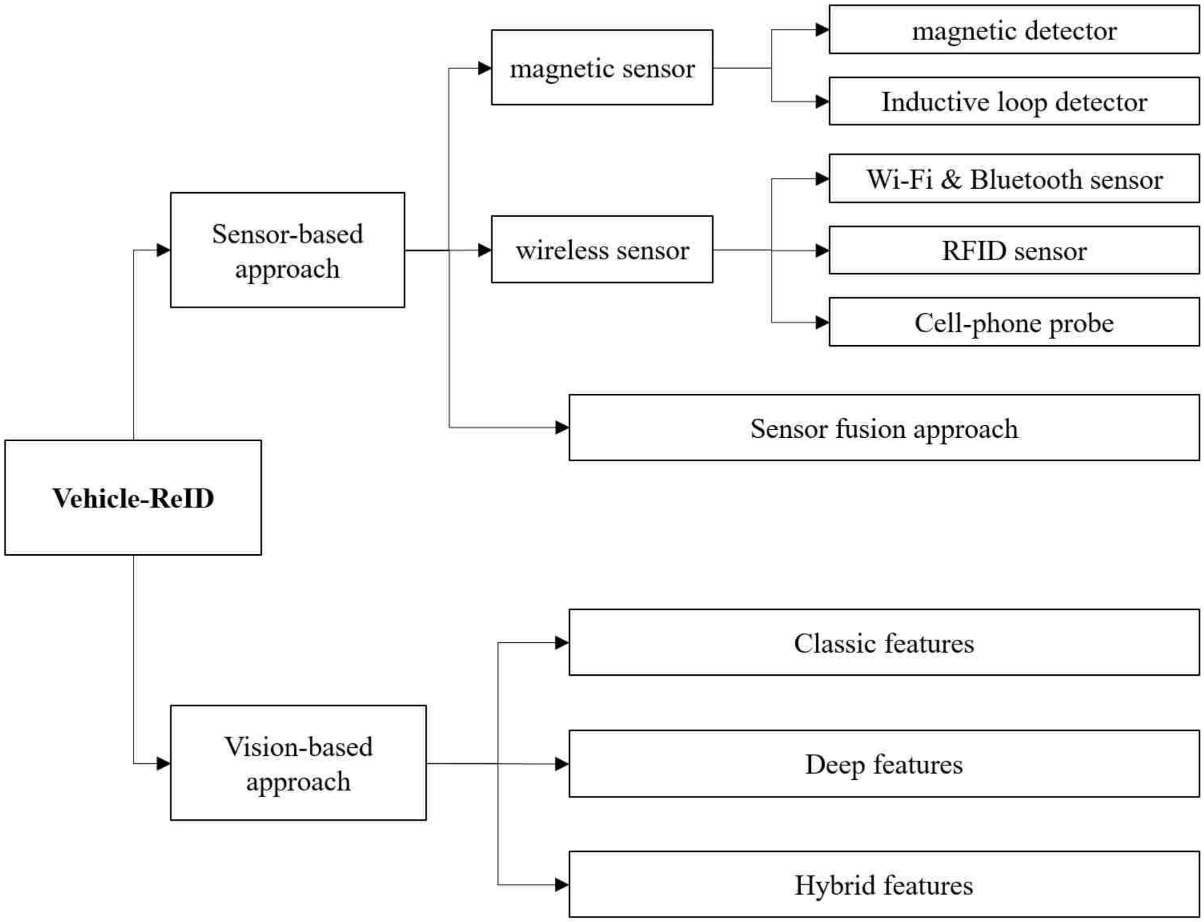


Figure 2-3 The vehicle Re-ID methodology summary

In this paper, the author divided the vehicle Re-ID methods into two big categories: sensor-based approach and vision-based approach. Vehicle Re-ID research was born based on multiple sensor-based approaches by matching the vehicle signatures detected by traffic sensors. The solutions are including magnetic sensors, inductive loop detectors, GPS & RFID sensors, Cellular phones, and even sensor fusion and hybrid methods. Except for the sensors-based approach, with the development of the hardware computational ability, vision-based methods boom and show a lot of potentials. In this paper, the author classified the vison-based approach into two sub-categories: classic features-based method and deep features based methods. In this research, the authors mainly focus on the vision-based approach.

2.3.1 *Sensor-based Approach*

Ten years ago, vehicle Re-ID researches were mainly relied on sensors. The magnetic sensor was the first kind of sensors used in the vehicle Re-ID problem in 1990. As we all know that vehicles are mainly made of metal. When an object composed of metal moves in the magnetic field, the distribution of the magnetic field of generated by the Earth will be interfered. So, researchers try to match the interference pattern triggered by different vehicles and aim to find the same one passing different sensors installed at various locations. Researchers proposed the magnetic sensors-based vehicles Re-ID solutions based on different methodologies, including magneto-inductive probes, three-axis magnetic sensor and anisotropic magneto-resistive sensor etc. from 1990s [62-76]. Based on these solutions, researchers tried to estimate the real-time travel time information from point to point. However, the accuracy of the approaches is poor since the vehicles in the same size are hard to distinguish based on magnetic changes. Also, the maintenance of the magnetic sensors is an inconvenience work. Engineers need to close the road sections during the installation and maintenance process.

With the development of wireless communication technology, the Wi-Fi, Bluetooth, GPS, RFID and cell phone data are become approaches to distinguish and find vehicles in the vehicle Re-ID problem. Since each wireless communication protocols are including the signature which use to find and distinguish the target to build communication, the special signature is also use to Re-ID vehicles. Wi-Fi & Bluetooth mac address is a unique signature for every device in the protocol and researchers, including [77-80], use it to match the same vehicles in road network. At the same time, many researchers use cell phone data to find the same vehicle and try to estimate traffic information [81-83]. The Radio-frequency identification (RFID) tag, widely used for toll collection, is also used to solve the vehicle reidentification challenge [85]. Researchers consider the vehicle which carried the wireless facilities as a moving node in the road network and attempt to match the same node when the detectors capture the same one in the different locations. However, these methods are born with many inherent limitations such as the wireless protocols are opened on the user's device. If the user closes the Wi-Fi, Bluetooth, or RFID functions, the vehicle cannot be detected. Also, the penetration rate of wireless facilities is difficult to estimate, and there will be significant differences in different road sections and different cities. At the same time, the data privacy issues are becoming a general concern since the mac address is unique for each user. With encryption algorithms are becoming more reliable, these wireless solutions are powerless.

People have an intuition that a better result will be achieved if multiple sensors provide the detection results. So, some researchers try to combine multi-sensors, including wireless sensors, magnetic sensors and even images to get more reliable and comprehensive results [86-90]. The popular hybrid sensor solution is to try to combine the wireless sensor and magnetic sensor for vehicle Re-ID results and proposed the signal processing algorithm to find the same object vehicle in different locations. However, the costs of a multi-sensor solution are obvious since more sensors need to install in the road network. Also, the complicated algorithm designed for hybrid sensor solutions and the high price makes it hard to use in the large-scale road network. With the computer vision technology booming, vision-based methods are recognized by more researchers.

2.3.2 *Vision-based Approach*

In the computer vision-based approach, the task of the vehicle Re-ID is to identify and match the target vehicle among multiple surveillance cameras installed in different positions with non-overlapping views. Due to increased demands on traffic sensing and public safety, camera networks are installed on many public areas, including roads, parks, communities, universities, streets and even communities. In such a complicated environment, it’s a huge challenge for the identify target vehicles by laborers. So, to free labor, a vision-based approach in vehicle reidentification becomes a hot research topic recently.

2.3.2.1 Classical features Approach [93-100]

Vehicle Re-ID algorithms based on classical features are mainly based on traditional empirical rules, extracting and identifying differentiated features in different images, and then matching the same target object. These traditional special features usually include license plate numbers, colors, textures, sizes, the histogram of oriented gradients (HOG) features and so on. Researchers have designed corresponding feature extraction algorithms to further match features in images cross cameras. The advantage of the classical feature-based method is that the feature capture logic is straightforward, and it is easy to interpret and explain the matching results. However, the problem is also apparent. The accuracy of classical features approaches is limited since these traditional features are not sufficient to Re-ID the vehicles across many cameras. Different kind of classical features extraction approach for each camera view is unique, and special algorithms need to be designed for extraction. At the same time, a lot of manual work is unavoidable, such as labeling a large number of outline features and key point features. At the same time, the matching between different features is not a simple add or linear relationship. When multiple features are superimposed and used, sophisticated algorithms are needed to fuse them. At present, vehicle re-recognition methods based on traditional features have gradually faded from people's research.

2.3.2.2 Deep features Approach [13] [101-115]

The rapid development of convolutional neural networks in recent years has dramatically promoted research topics related to vehicle recognition, such as vehicle verification, vehicle classification and attribute estimation. The task of vehicle Re-ID and retrieval under traffic cameras has always been a challenging subject. The focus of the former researchers tried to extract the vehicle features based on the whole vehicle image. However, the size of the vehicle in the surveillance are generally not large enough to support these methods, which leads to a bottleneck for the vehicle Re-ID. Therefore, some researchers have started to pay attention to local scales. Commonly used ideas for extracting local features are vehicle key-point localization and region segmentation. Based on the key-point localization and alignment results, some methods extract the features of the key part of the object and make a detailed comparison to achieve good results [101-105].

In the vehicle Re-ID work published by ICCV 2017, [104] used the method of key points positioning and area segmentation based on key points to label the vehicle image as 20 key points. From the image to be identified, multiple regions of the target vehicle were obtained. Then, use a convolutional neural network to extract regional feature vectors from multiple region segmentation results. After that, fuse the regional feature with global feature vectors to obtain the appearance feature vector of the target vehicle. Finally, the fused feature vector instead of the direct comparison of the appearance features is used for vehicle Re-ID, which solves the problem that different regions of different vehicle images cannot be compared efficiently. The authors considered the impact of orientations in the vehicle Re-ID; however, the result was still limited by the diversity of the real situation. Vehicles in a different location always have a different orientation, and the feature distribution also varies to each other. Traffic engineers need to mark key points for vehicle pictures at different angles, resulting in huge labor work. Therefore, from the perspective of feasibility and adaptability, this method is too complicated.

In the past two years, many vehicle Re-ID methods have used quantitative features and attributes of vehicles to detect and retrieve regions of interest in images quantitatively. Liu et al. [101] proposed a vehicle Re-ID system to complete coarse-to-fine-grained vehicle retrieval in the feature space. At the same time, a vehicle Re-ID data set Veri-776 was proposed. Using this data set, combined with the vehicle appearance, spatial-temporal information and license plate information are used to learn the similarity between image pairs. Liu and others [102] proposed a deep relative distance learning (DRDL) method, which uses two branches of deep convolutional neural network. The network transforms the original vehicle image into Euclidean space and can directly use the distance to measure the similarity of any two vehicles. Wang et al. [104] proposed a framework containing functional modules, which extract local area features in different orientations based on the position of 20 key points. Moreover, the extracted local features can be well combined with global features by the regularization module. Shen et al. [105] proposed a two-stage framework. A pair of vehicle images and the spatial-temporal information of the images need to be used as input of the method. Then, a vehicle candidate “visual-spatial-temporal” path is generated by a customized model with deep learning method, and then generate the query and gallery similarity scores. Zhou and others [111] proposed to take advantage of CNN and long-term short-term memory (LSTM) neural network to learn the transformation of vehicles across different camera views based on different orientations. Then, a multi-view vehicle representation information set, in which a fuse of different viewpoints can be inferred from a single orientation input. These former methods provide the author with beneficial potential ways to merge and to find vehicles in different camera views, which is very helpful to the multi-camera vehicle Re-ID process.

2.3.2.3 Hybrid-features approach [12], [115-116]

Recently, metric learning becomes more and more popular in vehicle Re-ID. Metric learning mainly solves the problems of similarity between classes (inter-class) and intra-class differences (intra-class). However, due to subtle inter-class differences and significant intra-class differences of vehicles, for example, vehicles belonging to the same ID show differences due to different attitudes, backgrounds, and orientations. At the same time, vehicles with different IDs (between classes) show a more significant similarity between classes, such as two different vehicles (with different IDs) of the same brand and color, the appearance characteristics of are very close under the same perspective. Therefore, compared with human Re-ID tasks, vehicle Re-ID is more complicated. So, to accurately distinguish two very similar vehicles smoothly, in addition to finding a distinguishable area that can distinguish the two vehicles in the image, the more important thing is being able to extract better, learn and compare the characteristics of these distinguishable areas. In addition to the appearance features representation, spatial-temporal information, road information, route information, trajectory information and even vehicle attributes information all becomes very important to learn and merge.

The deep metric learning and hybrid features are introduced to solve the vehicle Re-ID problems of feature learning between intra-class similarity and inter-class similarity further. In order to achieve the goal, many cutting-edge methods use deep networks to learn feature embedding space to maximize the distance of the feature between different classes, as well as minimize the distance in the same classes at the same time. In particular, the triplet constraint was introduced to learn feature embedding [12-13] based on the principle that "samples belonging to the same vehicle ID are closer than samples belonging to different IDs". This triplet constraint has been widely used for pedestrian re-recognition and face recognition tasks. Based on triplet loss, [12] customized the temporal-attention model and fuse the inter-class features (different models, brands, year of manufacture, etc.) as the ranking module to improve the generalization ability of the vehicle representations. Besides, some related work focuses on hybrid features and deep features to achieve good results in the recent vehicle Re-ID tasks on the public dataset [117].

Table 2-2 The vision-based vehicle Re-ID method summary

Approach	**Year**	**Model**	**Dataset**	**Performance measurement**	**Reference**
Classical features	2003	3D Color Model	Author collected	Standard deviation	Woesler (2003) [93]
	2005	Edge based model	Author collected	Hit rate	Shan et al. (2005) [94]
	2008	Appearance features model	Author collected	Correct probability	Guo et al. (2008) [95]
	2009	Pose and illumination model	Author collected	Hit rate	Hou et al (2009) [96]
	2012	Motion model	Author collected	Hit rate false positive	Feris et al. (2015) [97]
	2015	Individual paintings matching	Author collected	Cumulative match curve	Zheng ey al. (2015) [98]
	2016	3D color histogram model	Author collected	False positive	Zapletal and Herout (2016) [99]
	2017	String matching	Author collected	Hit rate	Watchar (2017) [100]

Deep features	2016	CNN SNN	VeRi-776, Author collected	Mean Average Precision (MAP)	Liu et al. (2016) [101]
	2016	Deep relative distance model	CompCars, VehicleID	Mean Average Precision (MAP)	Liu et al. (2016) [102]
	2016	Texture color	VeRi, Author collected	Mean Average Precision (MAP)	Liu et al. (2016) [103]
	2017	Siamese-CNN+Path-LSTM network	VeRi-776, CompCars	Mean Average Precision (MAP)	Wang et al. (2017) [104]
	2017	Siamese-CNN+Path-LSTM network	VeRi-776	Mean Average Precision (MAP)	Shen et al. (2017) [105]
	2017	CNN	VehicleID	Mean Average Precision (MAP)	Kanacı, A. et al (2017) [106]
	2017	CNN	VeRi	CMC measure	Zhang et al. (2017) [107]
	2017	CNN	VeRi	Mean Average Precision (MAP)	Tang et al. (2017) [108]
	2017	Deep joint discriminative model	VehicleID	Mean Average Precision (MAP)	Li et al. (2017) [109]
	2018	Deep neural network	VeRi	Mean Average Precision (MAP)	Liu et al. (2018) [110]
	2018	CNN-LSTM	BoxCars, Author collected	Mean Average Precision (MAP)	Zhou et al. (2018) [111]
	2018	GSTE	PKU-Vehicle, VehicleID, VeRi, CompCars	Mean Average Precision (MAP)	Bai et al. (2018) [112]
	2019	Deep features and attention model	Cityflow	Mean Average Precision (MAP)	Lv et al. (2019) [113]
	2019	Deep features and metric learning	Cityflow	Mean Average Precision (MAP)	Chen et al. (2019) [114]
	2019	Deep features and metric learning	Cityflow	Mean Average Precision (MAP)	Hsu et al. (2019) [13]
	2019	Deep features and metric learning	Cityflow	Mean Average Precision (MAP)	Chang et al. (2019) [115]
Hybrid Features	2019	Hybird features and attention model	Cityflow	Mean Average Precision (MAP)	Tan et al. (2019) [116]
	2019	Hybird features and attention model	Cityflow	Mean Average Precision (MAP)	Huang et al. (2019) [12]
	2019	Hybird features and deep learning model	Cityflow	Mean Average Precision (MAP)	Tan et al. (2019) [117]

Chapter 3. THE MCCTRI FRAMEWORK

3.1 OVERALL FRAMEWORK ARCHITECTURE

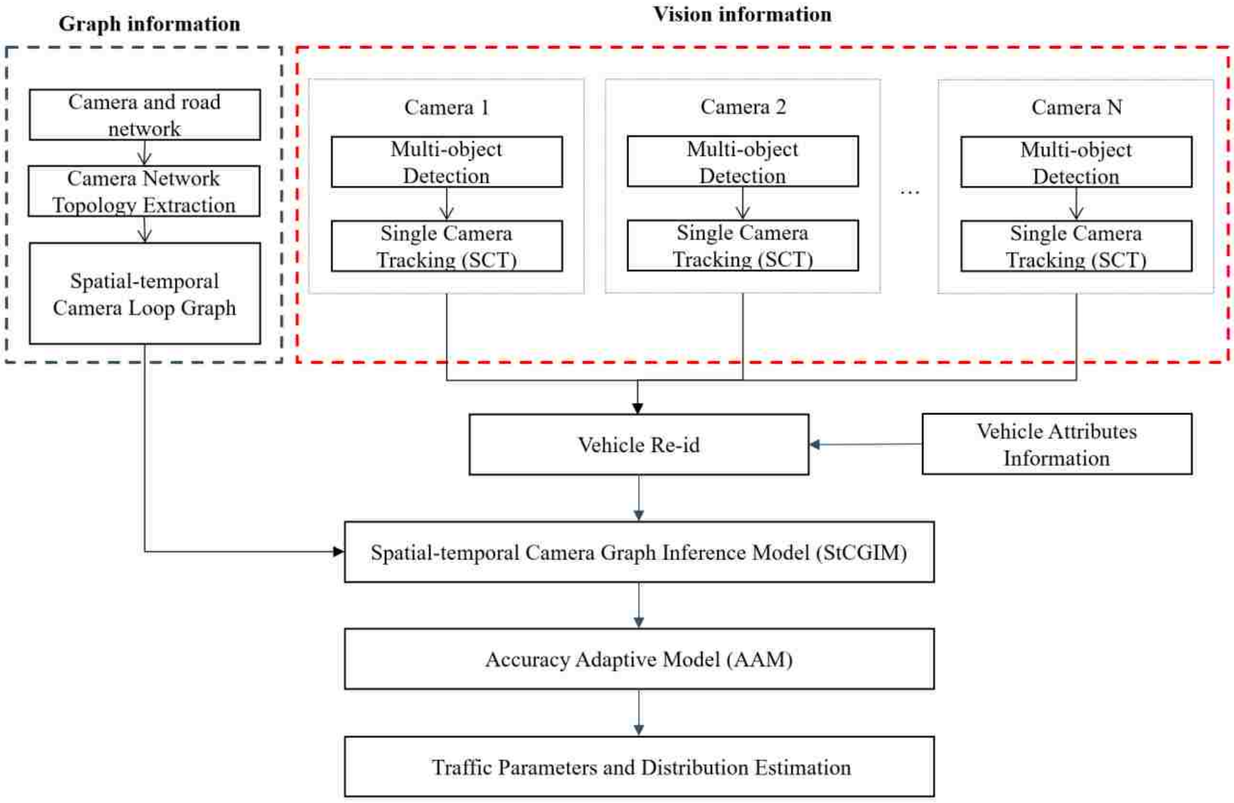

Graph information
Vision information
Camera and road network
Camera Network Topology Extraction
Spatial-temporal Camera Loop Graph
Camera 1
Camera 2
Camera N
Multi-object Detection
Single Camera Tracking (SCT)
...
Vehicle Re-id
Vehicle Attributes Information
Spatial-temporal Camera Graph Inference Model (StCGIM)
Accuracy Adaptive Model (AAM)
Traffic Parameters and Distribution Estimation

Figure 3-1 The overall framework of MCCTRI

The overall framework of MCCTRI is including several parts, which shows in figure 3-1 the overall framework of MCCTRI. The first and foremost is two parts of information extraction, including the vision information and the graph information extraction. For the vision information, the multi-object detection and single-camera tracking are two fundamental procedures, and details are in chapter 3.2.1 and 3.2.2. For the graph information, the spatial-temporal camera loop graph needs to be extracted and send to the StCGIM. The details are in chapter 3.5. With the graph information and the vision information, a cutting-edge vehicle Re-ID method is customized and developed here (chapter 3.3). The vehicle attributes are also integrated into the cross-camera Re-ID process (chapter 3.3.2). After the first round of Re-ID, the StCGIM will re-rank the candidates and select the top1 target as a result (chapter 3.5). Based on the result of StCGIM, an Adaptative Accuracy Model (AAM) is built to estimate the traffic parameters and distribution, even the multi-camera tracking accuracy is different (chapter 3.6). With the help of AAM, the last step is to estimate the information and get the results.

