MULTI-TARGET MULTI-CAMERA TRACKING OPTIMIZATION USING PROBABILISTIC TARGET SEARCH

K. S. Wijayasekara

199492L

Degree of Master of Science

Department of Computational Mathematics

University of Moratuwa, Sri Lanka

March 2022

Declaration

I declare that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a Degree or a Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organizations.

Name of the Student

K. S. Wijayasekara

Signature of Student

Date: 16/07/2022

Supervised by

Dr. Subha Fernando

Signature of the Supervisor

Date:

Prof. Chinthaka Premachandra

Signature of the Supervisor

Date:

Acknowledgment

Throughout the development of this dissertation, I received a great lot of assistance from a variety of people. Firstly, I would like to express my gratitude to my supervisors, Dr. Subha Fernando and Professor Chinthaka Premachandra, for their dedication, patience, and direction in ensuring the success of this study. Her input was key to developing the research topic and technique. Your unwavering support and insightful comments have always aided me in raising the bar on my work.

I'd also like to express my gratitude to Prof. Asoka Karunananda for his assistance in preparing the thesis materials and demonstrating proper research methodology. In addition, I want to express my gratitude to Dr.Sagara Sumathipala for his invaluable assistance in completing the project on time.

My heartfelt thanks to all of the other lecturers and non-academic staff members who assisted me in making this project a success. I also want to express my deepest gratitude to my coworkers for their assistance in completing this project.

Finally, I thank my beloved parents, wife, and other family members for their continuous support and encouragement. If not for you, this project would not have been as successful as it is.

Abstract

Smart surveillance in smart cities has become an important feature to be used in resource utilization and city-wide security areas. Multi-target multi-camera tracking has been one of the core areas in smart surveillance since the overlapped field of views within cameras cannot be expected in the real world scenarios. The inefficiency in MTMCT has caused this feature to not be used in real time applications. Hence how to make vehicle