3.2 Vision Information Extraction

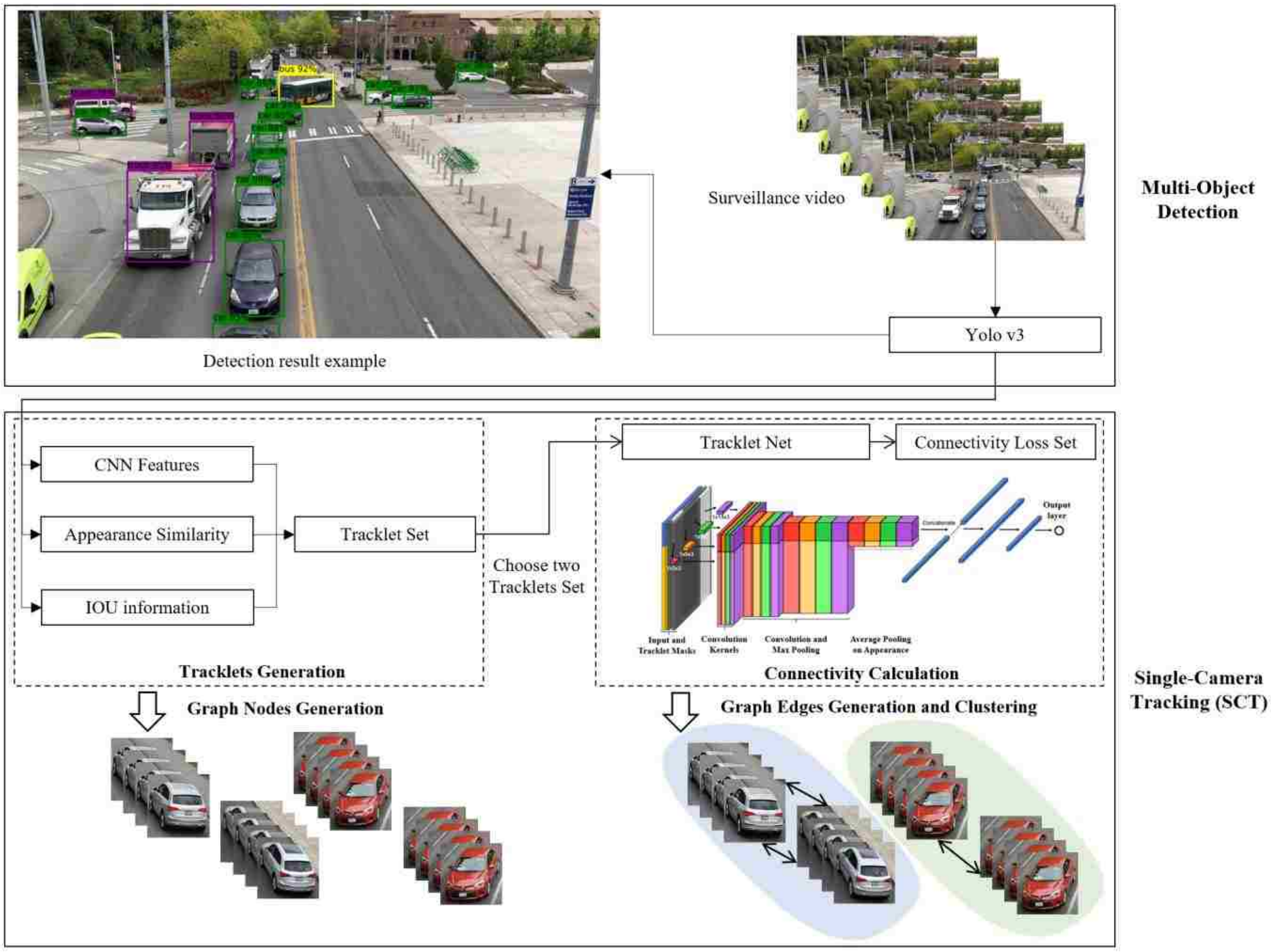


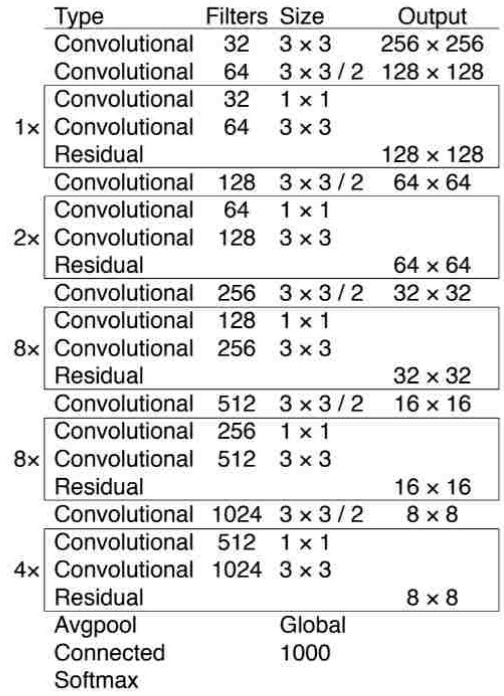
Figure 3-2 The overall structure of the vision information extraction

For the vision information, there are two parts: Multi-Object Detection (MOD) and Single-Camera Tracking (SCT). Figure 3-2 the overall structure of the vision information extraction shows the overall framework of the procedure.

3.2.1 *Single Camera Detection*

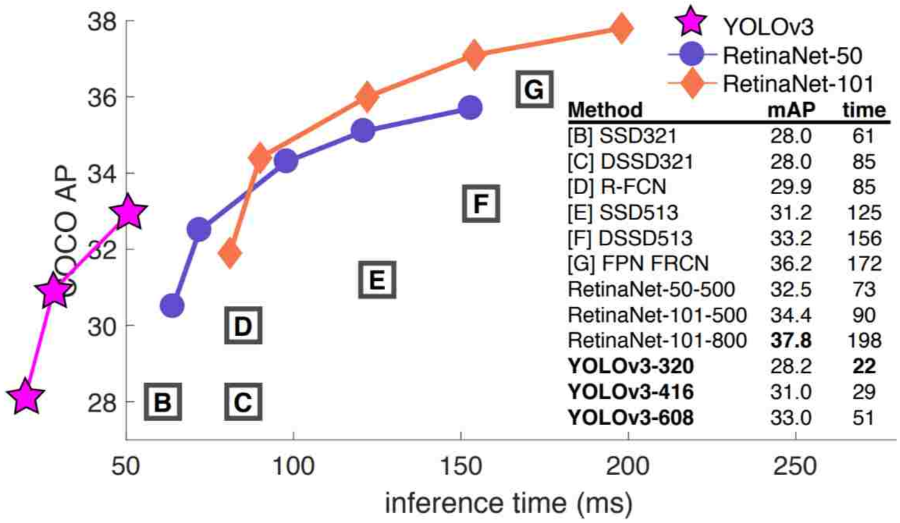
For the single-camera detection part, the author chooses YOLOv3 as the detector in this scheme [54]. The first procedure of YOLOv3 is to extract features from the input image through the feature extraction network to obtain a feature map of a specific size. Then divide the input image into multiple grid cells according to the size. If the center coordinates of an object in ground truth fall in which grid cell, then the grid cell will use to predict the object. Since each grid cell will predict a fixed number of bounding boxes (3 bounding boxes in YOLO v3), the initial sizes of these bounding boxes are different. Among these bounding boxes, only the largest bounding box with the ground truth Intersection Over Unit (IOU) is used to predict the object. After that, the network will further perform category prediction based on the characteristics of the features in the grid cell. The YOLO v3 architecture is called Darknet-53, which is shown in figure 3-3, the YOLOv3 architecture [54]. This network basically uses 53 convolutional layers with five residual blocks. The overall performance with other cutting-edge methods is in figure 3-4, the performance comparison of the YOLOv3 with other cutting-edge methods.

From figure 3-4, it can be seen that the YOLO v3 has a good performance as while as maintaining a fast detection speed. If the size of the input image is an image of 320 × 320, which can be run in 22ms through Yolo v3, and the mAP reaches 28.2. This record three times faster than SSD and the performance is very close. Through the YOLOv3 can largely boost the video processing speed as well as maintain an outstanding object detection results in this research.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3-3 The YOLOv3 architecture [54]



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

Figure 3-4 The performance comparison of the YOLOv3 with other cutting-edge methods [54]

3.2.2 *Single Camera Tracking*

Tracking-by-detection schemes are widely used in nowadays popular Multi-Object Tracking (MOT) methodologies, especially in the traffic surveillance camera scenario. Generally, these methods aim to associate the detection results across different frames and connect the same objects based on the common features, which achieve reliable results even in some occlusions scenarios. Graph models are also used to solve MOT problems and improve model efficiency by optimizing the frames' connectivity. Two types of graph models are widely used, including treating the detected objects as vertices, or building the graph vertices according to the tracklets. In common, the tracklet-based graph model can achieve better results in most scenarios since these methods not only utilize the information from a single frame, but also use the short time period trajectories to capture and measure the connectivity between vertices.

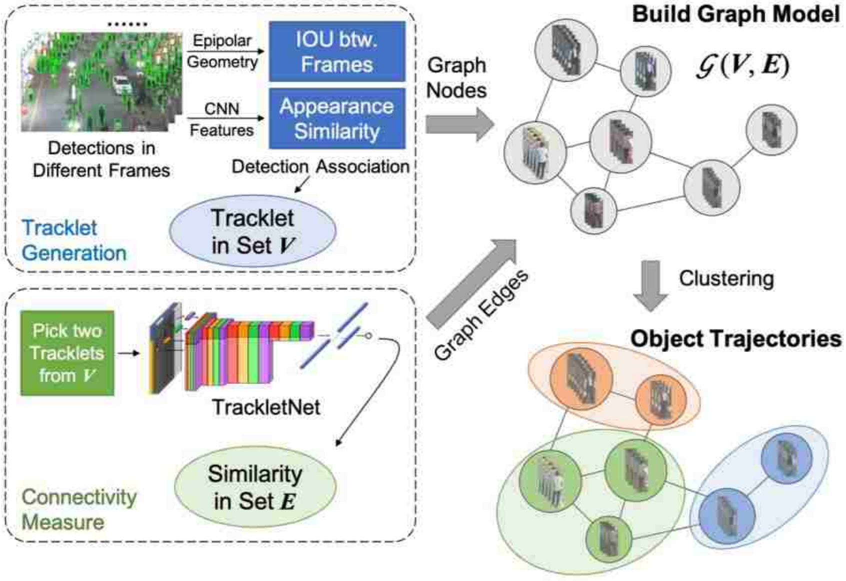
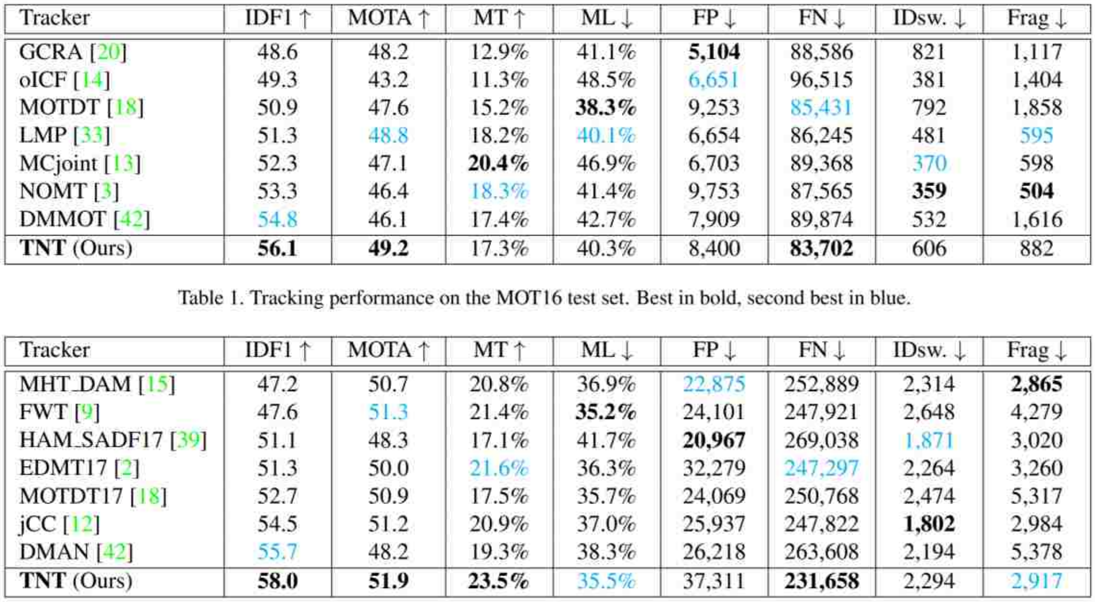


Figure 3-5 The TrackletNet Tracker (TNT) network structure [118]

The TrackletNet Tracker (TNT) network is used for SCT part in the MCCTRI framework [118]. The TNT network is a tracklet graph-based model, including two key components: 1) tracklet generation and 2) graph connectivity calculation and clustering, as shown in the first part of figure 3-5, the TrackletNet Tracker (TNT) network structure. With detection results from Yolo v3 in each frame, the tracklets are generated based on the intersection-over-union (IOU) and the appearance similarity between continuous frames. At the same time, each generated tracklet is used as a node in the graph model. The edge weights of the graph are calculated based on the connectivity of tracklets and represent the similarity of the two tracklets belong to the same vehicle. A classifier based on deep CNN, called TrackletNet is built to calculate the connectivity of two tracklets, which measures and quantifies both temporal and spatial features in the likelihood estimation. Based on the classifier's results, clustering is used to merge the same object in different nodes into a group and minimize the cost of the whole graph. Based on the researcher's experiment results, TNT network is robustness, especially in dealing with some occlusions scenarios and even congestion scenarios. The convolution kernels of the TNT Net have a strong ability to capture the temporal dependency, which significantly reduces the lost target of the tracking. Also, the TNT network enlarges and connects the object from one node into a group. The comparison of the TNT network with other advanced SCT methods based on the MOT16 and MOT17 are summarized in figure 3-6, the TrackletNet Tracker (TNT) network performance comparison. No doubt that this method performs the best IDF1 result, which is very helpful in this research scenario.



Tracker	IDF1 ↑	MOTA ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDsw. ↓	Frag ↓
GCRA [20]	48.6	48.2	12.9%	41.1%	**5,104**	88,586	821	1,117
oICF [14]	49.3	43.2	11.3%	48.5%	6,651	96,515	381	1,404
MOTDT [18]	50.9	47.6	15.2%	**38.3%**	9,253	85,431	792	1,858
LMP [33]	51.3	48.8	18.2%	40.1%	6,654	86,245	481	595
MCjoint [13]	52.3	47.1	**20.4%**	46.9%	6,703	89,368	370	598
NOMT [3]	53.3	46.4	18.3%	41.4%	9,753	87,565	**359**	**504**
DMMOT [42]	54.8	46.1	17.4%	42.7%	7,909	89,874	532	1,616
TNT (Ours)	**56.1**	**49.2**	17.3%	40.3%	8,400	**83,702**	606	882

Table 1. Tracking performance on the MOT16 test set. Best in bold, second best in blue.

Tracker	IDF1 ↑	MOTA ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDsw. ↓	Frag ↓
MHT_DAM [15]	47.2	50.7	20.8%	36.9%	22,875	252,889	2,314	**2,865**
FWT [9]	47.6	51.3	21.4%	**35.2%**	24,101	247,921	2,648	4,279
HAM_SADF17 [39]	51.1	48.3	17.1%	41.7%	**20,967**	269,038	1,871	3,020
EDMT17 [2]	51.3	50.0	21.6%	36.3%	32,279	247,297	2,264	3,260
MOTDT17 [18]	52.7	50.9	17.5%	35.7%	24,069	250,768	2,474	5,317
jCC [12]	54.5	51.2	20.9%	37.0%	25,937	247,822	**1,802**	2,984
DMAN [42]	55.7	48.2	19.3%	38.3%	26,218	263,608	2,194	5,378
TNT (Ours)	**58.0**	**51.9**	**23.5%**	35.5%	37,311	**231,658**	2,294	2,917

Figure 3-6 The TrackletNet Tracker (TNT) network performance comparison [118]

3.3 MULTI-CAMERA RE-ID & RE-RANKING

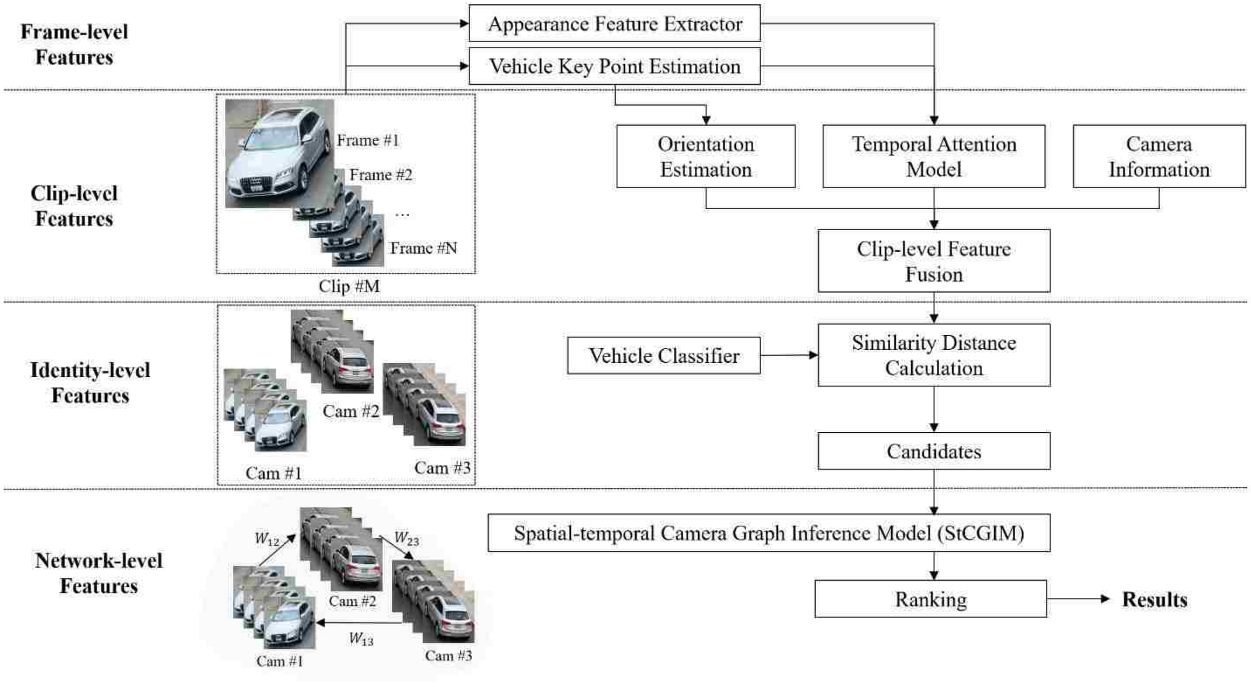
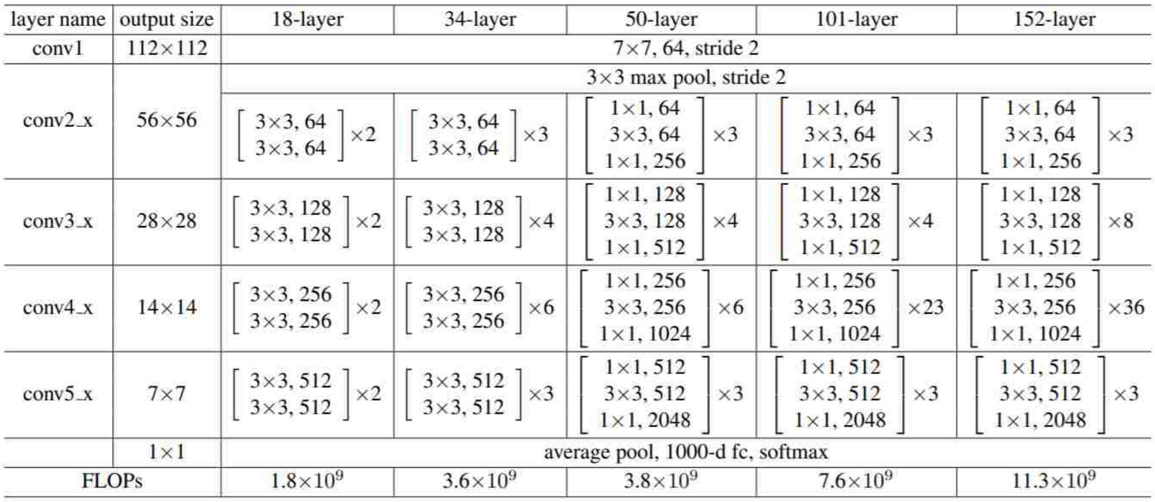


Figure 3-7 The vehicle Re-ID method architecture of MCCTRI

In MCCTRI, researchers adopted and improved the cutting-edge vehicle-Re-ID method proposed by [12]. Not only the frame level of features and clip level of features are integrated into the Re-ID framework, but also the identity level and the network level of features are combined. Four levels of features are included in the Re-ID process: frame-level features, clip-level features, and identity-level features, and network-level features are summarized into the cross camera Re-ID process, as shown in the figure 3-7 the vehicle Re-ID method architecture of MCCTRI.

3.3.1 *Frame-level & Clip-level Feature Extraction*

3.3.1.1 Appearance Feature Extractor



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4_x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3-8 The structure of ResNet50 [49]

To reduce the background noise, all the vehicle images detected by the first step are used to remove the background based on a well-trained Mask-RCNN [50]. Then the output will be sent into the image quality check process to filter the unsuitable frames. After that, tor the frame-level feature extraction, the first step is the appearance feature extraction. In this step, the vehicle image frames captured by SCT are passed into a well-trained ResNet50 based on ImageNet [49]. The architecture of ResNet50 is showing in figure 3-8. After testing, the researcher using the 2048 dimension fully connected layer vector represent the number of frames) to represent the appearance feature of each frame. This layer contains a lot of information about the image and will be used as the input for the Temporal-Attention Model.

3.3.1.2 Vehicle key-point and Orientation Estimation

In the multi-camera tracking system, the vehicles captured by the same camera are likely to driving towards different directions. In order to distinguish the traffic flow in different directions and extract the lane-level traffic information, vehicle orientation estimation is an important part of this study.

Orientation estimation can be achieved by pose estimation was first proposed by [120]. However, the original target was human. In human pose estimation, the key point detection and skeleton graph estimation are two fundamental parts. Through the human skeleton, a graphic description of a person's orientation can be estimated. Essentially, a skeleton is a set of coordinate points that can be connected to describe the person's pose. Each coordinate point in the skeleton is called a "part" (or joint, key point). A valid connection between the two parts is called a "pair" (or limb).

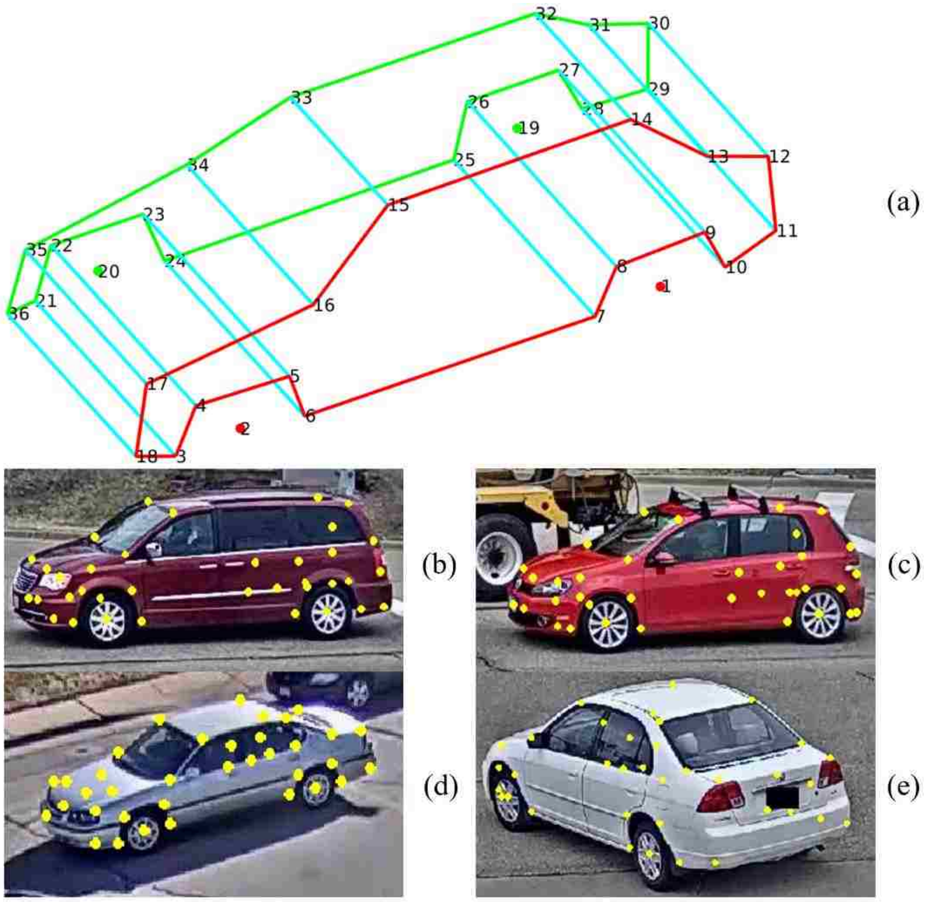


Figure 3-9 The 36-vehicle key-point estimation and example visualization

In this study, the estimation of vehicle orientation is also carried out with reference to the idea of human pose estimation. There are three steps in total: the first step is to detect the key points of the vehicle body, the second step is to locate the key plane, and the third step is to estimate the driving direction.

The first step is vehicle key points estimation. Vehicle key point features can represent the structural and appearance characteristics of vehicles, and have been widely used in vehicle detection, vehicle features extraction and scene reconstruction. [120] using a comprehensive data set constructed from a 3D CAD model of a vehicle, trained a 36-car key point positioning neural network based on a stacked hourglass architecture. This method of vehicle key point estimation can provide position information and confidence probability of each key point. After experiments, researcher found that the information contains the structural and positional relationship of a vehicle and provides a reliable input for us to estimate the orientation of the vehicle. The figure 36-vehicle key-point estimation and example visualization shows the examples of the procedure.

After 36 key points have been estimated, the next step is to determine the driving orientation of the vehicle. First, due to the different confidence probabilities of vehicle key points detection, the researchers chose two planes consisting of eight points as reference planes for driving direction determination. These two principal planes are the plane S_1 consisting of points #24, #34, #16, #6, and the plane S_2 consisting of points #10, #13, #31, #28. The reason for choosing these 8 points is 1) the average accuracy of key-points estimation is high, 2) the determination of driving direction is relatively straight forward. After the determination of the planes, S_1 and S_2 is completed, starting from the smallest node (point #6 of S_1 and point #10 of S_2), using the right-hand rule to determine the position of the normal vector of the plane. Then, according to the angle between the normal vector and the horizontal vector of the plane S_1 and S_2, the value of α and β can be calculated. Then, researchers use the value of α and β to determine the vehicle orientation as the input of next step. Besides, the ranges of α and β are: $0° \leq \alpha \leq 360°$ and $0° \leq \beta \leq 360°$. The figure 3-10 vehicle orientation angle estimation illustrates the whole process.

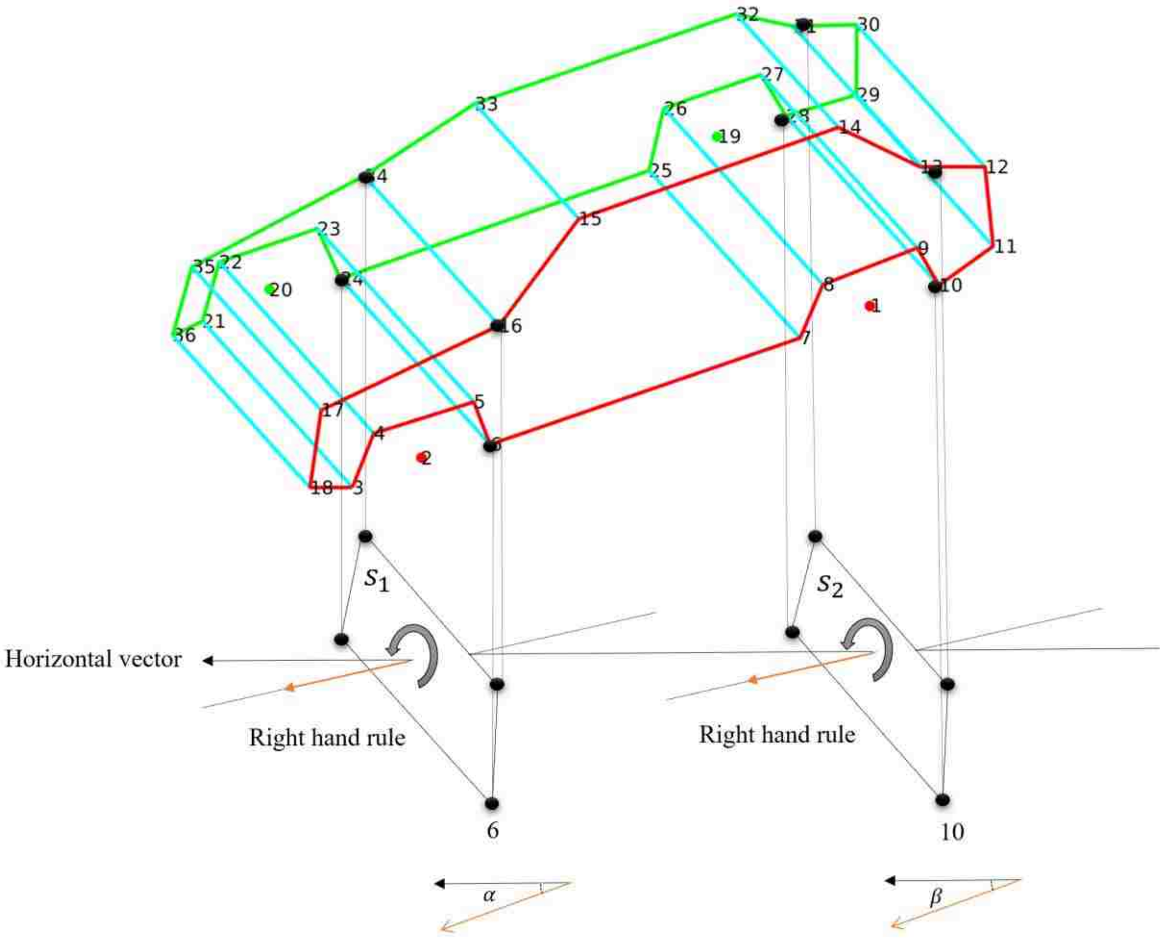


Figure 3-10 Vehicle orientation angle estimation

3.3.1.3 Temporal-attention Model （3）

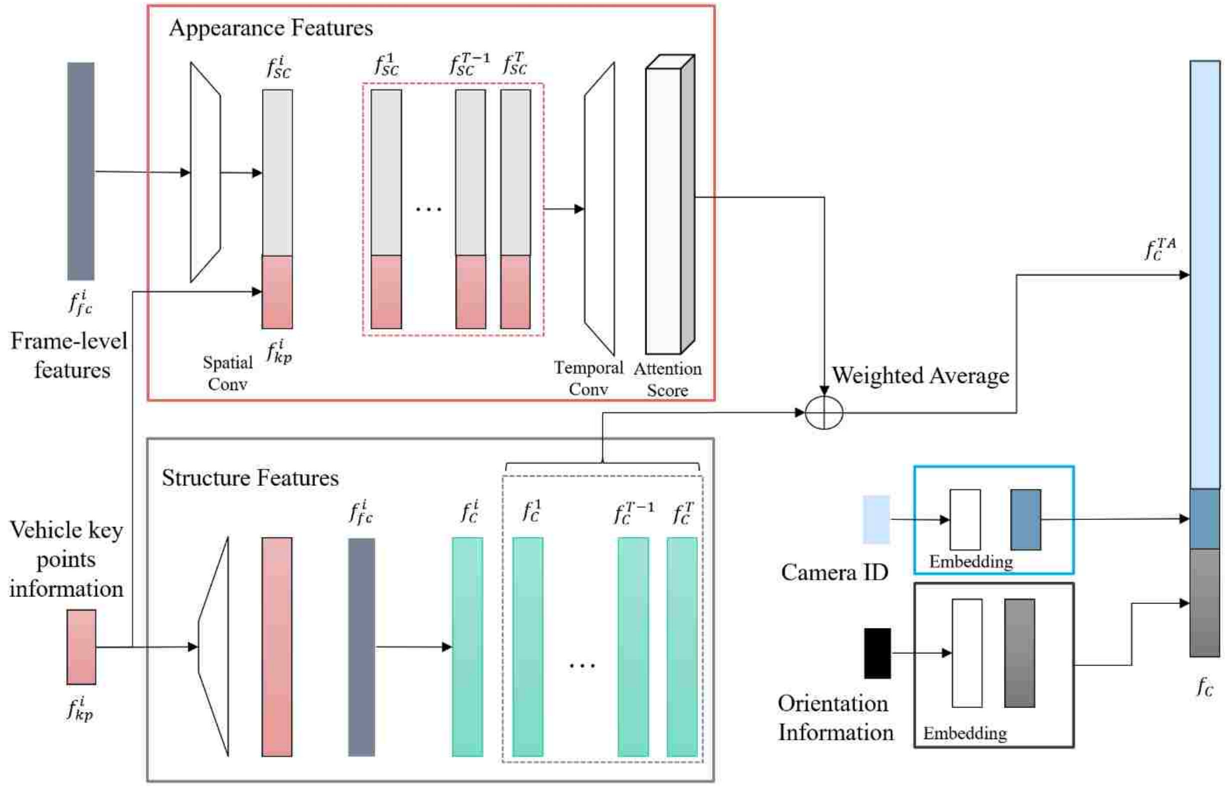


Figure 3-11 The Temporal-Attention model structure for Clip level of features fusion

After the frame-level features, including appearance features and structure features, the next step is to summarize and fuse the frames from the same clip into the clip level features. Here, a temporal-attention model [12] [121] is adopted to carry out the task.

There are two parts in the temporal-attention model, which used to fuse the appearance features and structure features. The whole model architecture shows in figure 3-11. For the appearance features part, the 2048 dimension fully-connected layer of the ResNet50 f_{fc}^{i} is used as vehicle appearance features. A 2-D convolutional neural network is used to capture the spatial features (f_{SC}^{i}). In a clip, T frames spatial features, from f_{SC}^{1} to f_{SC}^{T} are used as input of a 1-D convolutional neural network to extract the temporal information. The two convolution networks are trained to get reliable frame attention scores in different video clips. For the structure features set, researchers transfer the 2D image into a 3D format to show the spatial relationship further based on the 36 car key points estimation results (f_{kp}^{i}). Researchers use the 36 car key points to calculate the area of 18 surface, and then combined with the f_{fc}^{i} to represent the vehicle structure features. Finally, the weighted average of frame scores f_{C}^{TA} is used as a main part of the clip-level feature of a vehicle.

3.3.1.4 Clip-level Feature Fusion (2)

The output of the temporal-attention model f_{C}^{TA} combined the clip appearance features and vehicle structure features. However, for MTMCT problem, the camera information and camera orientation is necessary for the vehicle Re-ID process. The orientation features input is the angel value of α and β. For a better expression of the angel α and β in the neural network, researchers add an embedding layer to mapping the value of α and β into a 64-dimension vector. And then concatenate the vector with the f_{C}^{TA}. Except for the orientation information, the camera information is also very important in daily life, including the camera ID, location and installation height. In this model, for better use of the camera connectivity information of a road network, the camera ID are added into the clip level of features to represent the clip features with the camera attributes. Also, for a better understanding the ID features in the neural network, a binary embedding was made and expand the whole camera ID domain into a 32-dimension vector. Then, this part is concatenated with the orientation vector. The output of the temporal-attention model, the orientation information and the camera information consist of the whole clip level of features (f_{C}).

3.3.1.5 Loss Function Design （2）

In order to achieve the human face recognition task, the triplet loss was first proposed by Google, FaceNet [122]. Google researchers have proposed a new vector representation for training human faces through online triplet loss. In the field of supervised machine learning, there are usually fixed categories. In that situation, researchers can use the SoftMax-based cross-entropy loss function for training. However, sometimes, the category is a flexible domain. The strict margin categories is not suitable in that case (such as the same people captured by different cameras). At that time, the triplet loss can solve the problem. In face recognition and Quora question pair tasks, triplet loss has the advantage of distinguishing details, not only from category to category, but also from a domain to another. That is, when two inputs are similar, triplet loss can better model the details difference. Generally, triplet loss is widely used in training transformations from an input space (such as words, images) to an embedding feature space. So, the embedding features measured by the Euclidean distance are optimized based on the training process.

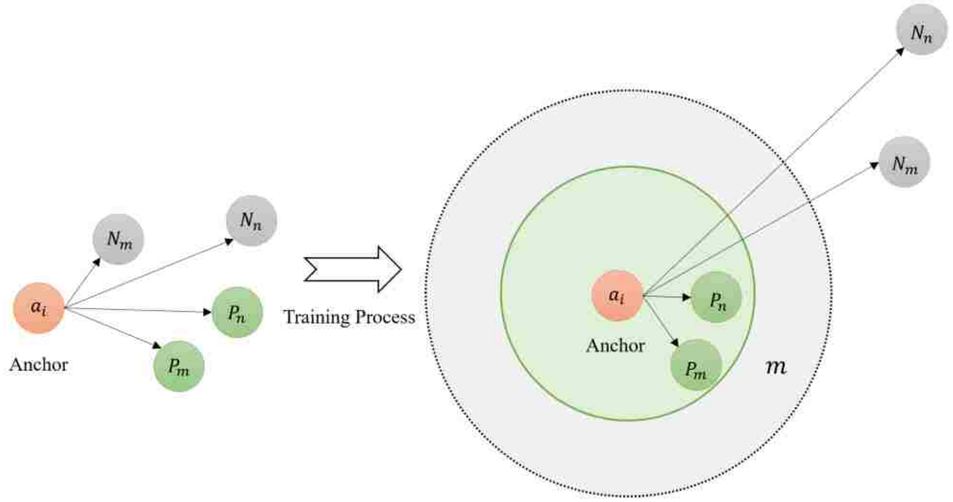


Figure 3-12 Illustration of the triplet loss function training process

As figure 3-12 shows, assuming the i input vehicle as an anchor a_i, the same vehicle should by placed at the positive sample features position, using P_m and P_n. Moreover, the negative sample features, which are not the same one, are called N_m and N_n. Using m to represent the margin area in the equation. Based on the previous work finished by [12], during the training process, the author replaced the batch sample (BS) approach [14] instead of the batch hard (BH) approach in triplet generation. During the training process, the anchor-to-sample distances of BS data sampling is based on multinomial distribution. The obvious advantage is of the replacement can be summarized into two parts 1) speed up the training process; 2) improve the robustness of the model. Use χ to represent a mini-batch sample. Furthermore, the mathematical equation can be defined as:

$$\zeta_{BS_{Tri}}^{i} = \sum_{B} \quad \sum_{a_i \in B} l_{Tri}(a_i) \tag{1}$$

And

$$l_{Tri}(a_i) = \left|\left(\sum_{P_m \in P(a_i)} w_{P_m} d_{a_i P_m} - \sum_{N_m \in N(a_i)} w_{N_m} d_{a_i N_m} + m\right)\right| \tag{2}$$

In equation (2), w_{P_m} and $w_{N_m}$are the weight of the positive sample and negative sample. The $d_{a_i P_m}$ and $d_{a_i N_m}$ are the distance of the anchor a_i to the sample position. The weight of w_{P_m} and w_{N_m} are defined in the following:

$$w_{P_m} = P\left(x_p == \ multinomial_{\ x \in P(a_i)}\{D_{a_i x}\}\right) \tag{3}$$
$$w_{P_n} = P\left(x_N == \ multinomial_{\ x \in N(a_i)}\{D_{a_i x}\}\right) \tag{4}$$

In the above equations, the x_p and x_N are positive and negative samples.

Except for the triplet loss, the cross-entropy loss is also included into the model. Cross-entropy loss are always used to measure the performance of a classification model. The equation of the loss is in the following:

$$\zeta_{Xent}^{i} = -\sum_{i=1}^{p} \log\left(p(i)q(i)\right) \tag{5}$$

In the equation, the $q(i)$ is the ground truth label, $p(i)$ is the probability of the probe image belongs to vehicle i.

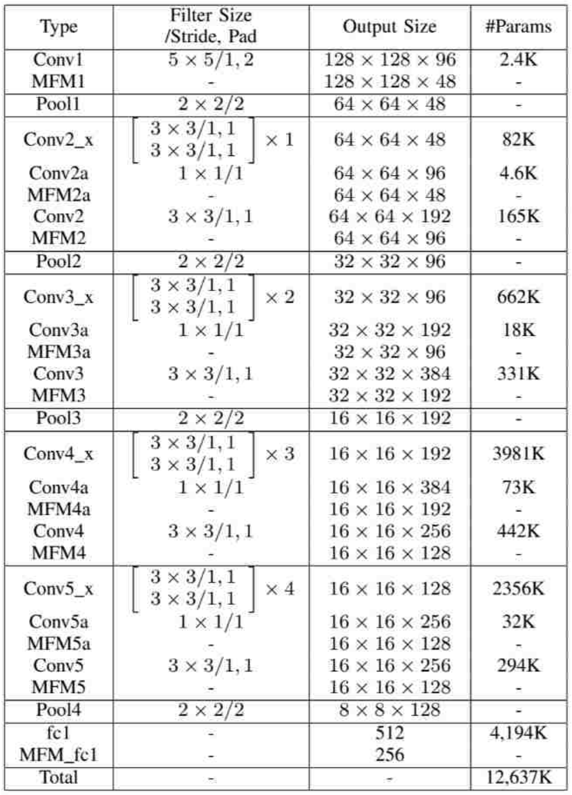
Moreover, the overall loss function is:

$$\zeta = \lambda \zeta_{BS_{Tri}}^{i} + (1-\lambda)\zeta_{Xent}^{i} \tag{6}$$

3.3.2 *Identity-level Features Extraction*

After extracting the appearance features of different frames and the structural features of the vehicle through a deep neural network, the temporal-attention model was fused to generate clip-level features. Clip-level features represent the feature set of a car at a particular camera view. In other words, this feature set represents s vehicle at a camera (the orientation relationship between the camera and the vehicle) and at the light condition (the time passing the camera). However, when a vehicle passed through multiple cameras in different orientations and different lighting conditions, the clip-level features is not enough to Re-ID the same vehicle. Currently, researchers usually complement the clip-level of features with individual-level features. These individual-level features are usually the inherent attributes of the vehicle and always unchangeable, including vehicle category features (for example: SUV, sedan, truck, etc.), color features (black, white, silver, etc.), brand features such as (BMW, Audi, Lexus, etc.) and even model features (such as Audi A4, A5, A6, etc.) and so on. Based on the identity-level of information, people can better integrate vehicle information from different road segments (such as intersections or roadsides) and different orientations (such as front and rear perspectives) to better identify whether they are the same or different vehicles.

In the current research, individual-level feature extraction relies heavily on deep learning. In this research, the author adopted the method proposed by [123] named Light CNN. Light CNN-29 is a CNN-based deep learning architecture. It was mainly proposed for human face recognition. The advantage of the lightweight CNN-29 framework is that it is faster and more efficient than other published CNN methods. In this method, the Max-Feature-Map (MFM) operation can obtain a compact, low-dimensional and efficient face representation feature set quickly. Smaller kernel sizes for convolutional layers, network layers, and remaining blocks in the network have been implemented to reduce parameter space and improve performance. Figure 3-13 illustrates the Light CNN architecture in detail.



Type	Filter Size /Stride, Pad	Output Size	#Params
Conv1	$5 \times 5/1, 2$	$128 \times 128 \times 96$	2.4K
MFM1	-	$128 \times 128 \times 48$	-
Pool1	$2 \times 2/2$	$64 \times 64 \times 48$	-
Conv2_x	$\begin{bmatrix} 3 \times 3/1, 1 \\ 3 \times 3/1, 1 \end{bmatrix} \times 1$	$64 \times 64 \times 48$	82K
Conv2a	$1 \times 1/1$	$64 \times 64 \times 96$	4.6K
MFM2a	-	$64 \times 64 \times 48$	-
Conv2	$3 \times 3/1, 1$	$64 \times 64 \times 192$	165K
MFM2	-	$64 \times 64 \times 96$	-
Pool2	$2 \times 2/2$	$32 \times 32 \times 96$	-
Conv3_x	$\begin{bmatrix} 3 \times 3/1, 1 \\ 3 \times 3/1, 1 \end{bmatrix} \times 2$	$32 \times 32 \times 96$	662K
Conv3a	$1 \times 1/1$	$32 \times 32 \times 192$	18K
MFM3a	-	$32 \times 32 \times 96$	-
Conv3	$3 \times 3/1, 1$	$32 \times 32 \times 384$	331K
MFM3	-	$32 \times 32 \times 192$	-
Pool3	$2 \times 2/2$	$16 \times 16 \times 192$	-
Conv4_x	$\begin{bmatrix} 3 \times 3/1, 1 \\ 3 \times 3/1, 1 \end{bmatrix} \times 3$	$16 \times 16 \times 192$	3981K
Conv4a	$1 \times 1/1$	$16 \times 16 \times 384$	73K
MFM4a	-	$16 \times 16 \times 192$	-
Conv4	$3 \times 3/1, 1$	$16 \times 16 \times 256$	442K
MFM4	-	$16 \times 16 \times 128$	-
Conv5_x	$\begin{bmatrix} 3 \times 3/1, 1 \\ 3 \times 3/1, 1 \end{bmatrix} \times 4$	$16 \times 16 \times 128$	2356K
Conv5a	$1 \times 1/1$	$16 \times 16 \times 256$	32K
MFM5a	-	$16 \times 16 \times 128$	-
Conv5	$3 \times 3/1, 1$	$16 \times 16 \times 256$	294K
MFM5	-	$16 \times 16 \times 128$	-
Pool4	$2 \times 2/2$	$8 \times 8 \times 128$	-
fc1	-	512	4,194K
MFM_fc1	-	256	-
Total	-	-	12,637K

Figure 3-13 Illustration of the Light CNN architecture [123]

In this study, the researcher selected 4 characteristics as individual-level characteristics, which are 8 vehicle types, 36 brands, 11 color characteristics, and 4 manufacture year characteristics. These feature sets and corresponding figures are used to train light CNN-29. After obtaining a well-trained Light CNN, a fully connected layer will map the output to 2048 dimension. This 2048-dimension vector is used as the identify level feature. The details about the dataset will be introduced in Chapter 4.2, data set description. The figure 3-14 illustrates the identity-level extraction based on Light-CNN 29 process.

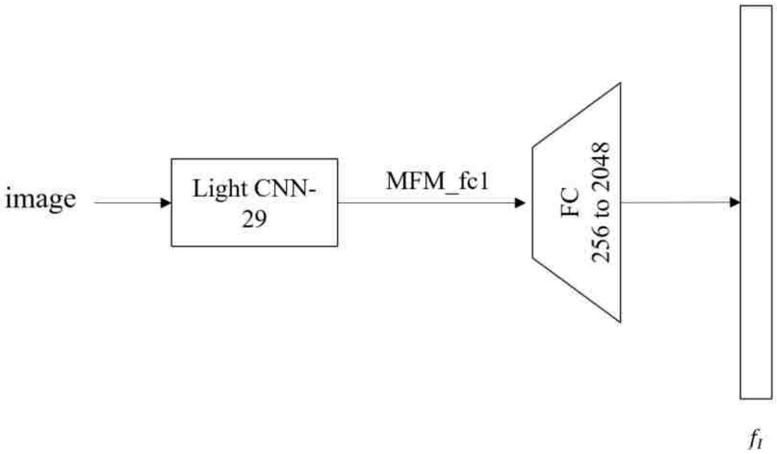


Figure 3-14 Illustration of the identify level of features

3.4 CANDIDATES SELECTION

After the identity level of features and clip level of features are obtained from the former procedures, the first round of Re-ID candidates is calculated based on the equation (7). Where the $d(x, y)$ is the cosine similarity of two vector. The C_n is the categories of the identity-level of features. $x_{i,}$ y_i are the query and the gallery target.

$$d_{IC}(x_{i,}y_i) = \sum_n C_n \, d_{f_I}(x_{i,}y_i) + \alpha d_{f_C} \, (x_{i,}y_i) \qquad (7)$$

3.5 SPATIAL-TEMPORAL CAMERA GRAPH INFERENCE MODEL (STCGIM)

The focus of this research is how to Re-ID vehicles across multiple cameras installed in the road network at different locations. Therefore, in addition to the frame-level features, clip-level features and identity-level features mentioned in the previous section, the spatio-temporal information and connectivity information contained in the camera network and road network are also essential information sources that can effectively assist cross-camera vehicle Re-ID. The road network composed of roads and the network of cameras installed on the road forms two highly overlapping networks. In these two networks, vehicles are photographed by different cameras at different locations and at a time point. In this section, the researcher discussed in detail how to construct a road network connectivity map that includes both road network relationship and camera network relationship and serves the vehicle Re-ID neural network.

3.5.1 *Network Graph Extraction*

In this research, there are two major networks, one is the road network $G(R)$, and another is camera network $G(C)$. The $G(R)$ represents the road network, which includes the road connectivity information, consists of intersections and road sections. At the same time, the cameras are installed at different locations in the road network. In this task, using the $G(T)$ to represent our target research network.

The next step is to fuse the camera surveillance area and the road segment into the whole research graph $G(T)$. Each camera C_i is installed at a road section R_i. To better extract the lane-level of microscope information from the video, the author added a new zone called camera loop L_{ij} (i represent the camera loop belongs to the camera C_i, and j is the index number of camera C_i). Inspired by the traditional loop detector algorithms, the camera loop is a small rectangular area on each lane.

After the vehicle detected by the detector, each vehicle is bounded by a bounding box. This bounding box will coincide with the area of the camera loop and the overlapping area is a function of time, using $F_{overlapping} = f(t)$. Based on the change of this overlapping part area, research can extract the camera loop that each car passes in each camera, and then further extract the lane level spatial information. With the information, the attributes can be added to each passing vehicle, such as straight, left-turn and right-turn, driving direction, and lane conditions. Also, a more detailed graph based on traffic lane-level can be established based on the real road network constraint.

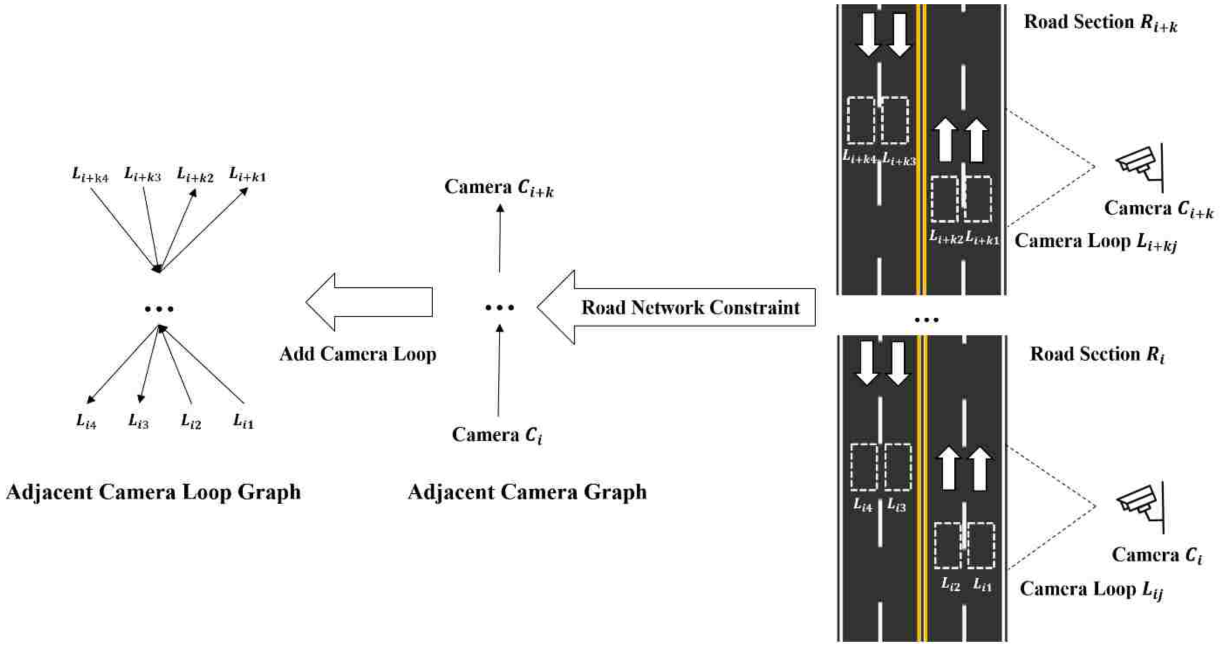


Figure 3-15 Adjacent camera graph and adjacent camera loop graph establishment

Five steps are necessary to build a camera graph inference model in the following, and the illustration of the steps shows in figure 3-15:

1. Select the research target graph $G(T)$.
2. Divided the $G(T)$ into multiple road sections from R_i to R_{i+k}, each road section is covered by one or more cameras C_{ij} to $C_{(i+k)j}$.

$$G(T) = \{R_i \ldots R_{i+k}\} \cup \{C_{ij} \ldots C_{(i+k)j}\} \quad (8)$$

3. According to the perspective view of each camera from C_i to C_{i+k}, determine the camera loop for each camera. For each camera C_i,

$$C_i = \{L_{i1} \ldots L_{ij}\} \tag{9}$$

4. Select the neighbor camera and then build the adjacent camera graph. The adjacent camera means that the vehicles are possible to be driven from a camera directly to another without passing the third one. If there are multiple routes, choose the shortest route or analysis case by case.
5. Based on the adjacent camera graph, build the adjacent camera loop graph. Then select the adjacent camera loop pairs as the cross-camera search index graph.

3.5.2 *Trajectories Extraction*

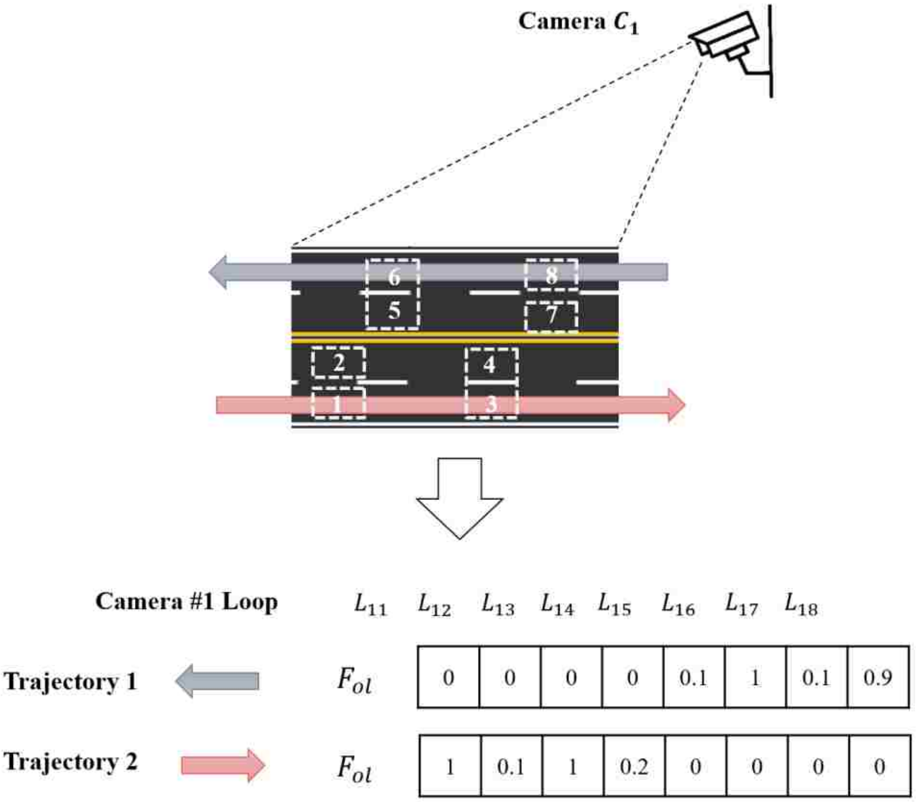
In the same camera view, especially in the area near the intersection, there are often many vehicles moving towards different directions. In order to better assist the camera loops to distinguish trajectory and match the vehicle across cameras, the author uses the camera loop sequence to distinguish trajectories in the same camera further. With a sequence of camera loop overlapping factor F_{ol}, the researcher can determine the trajectory of each vehicle and distinguish different directions.

The calculation process of the camera loop overlapping factor F_{ol} is as follows. After each vehicle has passed its own camera, the bounding box of the vehicle will have a specific overlapping area with the camera loop area defined by the researcher. The size of this overlapping area is a function related to time t. Moreover, the characteristic of this function is to increase first and then decrease. Therefore, when calculating the overlapping factor F_{ol} for each camera loop area, the researchers take the maximum value of $F_{olL_{ij}}$ as the value of this function, which is used to distinguish the size of the overlapping area of the vehicle from different camera loops, and then to distinguish different trajectory features.

$$F_{olL_{ij}} = Max(R_{olL_{ij}}(t)) \qquad (10)$$

$$R_{olL_{ij}}(t) = \left| \frac{A_{L_{ij}} \cap A_{Vb}(t)}{A_{L_{ij}}} \right| \qquad (11)$$

Furthermore, the $R_{olL_{ij}}(t)$ is the ration of overlapping area, which equals to the overlapping area of the camera loop ($A_{L_{ij}}$) and the vehicle bounding box (A_{Vb}) over the camera loop ($A_{L_{ij}}$) area. The illustration figure of the trajectory extraction of camera #1 shows in figure 3-16.



Camera #1 Loop		L_{11}	L_{12}	L_{13}	L_{14}	L_{15}	L_{16}	L_{17}	L_{18}
Trajectory 1	F_{ol}	0	0	0	0	0.1	1	0.1	0.9
Trajectory 2	F_{ol}	1	0.1	1	0.2	0	0	0	0

Figure 3-16 Vehicle trajectory extraction based on camera loop graph

With the camera loop sequence, the trajectory of each vehicle can be separated and distinguished. After that, through the cross-camera graph model matching, the researcher can get the trajectory matching relationship between different cameras, and then perform more accurate cross-camera Re-ID candidates.

3.5.3 *Camera Loop Link Model Establishment*

Since today's traffic monitoring systems often need to cover multiple areas with large sections and many roads, the accuracy required for information is high, and a single camera is difficult to handle. A single-camera is often limited by the **F**ield **O**f **V**iew (**FOV**) and can only cover a limited range, so the current monitoring system often uses multiple fixed FOV cameras to complete the monitoring and information extraction work. Based on this assumption, how to obtain a reliable camera feature link model from a large number of training videos has become a vital issue.

Therefore, in this study, the researcher designed a unique framework to unify, transfer, and link features among different cameras. These characteristic parameters include travel time, and vehicle direction and trajectory matching across cameras. Here, the researcher combines each feature with the graph architecture of the specific network being studied and targets the overall goal as a weighted optimization problem. Here, the author uses X and Y to represent the vehicle passing two different cameras C_i and C_j. And then optimize the weight based on the training set and the tuning results.

3.5.3.1 **T**ravel **T**ime **E**stimation (**TTE**) Constraint

Cross-camera vehicle travel time is a key factor for Re-ID vehicles. Combining the length of the specific road section and the speed limit, the approximate travel time can be estimated, and possible vehicles are filtered within the time period and then further compared. This step can greatly reduce the number of candidate vehicles. In this research, the travel time distribution $f_{TTE}(t)$ is built based on the Gaussian kernel estimation:

$$F_{TTE}(t) = \frac{1}{\sigma_{ij}\sqrt{2\pi}} \sum_{i=1}^{N} \exp\left(-\frac{(t-t_{ij})}{2{\sigma_{ij}}^2}\right) \tag{12}$$

In the above formula (12), the σ_{ij} is the variance of the travel time from camera C_i to camera C_j. The t_{ij} is the average travel time from camera C_i to camera C_j. And the constraint of camera pair C_i and C_j is evaluated by the KL distance $D_{v_i ij}$. In the equation (13), the $F_{TTE}(v_i)$ is the travel time distribution of vehicle i and the $F_{TTE}(t)_{ij}$ is the overall estimated travel time distribution of camera pair C_i and C_j. The $F_{TTE}(v_i)$ is obtained using the vehicle i's travel time as the mean value with the same σ_{ij} of the $F_{TTE}(t)_{ij}$.

$$D_{v_i ij}(F_{TTE}(v_i) \parallel F_{TTE}(t)_{ij}) = \sum_i F_{TTE}(v_i) ln \frac{F_{TTE}(v_i)}{F_{TTE}(t)_{ij}} \tag{13}$$

3.5.3.2 **T**rajectory **S**equence **C**onstraint (**TSC**)

After constructing the adjacency graph of adjacent cameras, the adjacency map and the trajectory characteristics obtained based on the camera loop sequence can be used for further filtering different vehicle trajectories and reduce the number of candidate vehicles. Here, the trajectory sequence constraint can be evaluated by the following method:

$$F_{TSC} = \sum_{n=1}^{k} \left(\left| x_{L_{in}} - y_{L_{in}} \right| + \left| x_{L_{jn}} - y_{L_{jn}} \right| \right) \tag{14}$$

Where the $x_{L_{in}}$ is the value of the camera loop sequence during the vehicle x passing camera C_i and the $y_{L_{in}}$ value of the camera loop sequence during the vehicle y passing camera C_i. Also, $x_{L_{jn}}$ and $y_{L_{jn}}$ are the same meaning as the former. Moreover, the F_{TSC} is used to evaluate the overlapping trajectory situation of two vehicles when try to filter the Re-ID candidates.

3.5.3.3 Overall Objective Function

With the value of the former two constraints extracted from the $G(T)$, the overall target function for the optimization used for the camera graph inference model is in the following:

$$T(x, y) = w_{tte} * D_{v_i ij} + w_{tsc} * F_{TSC} \tag{15}$$

And the value of w_{tte}, and w_{tsc} is the optimization target during the training process to fuse two constraints together.

3.6 TRAFFIC INFORMATION ESTIMATION

After obtaining the cross-camera vehicle tracking result, the next step is to make good use of the information. In this chapter, based on the results obtained in the previous steps, the author performed not only link travel time, speed and volume estimation, but also the detailly distribution estimation. Different from previous traditional methods, based on the results of cross-camera tracking and Re-ID with different levels of accuracy, the Accuracy Adaptive Model (AAM) has been used in this research.

3.6.1 *Accuracy Adaptive Model (AAM)*

Before the process of estimating traffic parameters, the solid use of the cross-camera tracking results is a prerequisite. For each camera and each surveillance scenario, the accuracy of detection and tracking are different. However, the traffic information estimation is based on the result generated by the multi-camera tracking result. Based on the scenario, the author mainly uses the measurement of IDs to evaluate the tracking performance. With the tracking accuracy, the author proposed a system methodology to obtain the traffic information for considering the measurement of tracking results, which are integrated of the IDF$_1$ (ID corresponding F$_1$), IDP (IDentification Precision), and IDR (IDentification Recall) of each camera scenario.

The IDP, which called Identification precision is the fraction of computed detections that are correctly identified. The mathematical formula is in the following:

$$IDP = \frac{IDTP}{IDTP+IDFP} \tag{16}$$

The IDTP is the true positive value of the IDs tracking result, which means the correct tracking result. The IDFP means the false positive results of IDs. Except, the IDR is also evaluated for each camera. The mathematical formula of IDR is:

$$IDR = \frac{IDTP}{IDTP+IDFN} \tag{17}$$

Where the IDFN is the false negative of IDs is the evaluation set. Moreover, the IDF$_1$ is:

$$IDF_1 = \frac{2IDTP}{2IDTP+IDFN+IDFP} \tag{18}$$

Here, the IDF$_1$ means the ratio of correctly identified detections results above the average of ground-truth value and computed detections for each camera scenario. For every camera view and every link, the value of IDP, IDR and IDF_1 is different.

Assume, there are N_i vehicles (SCT results) passing through camera i. And now, the MCCTRI tracks n vehicles from the camera i to camera j. Based on the evaluation sets, the SCT IDR for the camera i is R_{IDR_i}. The value of IDF_1 from camera i to camera j is α_{ij} $(0 < \alpha_i < 1)$, and the IDR from camera i to camera j is β_{ij} $(0 < \beta_i < 1)$. Then, the traffic information can be obtained based on the obtained information.

3.6.2 *Link Average Travel Time and Distribution Estimation*

With the cross-camera tracking ID set and the accuracy, the traffic link average travel time distribution $T(t)_{ij}$ can be estimated based on the following math equation:

$$T(t)_{ij} = \frac{1}{\frac{\sigma_{ij_t}}{IDR_{ij}} * \sqrt{2\pi}} \sum_{i=1}^{N} exp\,(-\frac{\left(t_{n_{ij}} - \overline{t_{\imath\jmath}}\right)}{2\sigma_{ij}{}^{2}{}_{t}}) \tag{19}$$

In the formula (19), the σ_{ij_t} is the variance of the obtained tracking results from camera i to camera j. The IDR_{ij} is the ID recall accuracy of camera pair of i and j. And the average travel time results from camera i to camera j $\overline{t_{\imath\jmath}}$ is:

$$\overline{t_{\imath\jmath}} = \frac{\sum_n t_{n_{ij}}}{n} \tag{20}$$

3.6.3 *Link Speed and Distribution Estimation*

With the MCCTRI cross-camera tracking result from camera i to camera j, the speed of each vehicle can be obtained by:

$$S_{nij} = \frac{Dis_{ij}}{t_{ij}} \tag{21}$$

Moreover, the speed distribution $S(t)_{ij}$ can be obtained based on:

$$S(t)_{ij} = \frac{1}{\frac{\sigma_{ij_s}}{IDR_{ij}} * \sqrt{2\pi}} \sum_{i=1}^{N} exp\,(-\frac{\left(S_{nij} - \overline{S_{\imath\jmath}}\right)}{2\sigma_{ij}{}^{2}{}_{s}}) \tag{22}$$

In the formula (22), the σ_{ij_t} is the speed variance of the obtained tracking results from camera i to camera j. The IDR_{ij} is the ID recall accuracy of camera pair of i and j. And the average speed results from camera i to camera j $\overline{s_{\imath\jmath}}$ is:

$$\overline{S_{\imath\jmath}} = \frac{\sum_n S_{n_{ij}}}{n} \tag{23}$$

3.6.4 *Traffic Volume and Distribution Estimation*

The MCCTRI result can also estimate the link traffic volume V_{ij}. After the testing set to evaluate each camera link, the IDF_1 from camera i to camera j is α_{ij} $(0 < \alpha_i < 1)$, and the IDR from camera i to camera j is β_{ij} $(0 < \beta_i < 1)$ can be obtained. Then, the link volume can be estimated by the following equation:

$$V_{ij} = \left(\frac{\alpha_{ij} * \beta_{ij} * n}{(2\beta_{ij} - \alpha_{ij}) R_{IDR_i}}\right) \qquad (24)$$

Where n is MCCTRI track the number of vehicles from the camera i to camera j. And the SCT result of IDR for the camera i is R_{IDR_i}. The Volume distribution can be estimated by:

$$D_{v_{ij}} = \left(\frac{V_{ij}}{N_i}\right) \qquad (25)$$

Where N_i is the number of detected vehicles passing the camera i.

Chapter 4. EXPERIMENT AND RESULT DISCUSSION

4.1 OVERALL DESIGN

In this research, the whole system is built based on the Linux system (Ubuntu 18.04.2), with a Core-i7 CPU and an NVIDIA TITAN Xp GPU. All the system settings and useful packages are open-source online, including TensorFlow, CUDA, cuDNN, OpenCV etc. Moreover, the experiment process is divided into two parts: MCCTRI model evaluation and traffic information estimation evaluation. Large-scale High-resolution Traffic Video (LHTV) Dataset are used to evaluate the framework. The dataset details are in the chapter 4.2 data description.

4.2 DATASET DESCRIPTION

4.2.1 *Large-scale High-resolution Traffic Video (LHTV) Dataset*

High-quality open-source datasets are invaluable resources for a research field. However, large-scale high-definition traffic surveillance video datasets are quite limited, especially for the multi-camera scenarios. There are several open-source videos or image-based datasets, such as City-flow [124], VeRi-776 [101]. However, these datasets are designed for computer science problems, such as vehicle detection, vehicle Re-ID and tracking, with some inevitable questions and unnecessary challenges to do the transportation-based video analysis. A high-quality dataset that enables MTMC related transportation research is needed.

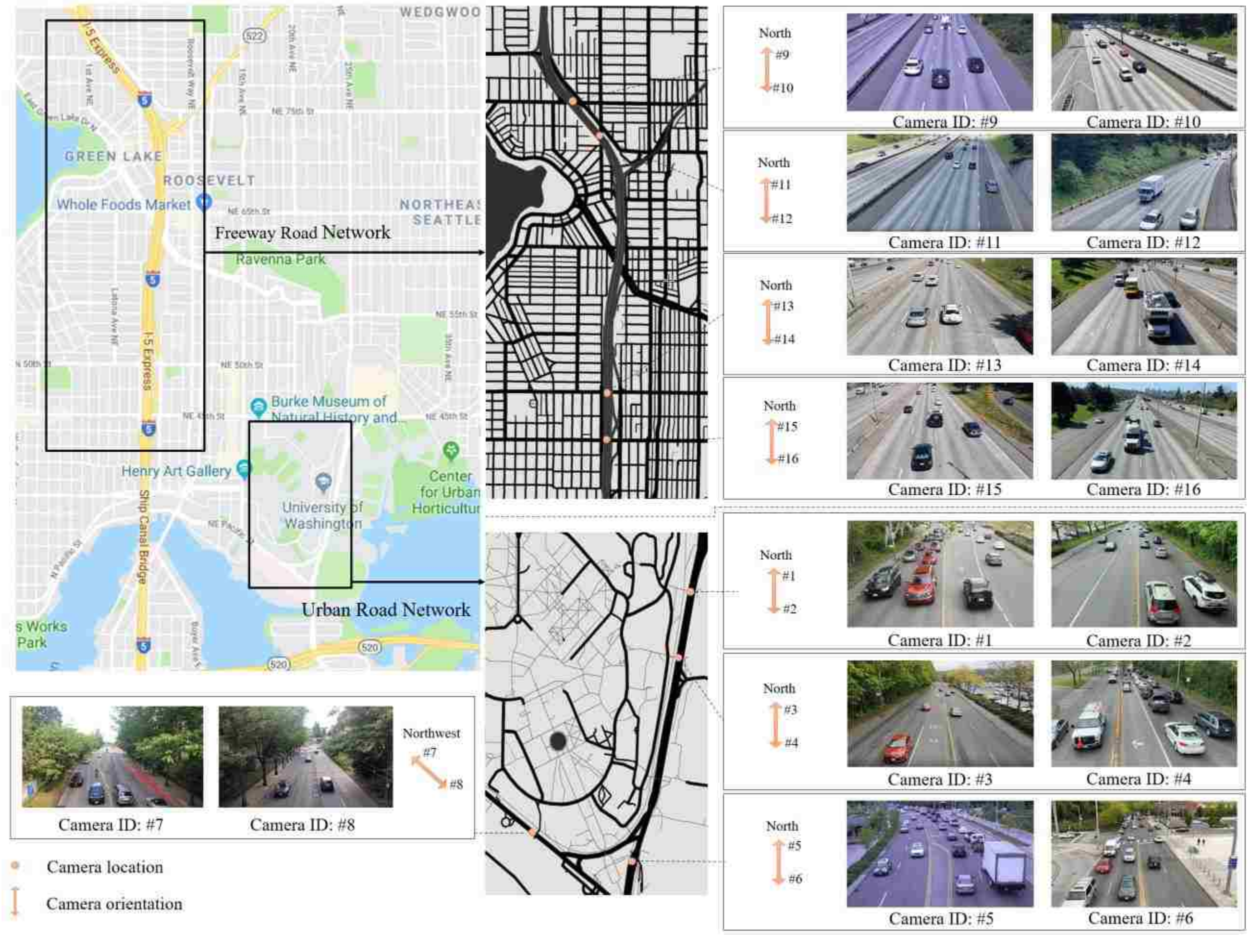


Figure 4-1 Illustration of the LHTV Dataset

LHTV dataset includes 1012.25 minutes video and includes 16 different cameras in Seattle, with the quality of 1080p resolution with 30fps. Details information is in the figure 4-1 illustration of the LHTV Dataset. Researchers labeled 76.78 min video of 6 different cameras. All the video includes time and location information. The longest distance between two cameras is 3.2 km, and the shortest one is 7.6 m. In this research, the author used the six labeled videos from no #9, #10, #12, #13, #14, #16 to train, evaluate and test the MCCTRI. 36.75 min video is used to train the model, and 20.01 min video, including five cameras (#9, #10, #12, #13, #14) are used to evaluate and 20.03 min video are used to test the MCCTRI for multi-camera tracking and traffic information estimation. For each evaluation and testing, five cameras are video are spited into multiple 1-min video clips, including 1800 frames and then finish the multi-camera tracking task. Besides, the author also used the Cityflow dataset [124] to train the model.

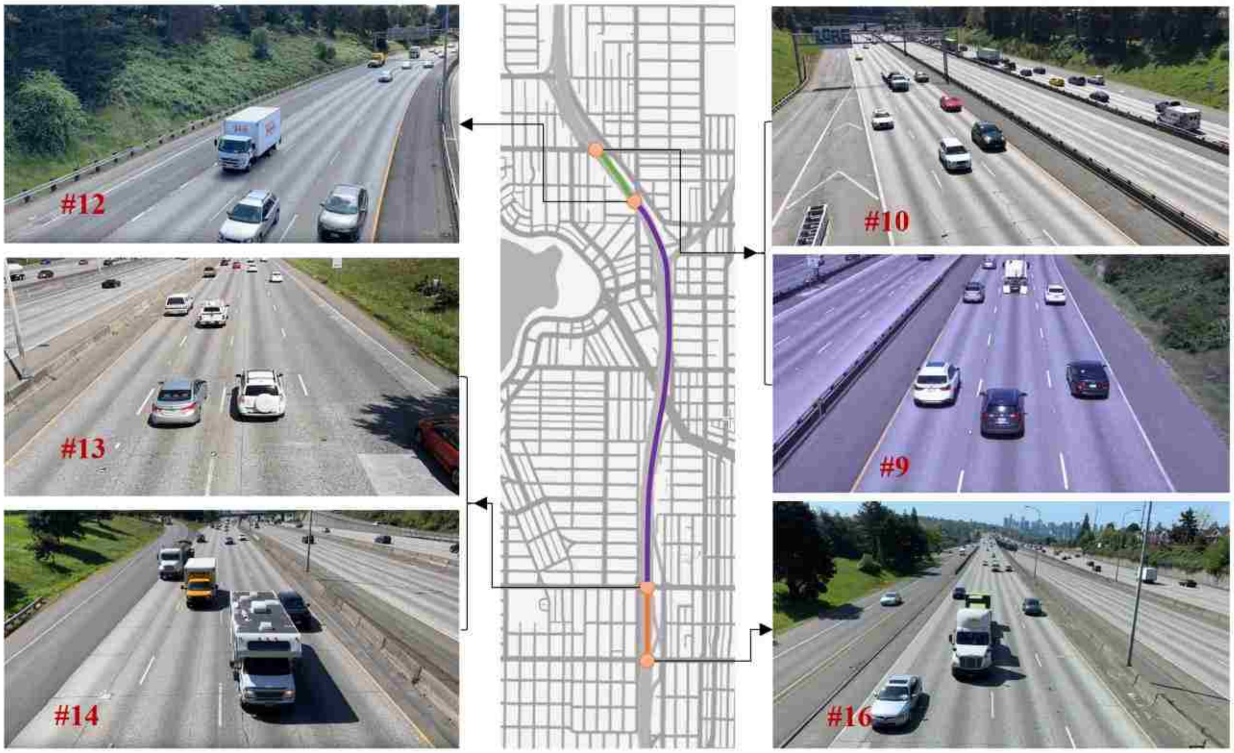


Figure 4-2 Visualization of six cameras used in the training and evaluation

In this study, the researcher selected four characteristics as vehicle individual-level characteristics, which are eight vehicle types (including SUV, sedan, minivan, hatchback, pickup truck, truck, bus and others), and 36 brand types. (Acura, Audi, BMW, Buick, Cadillac, Chevrolet, Chrysler, Dodge, Ford, GMC, Honda, Hyundai, Infiniti, Jaguar, Jeep, Kia, Land Rover, Lexus, Lincoln, Mazda, Mercedes-Benz, Mercury, Mini, Nissan, Pontiac, Porsche, Ram, Saab, Saturn, Scion, Subaru, Suzuki, Toyota, Volkswagen, Volvo and other), 11 color features (red, blue, yellow, gray, silver, black, green, dark green, white, gold and other), four year characteristics (before 2000, 2000-2010, 2010-2015, after 2015). The author used the vehicle cropped images from the LHTV dataset and Cityflow dataset [124] to train and evaluate the Light-CNN and then integrated into the MCCTRI framework.

4.3 MOD & SCT Results Summary

4.3.1 *MOD Result Summary and Visualization*

Here, the author used a well-trained YOLOv3 based on the COCO dataset and finetuned on the LHTV and Cityflow dataset. In order to avoid interference caused by unnecessary categories, three categories are the detection target: car, bus, and truck. The author filters the detection result by setting a confidence probability greater than 0.5. Since the MOD result is served for the SCT process, the connectivity of the object detection among different frames is essential. Considering the overall detection performance, the researcher sets a detection zone for each camera. In each zone, the object size, features, and frame connectivity are reliable to use as the SCT input. Visualization examples can be found in figure 4-3 MOD result examples visualization.

The multi-object detection results are summarized in table 4-1 MOD result summary. The YOLOv3 multi-object detection performs well on the LHTV dataset. Achieved an average accuracy of precision 0.94 and recall 0.91 in the five evaluated cameras. The camera with the maximum detection recall is 0.93, and the minimum is 0.90.

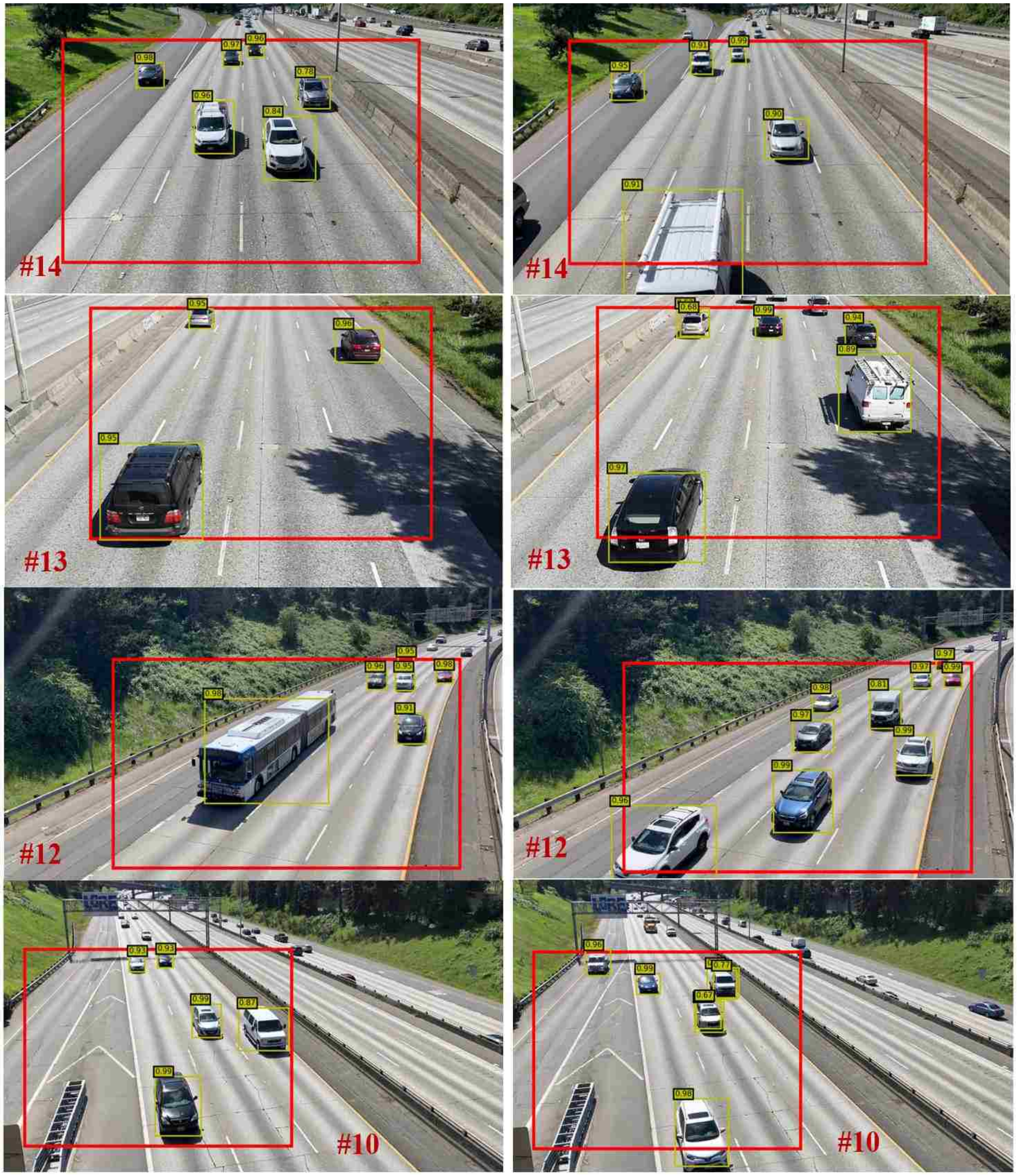


Figure 4-3 MOD result examples visualization

Table 4-1 MOD result summary

Dataset	Cameras used	Precision	Recall	Max Recall	Min Recall
LHTV	6	0.9379	0.9112	0.9277%	0.8998%

4.3.2 *SCT Result Summary and Visualization*

For the SCT process, firstly, the author set the target tracking area in each scene based on the detection zone for each camera view based on chapter 4.3.1. The objects in each area usually has the appropriate object size, less occlusion, and relatively stable detection results. According to the bounding box of the image detection and the delineated single target tracking area, the researcher deleted the bounding box with the area overlapping area ratio less than ρ ($\rho = 0.5$). Secondly, the researcher adjusted the parameters of each scene according to the model parameters of TNT to ensure the accuracy of the single-camera tracking results. The single-target camera tracking results are summarized in the following. The IDF1 accuracy of 83.35% was obtained. Among them, camera #9 has the highest IDF_1 accuracy of 86.55%, and camera #14 has the lowest IDF_1 of 79.21%. The average IDR of the eight cameras is 85.12%, and the IDP is 82.35%. Table 4-2 shows the SCT result summary, and figure 4-4, 4-5 shows the visualize the frames examples of SCT on camera #14, #13, #12 and #10.

Table 4-2 SCT result summary

Dataset	Cameras used	IDF_1	IDR	IDP	Max IDF_1	Min IDF_1
LHTV	6	**83.35%**	85.12%	82.35%	**86.55%**	79.21%

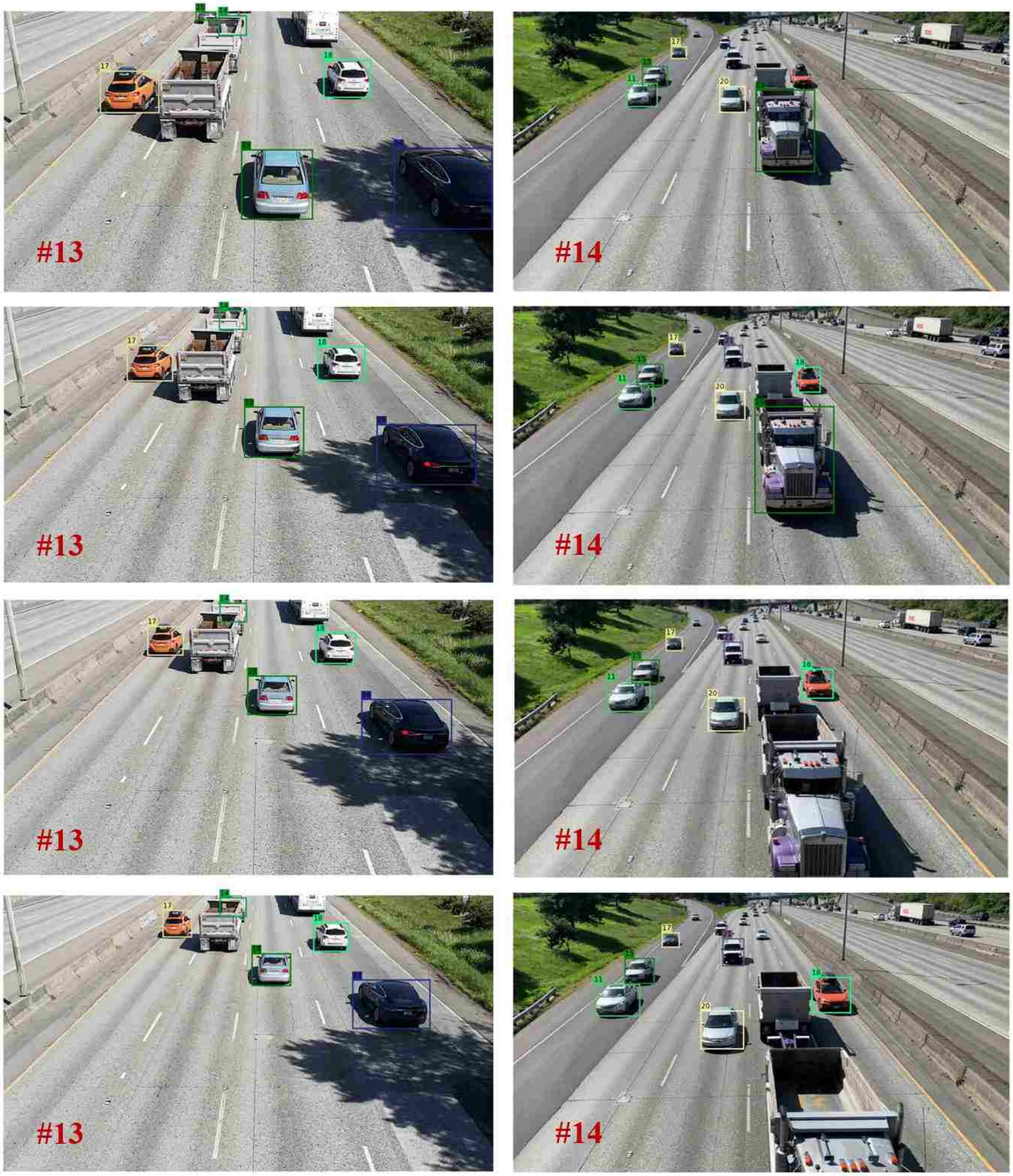


Figure 4-4 MOD result examples visualization (Camera #13, #14)

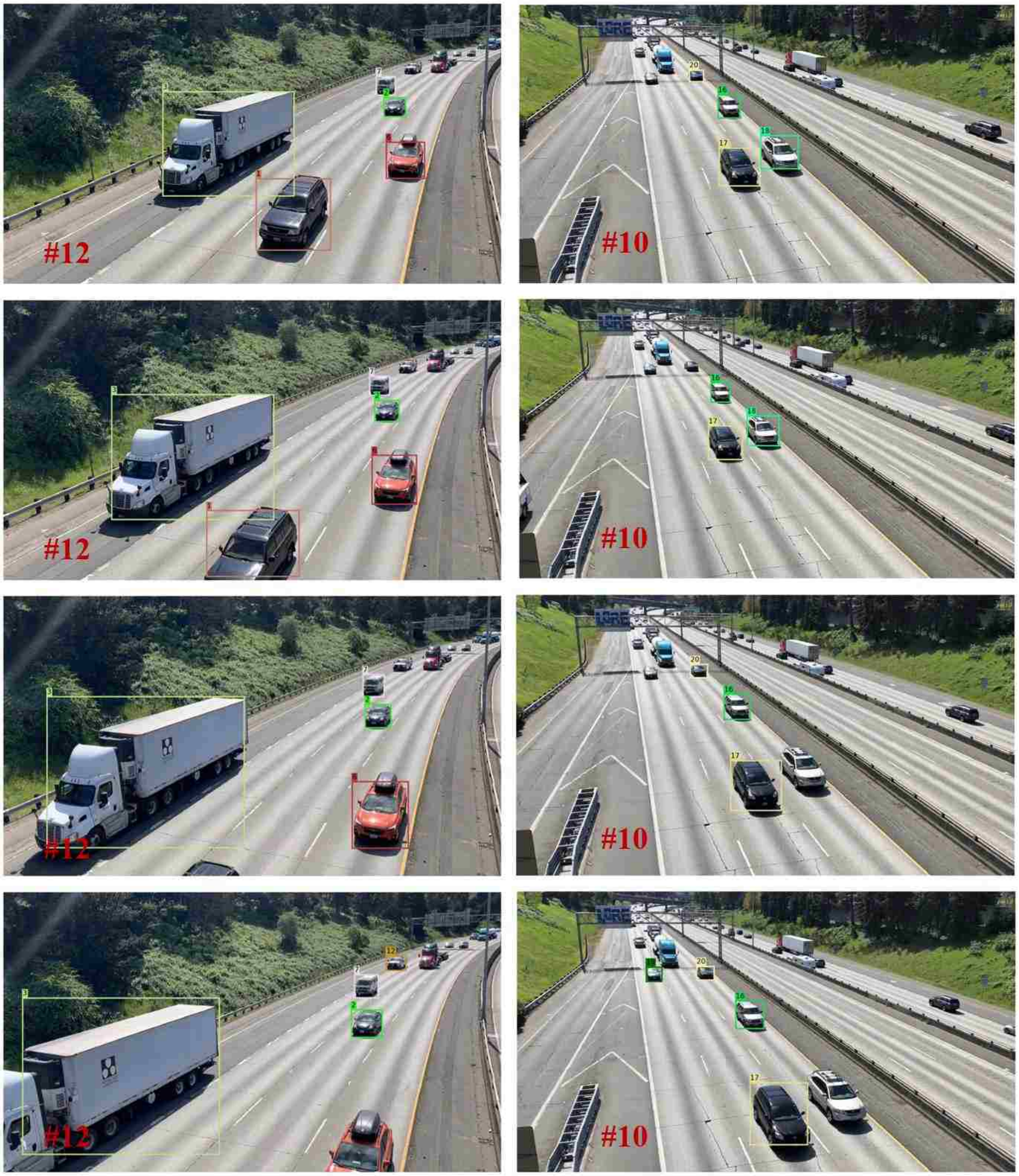


Figure 4-5 MOD result examples visualization (Camera #12, #10)

4.4 MCCTRI EXPERIMENT SUMMARY

4.4.1 *Camera Loop Determination*

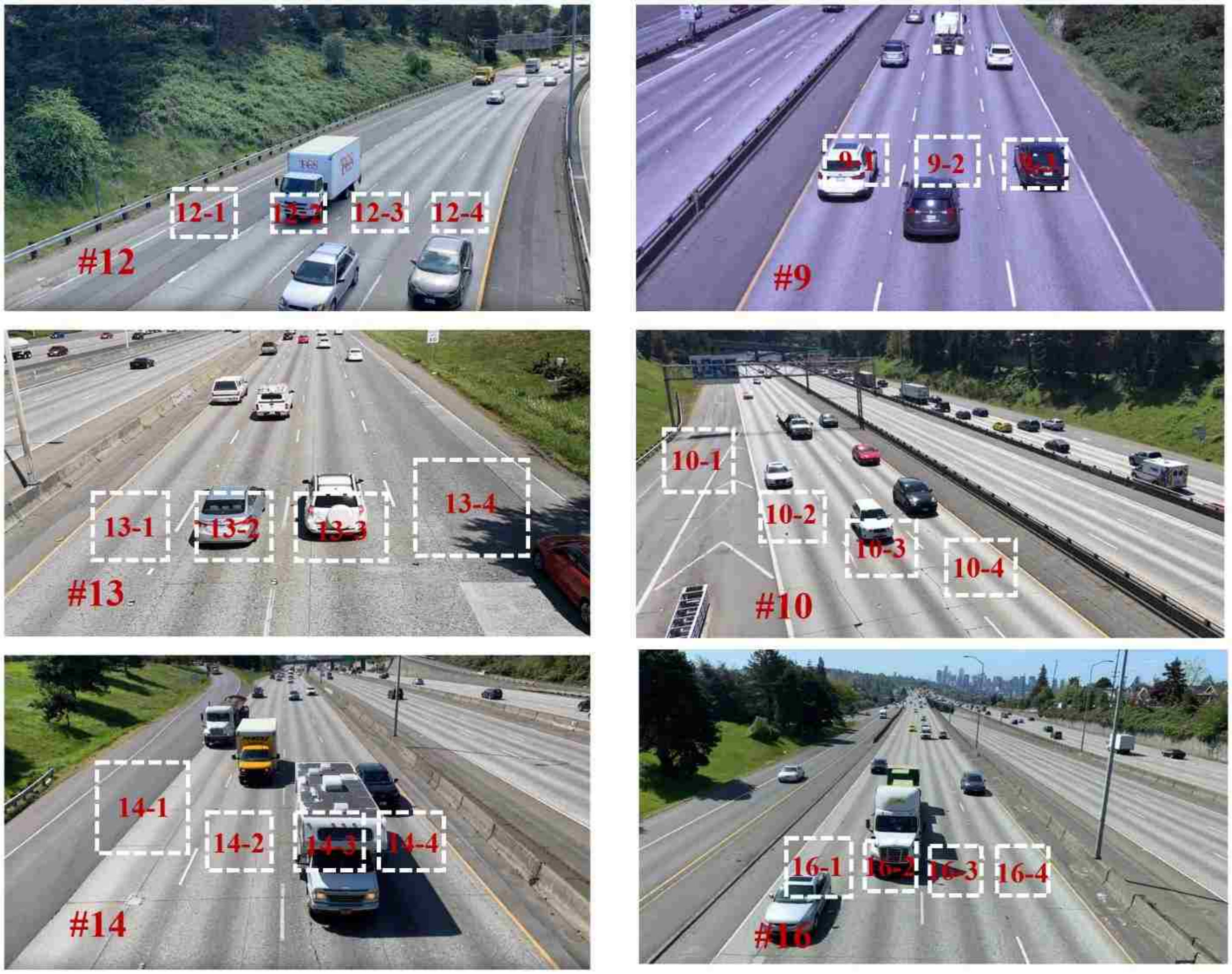


Figure 4-6 Camera loop location visualization

For each camera view, the author set the camera loop for each lane and visualize in the figure 4-6 camera loop location visualization. Based on the camera loop location and the camera graph, the whole inference graph can be built based on the connectivity relationship of each camera loop. The node is each camera loop and the edge are the loop relationship. The whole graph is used as the StCGIM link graph for re-ranking the candidates.

4.4.2 *Parameters Setting*

In the MCCTRI, the parameters setting is summarized in the following:

- For the clip-level of features extraction, the author sets T=5 (in the figure 3-11).
- For the Temporal-attention model, set the $\lambda = 0.5$ in overall loss fiction (in the equation (6)).
- For the searching time of T_{search}, set the $\mu = 0.7$ (in the equation (6)).
- For the candidates selection, the $\alpha = 1$ in the distance calculation equation (in the equation (7)).
- Based on the tuning results of the overall StCGIM target function, set the $w_{tte} = 0.5$ and $w_{tsc} = 0.5$.

4.4.3 *Result Summary and Comparison*

To further highlight the performance of MCCTRI, the author selected several current top-level methods to compare with MCCTRI. The evaluation results are summarized in table 4-3 MCCTRI multi-camera tracking result summary. The entire comparison method is divided into two categories, one is image-based vehicle Re-ID [12] [125], which mainly uses the characteristics of an image or multiple images to find and match with candidates, without the camera information. Another type is video-based vehicle Re-ID [13] [110] [115]. The video-based method generally incorporates more information, such as the spatial-temporal constraints, the camera information etc. These methods are:

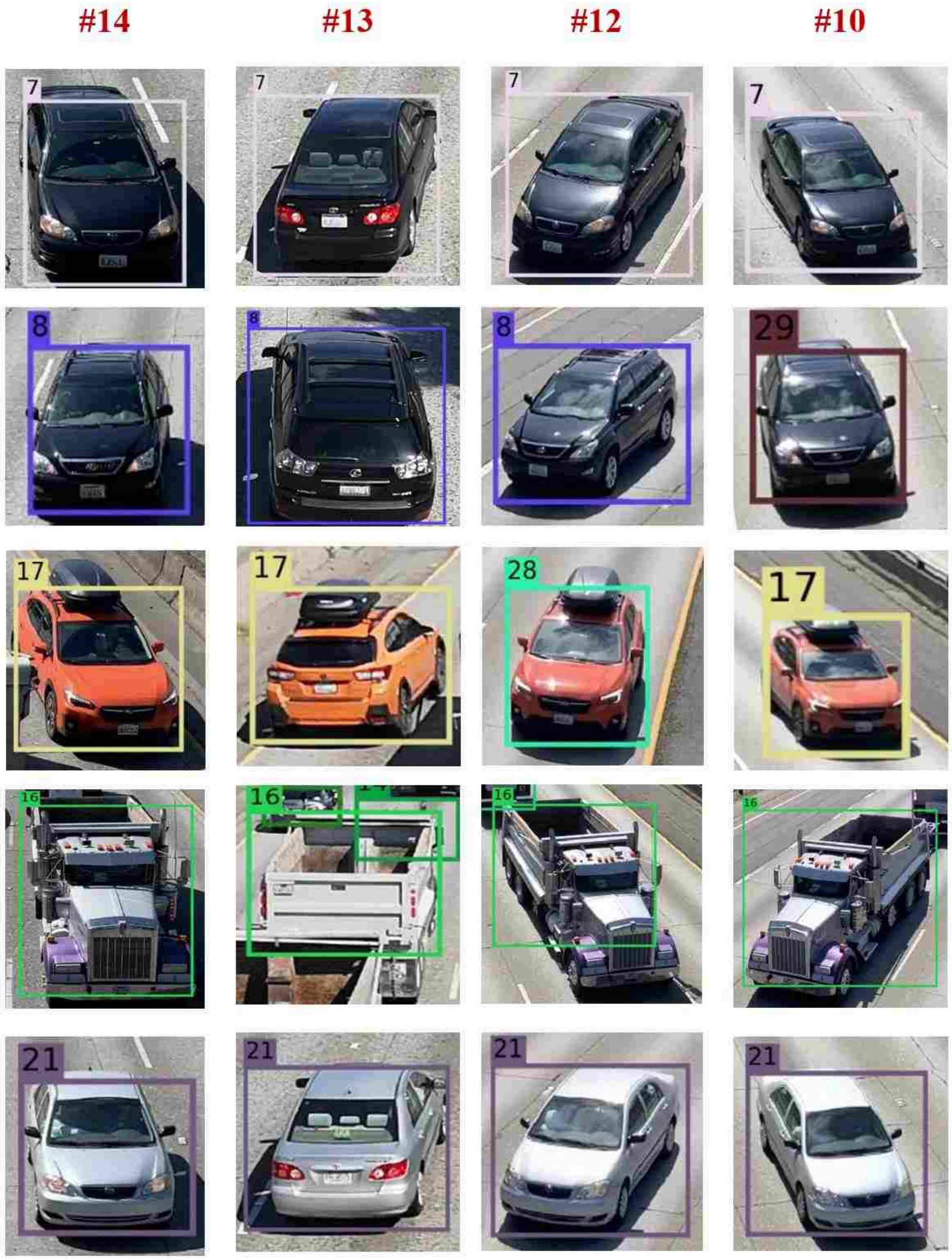


Figure 4-7 MCCTRI multi-camera tracking result example visualization

- The baseline method is proposed in the 2018 AI-city challenge by Zheng Tang etc. [125]; the author used the image-based Re-ID component in the method based on the inter-camera tracking on fusing visual and semantic features (FVS).

- AICUW_T2 [12], proposed by the UW image processing lab at 2019 CVPR AI city challenge track 2. Target at an image-based Re-ID solution, which ranked no.2 at the competition based on the temporal-attention model and metadata re-ranking.
- NCCU [115], proposed by the team at the University at Albany – SUNY at 2019. The authors proposed this method at the 2019 AI-city challenge at CVPR conference.
- PROVID [110], proposed by Xinchen Liu et al. at 2018. Since in the LHTV dataset, the plate number information is not allowed to use, so the author using the vehicle filtering by appearance part and the spatiotemporal relation model (STR) in the comparison.
- AICUW_T1 [13], proposed by the UW image processing lab at 2019 CVPR AI city challenge track 1. Target at a multi-camera tracking method, which ranked no.1 at the competition. This method integrated part of the AICUW_T2 [12] vehicle-based Re-ID method and add a spatial-temporal filter in the framework.

The overall comparison results are shown in table 4-3 MCCTRI multi-camera tracking result summary and the visualization example are in the figure 4-7 MCCTRI multi-camera tracking result example visualization. First of all, it can be seen that on the LHTV dataset, except for NCCU, video-based cross-camera vehicle Re-ID is always better than picture-based methods. Since the picture-based method does not include information such as the camera ID, the target matching procedure is much more complicated. Also, MCCTRI achieve a highest result for the evaluation of IDF_1 and IDR. The MCCTRI good performance are mainly due to 1) the four level of vehicle features integration, including frame-level, clip-level, identity-level and network-level of features; 2) the StCGIM method is useful for distinguishing and amplifying the better and worse candidates distance, which is helpful for the best candidate selection; 3) the camera loop graph narrow down the target searching range for the Re-ID for the cross camera targets.

Table 4-3 MCCTRI multi-camera tracking result summary

Method Types	**Methods**	**LHTV (6 cameras used)**				
		IDF1	**IDR**	**IDP**	**Max IDF1**	**Min IDF1**
Image-based Re-ID	Baseline (2017) [125]	0.4032	0.4105	0.4320	0.4279	0.4721
	AICUW_T2 (2019) [12]	0.5094	0.5211	0.5012	0.5421	0.5731
Video-based Re-ID	NCCU (2019) [115]	0.3403	0.3421	0.3031	0.3522	0.2998
	PROVID (2017) [110]	0.5312	0.5403	0.5377	0.5479	0.5235
	AICUW_T1(2019) [13]	0.7221	0.7272	**0.7289**	0.7492	0.6759
	MCCTRI	**0.7479**	**0.7395**	0.7254	**0.7543**	**0.6799**

For the adjacent link evaluation, the author summarized four adjacent link result in table 4-4 Adjacent link cross camera tracking result summary among the testing set. The distance is showing the adjacent camera location distance obtained based on the real route from google map. The orientation of the vehicle corresponding to the camera view, that is, which part of the vehicle captured by the camera. H represent the head, and T is the tail. Also, the ground truth means how many different IDs in each link video based on the real label. TP is the true positive value of IDs. FP is false positive value and FN is the false negative value. From the table, the results show that the orientation is an important factor to cross camera tracking result since the H2H accuracy (link 12-10) is the highest. Even the distance of two cameras is very close, the vehicle orientation still highly impacts the cross-camera Re-ID accuracy. Even the distance is only 8 meters, the camera link #14 - #13 is as good as the link #12 - #10 there are more false negative IDs. The same situation happened in camera #13 and camera #14. The best performance of adjacent link is the camera link #12 - #10, with the highest IDF$_1$ of 0.7543.

Table 4-4 Adjacent link cross camera tracking result summary

Link	**Distance**	LHTV (5 cameras used, adjacent link)					
		Orientation	Ground Truth	TP	FP	FN	IDF$_1$
14-13	8m	H2T	324	246	133	78	0.6998
13-12	2250m	T2H	342	252	135	90	0.6913
12-10	**356m**	**H2H**	**359**	**281**	**105**	**78**	**0.7543**
10-9	10m	H2T	329	258	125	71	0.7247

4.5 TRAFFIC INFORMATION ESTIMATION

4.5.1 *Evaluation Criteria*

For the traffic information estimation, there is a total of 20.03 min video from five different cameras used to evaluate the MCCTRI for traffic information estimation. For each evaluation, the five cameras video is split into 1 min video clip, each of them including 1800 frames, and then finish the multi-camera tracking task. Based on each clip tracking results and the parameters obtained from the evaluation process, the traffic information estimation is being evaluated by the following measurement:

For the traffic parameters value estimation, the error is using the difference of the value compared with the ground truth, then over the ground-truth value. The accuracy is one minus the error.

$$Err = \frac{X_{testing} - X_{truth}}{X_{truth}} \quad (26)$$

$$Acc = 1 - Err \quad (27)$$

Except the value estimation, the distribution accuracy estimation is using the KL distance of the distribution value, which measures of how the probability distribution differs from the testing and ground truth.

$$Err\_Dis = Dis_{KL}(P(X_{testing}) \parallel P(X_{truth})) \quad (28)$$

4.5.2 *Performance Summary*

4.5.2.1 Travel Time and Distribution & Speed and Distribution Estimation

Table 4-5 Link average travel time and speed value estimation result summary

Variable	**Type**	LHTV (5 cameras used, all 10 links)				
		Ave_Acc	Max_E	Min_E	Max_Elink	Min_Elink
Travel time	Average TT Value	94.89%	8.72%	1.21%	#14 - #9	#12 - #10
Speed	Average Speed Value	93.79%	10.98%	2.75%	#14 - #9	#12 - #10

The estimated results of average travel time and speed are summarized in table 4-5 link average travel time and speed value estimation result summary. Ave_Acc in the table means the average accuracy of the value. Max_E represents the maximum error of the camera link, and Min_E represents the minimum error. Max_Elink indicates which camera link has the largest error, which is helpful for researchers to analyze the specific cause of the error and possible improvements in the future. Min_Elink indicates the camera link where the smallest error is located. It can be seen that, regardless of the average travel time or speed, the average error is within 8%. MCCTRI can accurately extract point-to-point traffic value information. Among them, the most accurate of the traffic information estimation is camera link #12 - #10. The error of average travel time estimation error is only 1.21% and the speed error is only 2.75%. The link with the largest error is camera #14 - #9. The reasons for the larger error may be as follows: 1) The cross camera vehicle Re-ID results are not very accurate since the variety of the vehicle orientation and camera view. 2) The traffic flow is scattered, and the search window for travel time is hard to determined.

Table 4-6 Link average travel time and speed distribution estimation result summary

Variable	**Type**	LHTV (5 cameras used, all links)				
		Avg $\lvert Dis_{\mathrm{KL}} \rvert$	Max $\lvert Dis_{\mathrm{KL}} \rvert$	Min $\lvert Dis_{\mathrm{KL}} \rvert$	Max_Elink	Min_Elink
Travel time	TT Distribution	0.36	2.15	0.19	#14 - #9	#12 - #10
Speed	Speed Distribution	1.23	3.42	0.56	#14 - #9	#12 - #10

In this study, since there is an absolute error in vehicle multi-cameras tracking, it is not feasible to use only numerical results to estimate the distribution of traffic parameters. Therefore, the author estimated the distribution based on Gaussian distribution. In the parameters estimating, he has considered the different correlations between the accuracy of cross-camera tracking and the estimation of traffic parameters, and then estimated the traffic distribution of travel time and speed. As can be seen from table 4-6 link average travel time and speed distribution estimation result summary, the link with the largest travel time distribution KL distance error is from camera #14 - #9, and the smallest is camera #12 - #10. The average travel time distribution KL distance is 0.36. The speed error with the largest link is still from camera #14- #9, and the smallest is camera #12- #10.

The author also gives detailed statistics of the actual and estimated travel time and speed distribution of camera #12 - #10. The travel time information is shown in figure 4-8, and the speed information is shown in figure 4-9. Further analysis shows that the frequency distribution of the estimation is very close to the true distribution of travel time and speed distribution obtained by MCCTRI. The difference is that the distribution of real values is more gradual, not as concentrated as estimated. Since the StCGIM filters the extreme travel time to limit the number of candidates matching cross cameras, there is an error in the edge portion of the distribution. However, the true travel time distribution is very close to the estimated. The result shows that absed on MCCTRI, the dsitribution of traffic parameters can be estuimated in a high precision level.

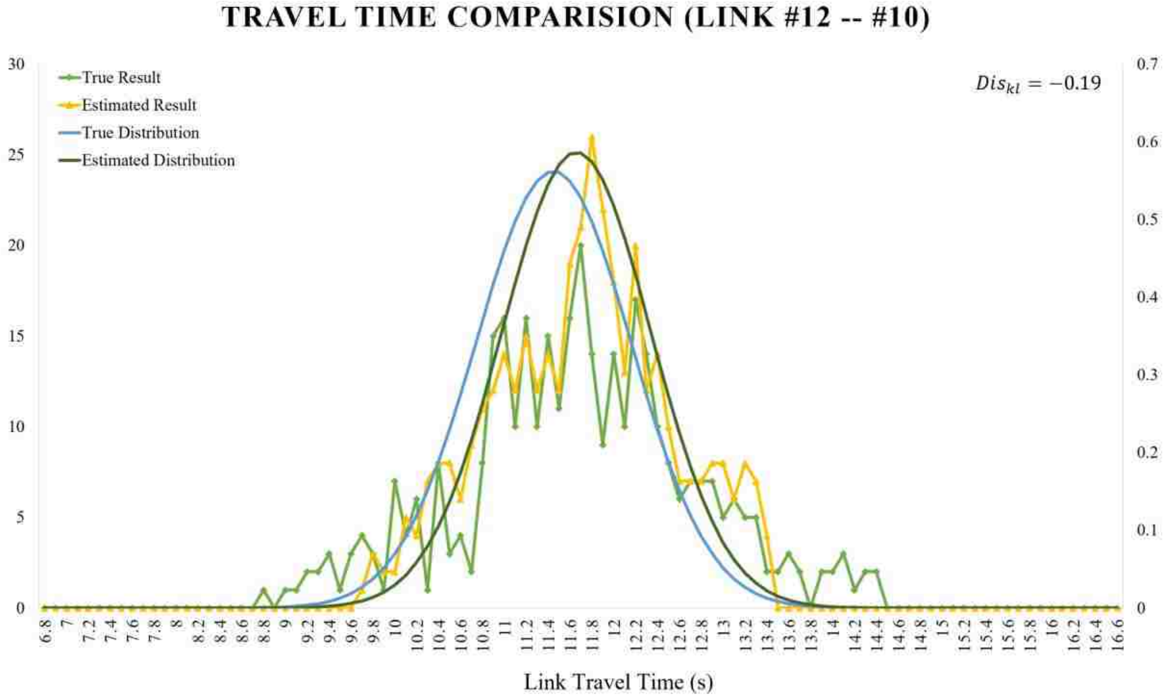


Figure 4-8 MCCTRI camera link (#12-#10) travel time distribution estimation compared with ground truth data visualization

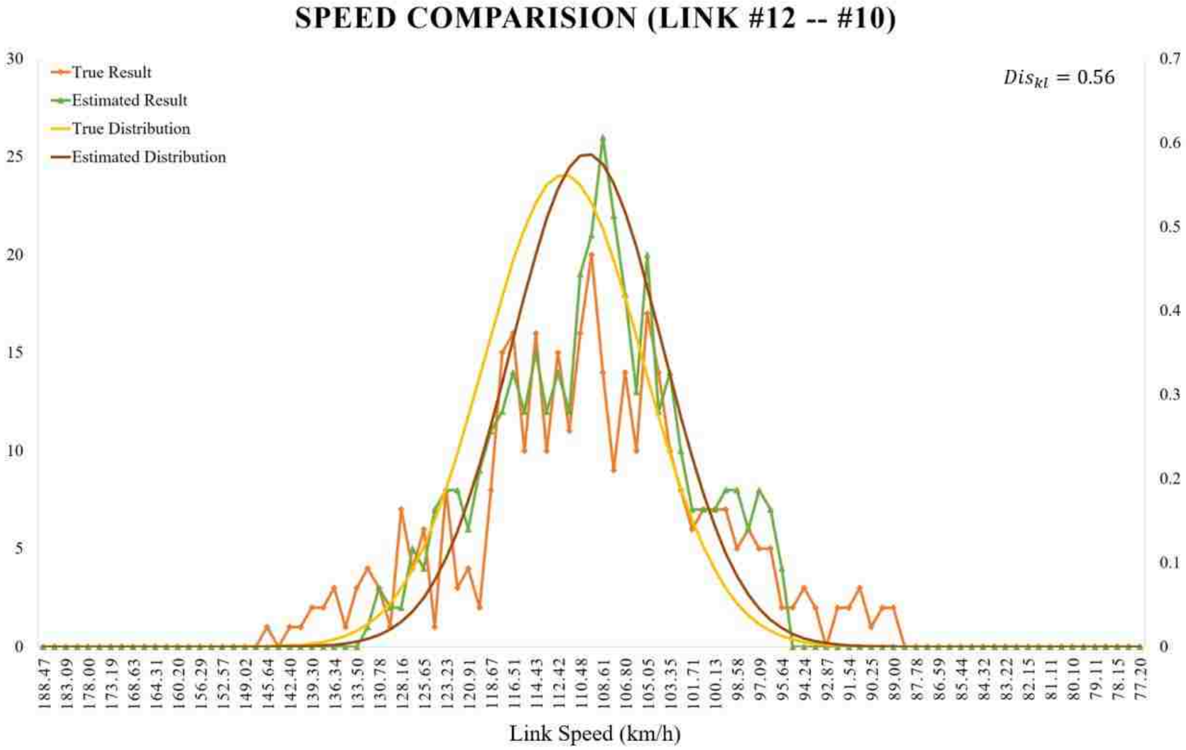


Figure 4-9 MCCTRI camera link (#12-#10) speed distribution estimation compared with ground truth data visualization

4.5.2.2 Volume and Distribution Estimation

Traffic volume estimation and the volume distribution estimation are unique advantages of the MCCTRI system. Through MCCTRI, not only can the user obtain a more accurate link volume, but also the corresponding multi-link volume distribution. This distribution information can greatly help planners and traffic operators to obtain the travel demand of different road sections. It can be seen from the table 4-7 link volume and distribution estimation result from the summary that in the volume and distribution estimation of all links, the average accuracy of the volume estimation reached 95.56%, and the overall accuracy of the traffic distribution reached 95.37%. Among them, the link with the largest error is camera #13 - #10. The difference between the actual value and the estimated value is 29 vehicles, and the maximum difference in flow distribution is 92.04%. The camera link with the smallest error is #10 - #9, the number is estimated to differ by only seven vehicles, and the traffic distribution is only 2.26% different.

Table 4-7 Link volume and distribution estimation result summary

Variable	LHTV (5 minutes, all links)				
	Ave_Acc	Max_Er	Min_Er	Max_Elink	Min_Elink
Link Volume	95.56%	29	7	#13 - #10	#10 - #9
Link Volume Distribution	95.37%	7.96%	2.26%	#13 - #10	#10 - #9

The author selected camera # 14 as an example to further analyze the performance estimates of MCCTRI's volume and volume distribution. As can be seen in figure Figure 4-10 link volume distribution estimation accuracy comparison (camera #14), the estimation result is accurate enough to obtain the cross camera volume distribution. The most significant error exists in camera #14 - #9, which is 5.54%. The estimated errors of the remaining camera links are less than 5%, and remain highly similar. Based on this experiment, it can be seen that the volume estimation based on MCCTRI is reliable and reasonable.

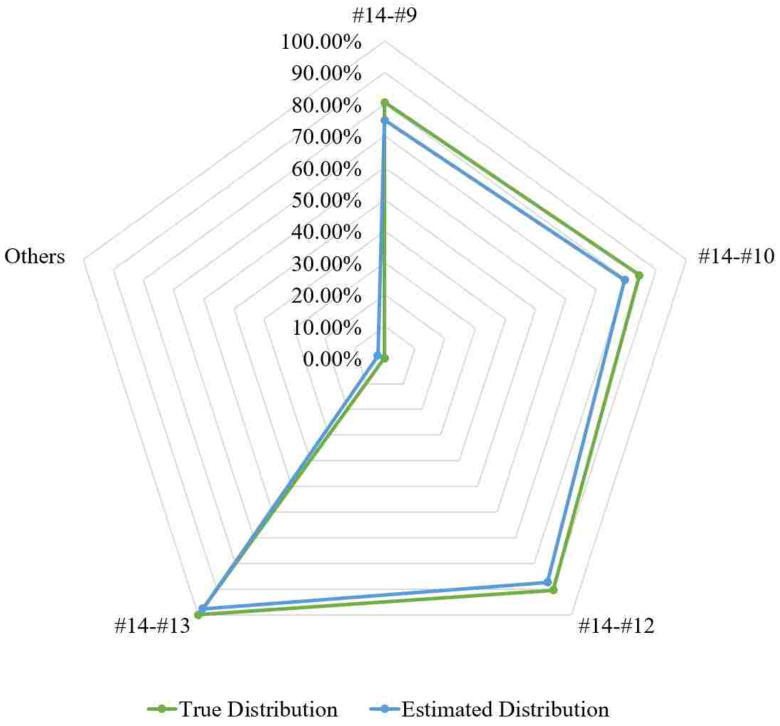


Figure 4-10 Link volume distribution estimation accuracy comparison (camera #14)

Similarly, the author uses camera #14 as an example to make a traffic distribution visualization combined with the spatial-temporal constraint based on I5 freeway in figure 4-11 link volume distribution visualization (camera #14). Of the vehicles passing camera #14, 97.65% passing camera #13, 87.37% passing camera #12, and 10.28% exit through the exits 169 and 170 between camera #13 - #12 to the local street. Of the remaining vehicles, 79.56% of the vehicles pass through camera #10, and 7.81% will leave from exit 171 at the camera #10. In the end, 75.02% of vehicles will pass through camera #9. From this traffic distribution map, MCCTRI can help researchers obtain accurate OD information and volume distribution across different camera regions.

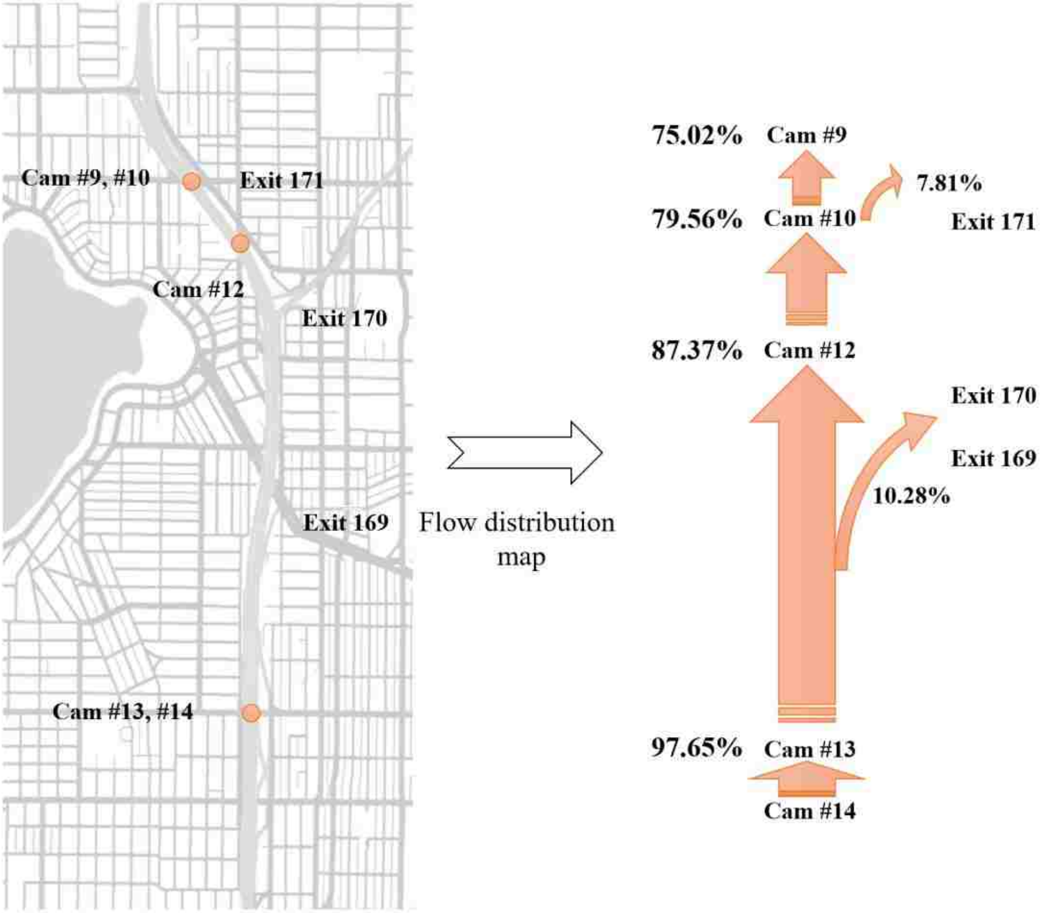


Figure 4-11 Link volume distribution visualization (camera #14)

Chapter 5. CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

In this research, the author proposed a novel framework for network-level traffic information estimation based on MCCTRI. The framework not only can enable each single camera to estimate traffic information precisely but also link each isolated camera into a graph and extract network level of traffic information based on MTMCT. Based on this framework, the author claims the following five main contributions:

1. A large-scale high-resolution (with 1080p quality) traffic video dataset is collected, which including 16 different cameras' videos. The total length of the videos is 1012.25 minutes. The dataset includes different road types (freeways and urban roads) and different traffic conditions (free flow, peak hour, congestion, and night traffic).
2. A cutting-edge image-based multi-camera tracking framework is improved and customized for the network-scale video-based traffic information extraction by integrated a new camera link graph model – Spatial-temporal Camera Graph Inference Model (StCGIM) based on the state-of-art MTMCT framework. Four levels of features, including frame-level, clip level, identity level, and network-level of features, are integrated into the MCCTRI. With efficient and effective information integration, the MCCTRI achieves 0.7479 of IDF1 and 0.7395 IDR on the LHTV dataset.
3. An Adaptative Accuracy Method (AAM) for traffic information estimation based on a different level of multi-camera tracking accuracy is proposed. Even the IDF_1 of the multi-camera tracking result is not precise enough to estimated directly, through the model, the traffic information can be estimated precisely.

4. Through the whole framework, not only including the traffic information value, such as link average speed, average travel time and volume, but also the distribution of each parameter can be estimated precisely. All the value information estimation error is less than 8% through the testing dataset, including five evaluation cameras. The KL distance of the estimated distribution and real distribution is less than 3.42.

The main originality and advantages of the framework are summarized in the following items. Firstly, it is the first method possible to obtain accurate traffic distribution information in a high penetration rate, which is very useful in traffic management and prediction. With the distribution information, the traffic signal timing can be optimized based on a different volume of directions. Also, the congestions can be predicted by merge the volume from different road sections. The volume distribution can also be used to estimate the network scale OD information. Secondly, in addition to obtaining necessary macroscopic information, it is also possible to obtain microscopic information based on different road service levels. For example, based on 95%, 70%, 50%, and other reliability levels of link travel time, link speed, and other information extraction. The information can better serve the roadway services nowadays. Thirdly, in addition to providing new information sources and new information scope, the method can also link the traffic information to each single target vehicle to obtain the identity-level of information of each tracking target. The information will be very helpful for the police officers to track an object and manage the roadway safety.

5.2 FUTURE WORK

The future works are towards two directions. For the computer vision oriented research, the author plans integrated more cameras, including different views, such as UAV camera and vehicle cameras, to challenge more complicated and useful Re-ID and tracking algorithms. Also, further explore the spatial-temporal information and find some way to draw the camera loop based on the algorithms autonomically is very necessary. Another direction is transportation-related research. Change the SCT module into an online tracking algorithm and boost the whole algorithm is a future research target. Also, it is a potential aim to use the valid traffic parameter distribution information and build new traffic forecasting model and congestion prediction model in the future.

BIBLIOGRAPHY

[1] Tang, J., Zou, Y., Ash, J., Zhang, S., Liu, F., & Wang, Y. (2016). Travel time estimation using freeway point detector data based on evolving fuzzy neural inference system. PloS one, 11(2), e0147263.

[2] Jenelius, E., & Koutsopoulos, H. N. (2013). Travel time estimation for urban road networks using low frequency probe vehicle data. Transportation Research Part B: Methodological, 53, 64-81.

[3] Papageorgiou, M. (1990). Dynamic modeling, assignment, and route guidance in traffic networks. Transportation Research Part B: Methodological, 24(6), 471-495.

[4] Angelelli, E., Arsik, I., Morandi, V., Savelsbergh, M., & Speranza, M. G. (2016). Proactive route guidance to avoid congestion. Transportation Research Part B: Methodological, 94, 1-21.

[5] Chowdhury, Mashrur, Amy Apon, and Kakan Dey, eds. Data analytics for intelligent transportation systems. Elsevier, 2017.

[6] Ke, R., Li, Z., Kim, S., Ash, J., Cui, Z., & Wang, Y. (2016). Real-time bidirectional traffic flow parameter estimation from aerial videos. IEEE Transactions on Intelligent Transportation Systems, 18(4), 890-901.

[7] G. Zhang, R. Avery, and Y. Wang, “Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras,” Trans. Res. Rec., J. Transp. Res. Board, vol. 1993, pp.138 -147, 2007.

[8] Malinovskiy, Y., Wu, Y. J., & Wang, Y. (2009). Video-based vehicle detection and tracking using spatiotemporal maps. Transportation research record, 2121(1), 81-89.

[9] González, Á., Garrido, M. Á., Llorca, D. F., Gavilán, M., Fernández, J. P., Alcantarilla, P. F., ... & De Toro, P. R. (2011). Automatic traffic signs and panels inspection system using computer vision. IEEE Transactions on intelligent transportation systems, 12(2), 485-499.

[10] Wan, Y., Huang, Y., & Buckles, B. (2014). Camera calibration and vehicle tracking: Highway traffic video analytics. Transportation Research Part C: Emerging Technologies, 44, 202-213.

[11] Babari, R., Hautière, N., Dumont, É., Paparoditis, N., & Misener, J. (2012). Visibility monitoring using conventional roadside cameras–Emerging applications. Transportation research part C: emerging technologies, 22, 17-28.

[12] Huang, T. W., Cai, J., Yang, H., Hsu, H. M., & Hwang, J. N. (2019, June). Multi-view vehicle re-identification using temporal attention model and metadata re-ranking. In Proc. CVPR Workshops (pp. 434-442).

[13] Hsu, H. M., Huang, T. W., Wang, G., Cai, J., Lei, Z., & Hwang, J. N. (2019, January). Multi-Camera Tracking of Vehicles based on Deep Features Re-ID and Trajectory-Based Camera Link Models. In AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California.

[14] Ke, R., Li, Z., Tang, J., Pan, Z., & Wang, Y. (2018). Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow. IEEE Transactions on Intelligent Transportation Systems, 20(1), 54-64.

[15] Ma, D., Luo, X., Jin, S., Guo, W., & Wang, D. (2018). Estimating maximum queue length for traffic lane groups using travel times from video-imaging data. IEEE Intelligent Transportation Systems Magazine, 10(3), 123-134.

[16] Ua-areemitr, E., Sumalee, A., & Lam, W. H. (2019). Low-Cost Road Traffic State Estimation System Using Time-Spatial Image Processing. IEEE Intelligent Transportation Systems Magazine, 11(3), 69-79.

[17] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X., & Kim, T. K. (2014). Multiple object tracking: A literature review. arXiv preprint arXiv:1409.7618.

[18] Barker, J. L. (1970). Radar, acoustic, and magnetic vehicle detectors. IEEE Transactions on Vehicular Technology, 19(1), 30-43.

[19] Anderson, R. L. (1970). Electromagnetic loop vehicle detectors. IEEE Transactions on Vehicular Technology, 19(1), 23-30.

[20] Le Pera, R., & Nenzi, R. (1973). Tana—An operating surveillance system for highway traffic control. Proceedings of the IEEE, 61(5), 542-556.

[21] Andersen, D. A., & McCasland, W. R. (1976). Alternate Designs for CCTV Traffic Surveillance Systems. Texas Transportation Institute Research Report, 173-1.

[22] Kennedy Jr, J. P. (1996). U.S. Patent No. 5,559,864. Washington, DC: U.S. Patent and Trademark Office.

[23] Uchiyama, T., Mohri, K., Itho, H., Nakashima, K., Ohuchi, J., & Sudo, Y. (2000). Car traffic monitoring system using MI sensor built-in disk set on the road. IEEE transactions on magnetics, 36(5), 3670-3672.

[24] Wang, Y., & Nihan, N. L. (2000). Freeway traffic speed estimation with single-loop outputs. Transportation Research Record, 1727(1), 120-126.

[25] Zhang, G., Wang, Y., Wei, H., & Chen, Y. (2007). Examining headway distribution models with urban freeway loop event data. Transportation Research Record, 1999(1), 141-149.

[26] Fu, X., Yang, H., Liu, C., Wang, J., & Wang, Y. (2019). A hybrid neural network for large-scale expressway network OD prediction based on toll data. PloS one, 14(5).

[27] Wang, D., Zhang, J., Cao, W., Li, J., & Zheng, Y. (2018, April). When will you arrive? estimating travel time based on deep neural networks. In Thirty-Second AAAI Conference on Artificial Intelligence.

[28] Sohn, K., & Hwang, K. (2008). Space-based passing time estimation on a freeway using cell phones as traffic probes. IEEE Transactions on Intelligent Transportation Systems, 9(3), 559-568.

[29] Horn, C., Klampfl, S., Cik, M., & Reiter, T. (2014). Detecting outliers in cell phone data: correcting trajectories to improve traffic modeling. Transportation research record, 2405(1), 49-56.

[30] Calabrese, F., Colonna, M., Lovisolo, P., Parata, D., & Ratti, C. (2010). Real-time urban monitoring using cell phones: A case study in Rome. IEEE Transactions on Intelligent Transportation Systems, 12(1), 141-151.

[31] Caceres, N., Romero, L. M., Benitez, F. G., & del Castillo, J. M. (2012). Traffic flow estimation models using cellular phone data. IEEE Transactions on Intelligent Transportation Systems, 13(3), 1430-1441.

[32] Calabrese, F., Di Lorenzo, G., Liu, L., & Ratti, C. (2011). Estimating origin-destination flows using mobile phone location data. IEEE Pervasive Computing, (4), 36-44.

[33] Marr D.Vision： A computational Investigation into the human representation and processing of visual information[M].[S.l.]：W H Freeman and Company，1982.

[34] Krizhevsky A，Sutskever I，Hinton G E.ImageNet classification with deep convolutional neural networks[J].Communications of the ACM，2012，60（2）.

[35] Girshick R，Donahue J，Darrell T，et al.Rich feature hierarchies for accurate object detection and semantic segmentation[C] Proceedings of IEEE Conference on Computer Vision and Pattern Recognition，2013：580-587.

[36] Lecun Y，Boser B，Denker J S，et al.Backpropagation applied to handwritten zip code recognition[J].Neural Computation，1989，1（4）：541-551.

[37] Freund Y，Schapire R E.Experiments with a new boosting algorithm[C] Proceedings of Thirteenth International Conference on International Conference on Machine Learning， 1996：148-156.

[38] Dalal N，Triggs B.Histograms of oriented gradients for human detection[C] Proceedings of IEEE Conference on Computer Vision and Pattern Recognition，2005：886-893.

[39] Vapnik V N.The nature of statistical learning theory[J]. Technometrics，1997，8（6）：1564.

[40] Lowe D G.Distinctive image features from scale-invariant keypoints[J].International Journal of Computer Vision， 2004，60（2）：91-110.

[41] [Viola P，Jones M. Rapid object detection using a boosted cascade of simple features[C] Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition，2003：511-518.

[42] Viola P，Jones M J.Robust real-time face detection[J]. International Journal of Computer Vision，2004，57（2）： 137-154.

[43] Lienhart R，Maydt J.An extended set of Haar-like features for rapid object detection[C] Proceedings of International Conference on Image Processing，2002：900-903.

[44] Sermanet P，Eigen D，Zhang X，et al.OverFeat：Integrated recognition，localization and detection using convolutional networks[J].Eprint Arxiv，2013.

[45] He K，Zhang X，Ren S，et al.Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence，2015，37（9）：1904-1916.

[46] Girshick R.Fast R-CNN[C] Proceedings of ICCV 2015， 2015.

[47] Ren S，He K，Girshick R，et al. Faster R-CNN：Towards real- time object detection with region proposal networks[C] Proceedings of International Conference on Neural Information Processing Systems，2015：91-99.

[48] Dai J，Li Y，He K，et al.R-FCN：Object detection via regionbased fully convolutional networks[C] Proceedings of NIPS 2016，2016.

[49] He K， Zhang X， Ren S， et al.Deep residual learning for image recognition[C] Proceedings of International Conference on Computer Vision and Pattern Recognition，

[50] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[51] Xie S，Girshick R，Dollar P，et al.Aggregated residual transformations for deep neural networks[C] Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition，2016：5987-5995.

[52] Redmon J，Divvala S，Girshick R，et al. You only look once：Unified, real- time object detectiono[C] Proceedings of CVPR 2015，2015：779-788.

[53] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[54] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[55] Liu W， Anguelov D， Erhan D， et al.SSD：Single shot multibox detector[C] Proceedings of European Conference on Computer Vision，2016：21-37.

[56] Jeong J，Park H，Kwak N.Enhancement of SSD by concatenating feature maps for object detection[C] Proceedings of CVPR 2017，2017.

[57] Ciaparrone, G., Sánchez, F. L., Tabik, S., Troiano, L., Tagliaferri, R., & Herrera, F. (2020). Deep learning in video multi-object tracking: A survey. Neurocomputing, 381, 61-88.

[58] Ciaparrone, G., Sánchez, F. L., Tabik, S., Troiano, L., Tagliaferri, R., & Herrera, F. (2020). Deep learning in video multi-object tracking: A survey. Neurocomputing, 381, 61-88.

[59] Weihao Gan, Shuo Wang, Xuejing Lei, Ming-Sui Lee, and C-C Jay Kuo. Online cnn-based multiple object tracking with enhanced model updates and identity association. Signal Processing: Image Communication, 66:95–102, 2018.

[60] Jun Xiang, Guoshuai Zhang, and Jianhua Hou. Online multi-object tracking based on feature representation and bayesian filtering within a deep learning architecture. IEEE Access, 2019.

[61] Ricardo Sanchez-Matilla, Fabio Poiesi, and Andrea Cavallaro. Online multi-target tracking with strong and weak detections. In European Conference on Computer Vision, pages 84–99. Springer, 2016.

[62] Kell, J.H., Fullerton, I.J., Mills, M.K., 1990. Traffic detector handbook. Technical Report.

[63] Yokota, T., Inoue, T., Nagai, T., Kobayahsi, Y., & Takagi, K. (1996). Travel time measuring system based on platoon matching: a field study. In Intelligent Transportation: Realizing the Benefits. Proceedings of the 1996 Annual Meeting of ITS America. ITS America.

[64] Kreeger, K. A., & McConnell, R. (1996). Structural range image target matching for automated link travel time computation. In Intelligent Transportation: Realizing the Future. Abstracts of the Third World Congress on Intelligent Transport SystemsITS America.

[65] Christiansen, I., & Hauer, L. E. (1996). Probing for travel time: Norway applies AVI and WIM technologies for section probe data. Traffic Technology International.

[66] Caruso, M. J., & Withanawasam, L. S. (1999, May). Vehicle detection and compass applications using AMR magnetic sensors. In Sensors Expo Proceedings (Vol. 477, p. 39).

[67] Kwong, K., Kavaler, R., Rajagopal, R., & Varaiya, P. (2009). Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors. Transportation Research Part C: Emerging Technologies, 17(6), 586-606.

[68] Charbonnier, S., Pitton, A. C., & Vassilev, A. (2012, May). Vehicle re-identification with a single magnetic sensor. In 2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings (pp. 380-385). IEEE.

[69] Gimeno, R. V. C., Celda, A. G., Pla-Castells, M., & Plumé, J. M. (2013, December). Improving similarity measures for re-identification of vehicles using AMR sensors. In 2013 9th International Conference on Information, Communications & Signal Processing (pp. 1-5). IEEE.

[70] Kuhne, R. D., & Immes, S. (1993). Freeway control systems for using section-related traffic variable detection. In Pacific Rim TransTech Conference (1993: Seattle, Wash.). Proceedings Pacific Rim TransTech Conference. Vol. 1.(1993)

[71] Sun, C., Ritchie, S. G., Tsai, K., & Jayakrishnan, R. (1999). Use of vehicle signature analysis and lexicographic optimization for vehicle reidentification on freeways. Transportation Research Part C: Emerging Technologies, 7(4), 167-185.

[72] Kwon, J., Coifman, B., & Bickel, P. (2000). Day-to-day travel-time trends and travel-time prediction from loop-detector data. Transportation Research Record, 1717(1), 120-129.

[73] Jeng, S. T. (2007). Real-time vehicle reidentification system for freeway performance measurements.

[74] JENG, S. T. C., & Chu, L. (2013). Vehicle reidentification with the inductive loop signature technology. Journal of the Eastern Asia Society for Transportation Studies, 10, 1896-1915.

[75] Guilbert, D., Le Bastard, C., Ieng, S. S., & Wang, Y. (2013, June). Re-identification by Inductive Loop Detector: Experimentation on target origin—Destination matrix. In 2013 IEEE Intelligent Vehicles Symposium (IV) (pp. 1421-1427). IEEE.

[76] Ali, S. S. M., George, B., & Vanajakshi, L. (2013, April). Multiple inductive loop detectors for intelligent transportation systems applications: Ramp metering, vehicle re-identification and lane change monitoring systems. In 2013 IEEE Symposium on Computers & Informatics (ISCI) (pp. 176-180). IEEE.

[77] Malinovskiy, Y., Wu, Y. J., Wang, Y., & Lee, U. K. (2010). Field experiments on bluetooth-based travel time data collection (No. 10-3134).

[78] Malinovskiy, Y., Lee, U. K., Wu, Y. J., & Wang, Y. (2011). Investigation of bluetooth-based travel time estimation error on a short corridor (No. 11-3056).

[79] Malinovskiy, Y., Saunier, N., & Wang, Y. (2012). Analysis of pedestrian travel with static bluetooth sensors. Transportation research record, 2299(1), 137-149.

[80] Abbott-Jard, M., Shah, H., & Bhaskar, A. (2013, October). Empirical evaluation of Bluetooth and Wifi scanning for road transport. In Australasian Transport Research Forum (ATRF), 36th (p. 14).

[81] Cheng, P., Qiu, Z., & Ran, B. (2006, September). Particle filter based traffic state estimation using cell phone network data. In 2006 IEEE Intelligent Transportation Systems Conference (pp. 1047-1052). IEEE.

[82] Yang, H., Liu, C., Gottsacker, C., Ban, X., Zhang, C., & Wang, Y. (2019). Cell-Speed Prediction Neural Network (CPNN): A Deep Learning Approach for Trip-Based Speed Prediction (No. 19-02492).

[83] Yoo, B. S., Kang, S. P., & Park, C. H. (2005). Travel time estimation using mobile data. In Proceedings of the Eastern Asia Society for transportation studies (Vol. 5, pp. 1533-1547).

[84] Cheng, P., Qiu, Z., & Ran, B. (2006, September). Particle filter based traffic state estimation using cell phone network data. In 2006 IEEE Intelligent Transportation Systems Conference (pp. 1047-1052). IEEE.

[85] Prinsloo, J., & Malekian, R. (2016). Accurate vehicle location system using RFID, an internet of things approach. Sensors, 16(6), 825.

[86] Cho, H., Seo, Y. W., Kumar, B. V., & Rajkumar, R. R. (2014, May). A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In 2014 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1836-1843). IEEE.

[87] Tian, Y., Dong, H. H., Jia, L. M., & Li, S. Y. (2014). A vehicle re-identification algorithm based on multi-sensor correlation. Journal of Zhejiang University SCIENCE C, 15(5), 372-382.

[88] Kerekes, R. A., Karnowski, T. P., Kuhn, M., Moore, M. R., Stinson, B., Tokola, R., ... & Vann, J. M. (2017, June). Vehicle classification and identification using multi-modal sensing and signal learning. In 2017 IEEE 85th Vehicular Technology Conference (VTC Spring) (pp. 1-5). IEEE.

[89] Kong, Q. J., Li, Z., Chen, Y., & Liu, Y. (2009). An approach to urban traffic state estimation by fusing multisource information. IEEE Transactions on Intelligent Transportation Systems, 10(3), 499-511.

[90] Ndoye, M., Totten, V. F., Krogmeier, J. V., & Bullock, D. M. (2010). Sensing and signal processing for vehicle reidentification and travel time estimation. IEEE Transactions on Intelligent Transportation Systems, 12(1), 119-131.

[91] Yifan Jiang, Hyunhak Shin, and Hanseok Ko. Precise regression for bounding box correction for improved tracking based on deep reinforcement learning. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1643–1647. IEEE, 2018.1

[92] Gaofeng, M. E. N. G., Pan, C., XIANG, S., & Wu, Y. (2018). Baselines Extraction from Curved Document Images via Slope Fields Recovery. IEEE transactions on pattern analysis and machine intelligence.

[93] Woesler, R. (2003, October). Fast extraction of traffic parameters and reidentification of vehicles from video data. In Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems (Vol. 1, pp. 774-778). IEEE.

[94] Shan, Y., Sawhney, H. S., & Kumar, R. (2005, October). Vehicle identification between non-overlapping cameras without direct feature matching. In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 (Vol. 1, pp. 378-385). IEEE.

[95] Guo, Y., Rao, C., Samarasekera, S., Kim, J., Kumar, R., & Sawhney, H. (2008, June). Matching vehicles under large pose transformations using approximate 3d models and piecewise mrf model. In 2008 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

[96] Hou, T., Wang, S., & Qin, H. (2009, June). Vehicle matching and recognition under large variations of pose and illumination. In 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (pp. 24-29). IEEE.

[97] Feris, R. S., Siddiquie, B., Petterson, J., Zhai, Y., Datta, A., Brown, L. M., & Pankanti, S. (2011). Large-scale vehicle detection, indexing, and search in urban surveillance videos. IEEE Transactions on Multimedia, 14(1), 28-42.

[98] Zheng, Q., Liang, C., Fang, W., Xiang, D., Zhao, X., Ren, C., & Chen, J. (2015, October). Car re-identification from large scale images using semantic attributes. In 2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP) (pp. 1-5). IEEE.

[99] Zapletal, D., & Herout, A. (2016). Vehicle re-identification for automatic video traffic surveillance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 25-31).

[100] Watcharapinchai, N., & Rujikietgumjorn, S. (2017, August). Approximate license plate string matching for vehicle re-identification. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). IEEE.

[101] Liu, X., Liu, W., Mei, T., & Ma, H. (2016, October). A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In European conference on computer vision (pp. 869-884). Springer, Cham.

[102] Liu, H., Tian, Y., Yang, Y., Pang, L., & Huang, T. (2016). Deep relative distance learning: Tell the difference between similar vehicles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2167-2175).

[103] Liu, X., Liu, W., Ma, H., & Fu, H. (2016, July). Large-scale vehicle re-identification in urban surveillance videos. In 2016 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). IEEE.

[104] Wang, Z., Tang, L., Liu, X., Yao, Z., Yi, S., Shao, J., ... & Wang, X. (2017). Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In Proceedings of the IEEE International Conference on Computer Vision (pp. 379-387).

[105] Shen, Y., Xiao, T., Li, H., Yi, S., & Wang, X. (2017). Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1900-1909).

[106] Kanacı, A., Zhu, X., & Gong, S. (2017). Vehicle reidentification by fine-grained cross-level deep learning. In BMVC AMMDS Workshop (Vol. 2, pp. 772-788).

[107] Zhang, Y., Liu, D., & Zha, Z. J. (2017, July). Improving triplet-wise training of convolutional neural network for vehicle re-identification. In 2017 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1386-1391). IEEE.

[108] Tang, Y., Wu, D., Jin, Z., Zou, W., & Li, X. (2017, September). Multi-modal metric learning for vehicle re-identification in traffic surveillance environment. In 2017 IEEE International Conference on Image Processing (ICIP) (pp. 2254-2258). IEEE.

[109] Li, Y., Li, Y., Yan, H., & Liu, J. (2017, September). Deep joint discriminative learning for vehicle re-identification and retrieval. In 2017 IEEE International Conference on Image Processing (ICIP) (pp. 395-399). IEEE.

[110] Liu, X., Liu, W., Mei, T., & Ma, H. (2017). Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. IEEE Transactions on Multimedia, 20(3), 645-658.

[111] Zhou, Y., Liu, L., & Shao, L. (2018). Vehicle re-identification by deep hidden multi-view inference. IEEE Transactions on Image Processing, 27(7), 3275-3287.

[112] Bai, Y., Lou, Y., Gao, F., Wang, S., Wu, Y., & Duan, L. Y. (2018). Group-sensitive triplet embedding for vehicle reidentification. IEEE Transactions on Multimedia, 20(9), 2385-2399.

[113] Lv, K., Deng, W., Hou, Y., Du, H., Sheng, H., Jiao, J., & Zheng, L. (2019). Vehicle reidentification with the location and time stamp. In Proc. CVPR Workshops.

[114] Chen, H., Lagadec, B., & Bremond, F. (2019, June). Partition and reunion: A two-branch neural network for vehicle re-identification. In Proc. CVPR Workshops (pp. 184-192).

[115] Chang, M. C., Wei, J., Zhu, Z. A., Chen, Y. M., Hu, C. S., Jiang, M. X., & Chiang, C. K. (2019, May). AI City Challenge 2019–City-scale video analytics for smart transportation. In Proc. CVPR Workshops (pp. 99-108).

[116] Tang, Z., Naphade, M., Birchfield, S., Tremblay, J., Hodge, W., Kumar, R., ... & Yang, X. (2019). Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In Proceedings of the IEEE International Conference on Computer Vision (pp. 211-220).

[117] Tan, X., Wang, Z., Jiang, M., Yang, X., Wang, J., Gao, Y., ... & Wen, S. (2019). Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 275-284).

[118] Wang, G., Wang, Y., Zhang, H., Gu, R., & Hwang, J. N. (2019, October). Exploit the connectivity: Multi-object tracking with trackletnet. In Proceedings of the 27th ACM International Conference on Multimedia (pp. 482-490).

[119] Z. Cui, K. Henrickson, R. Ke and Y. Wang, "Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting," in IEEE Transactions on Intelligent Transportation Systems.

[120] Newell, A., Yang, K., & Deng, J. (2016, October). Stacked hourglass networks for human pose estimation. In European conference on computer vision (pp. 483-499). Springer, Cham.

[121] Gao, J., & Nevatia, R. (2018). Revisiting temporal modeling for video-based person reid. arXiv preprint arXiv:1805.02104.

[122] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

[123] Wu, X., He, R., Sun, Z., & Tan, T. (2018). A light cnn for deep face representation with noisy labels. IEEE Transactions on Information Forensics and Security, 13(11), 2884-2896.

[124] Tang, Z., Naphade, M., Liu, M. Y., Yang, X., Birchfield, S., Wang, S., ... & Hwang, J. N. (2019). Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8797-8806).

[125] Tang, Z., Wang, G., Xiao, H., Zheng, A., & Hwang, J. N. (2018). Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 108-115).13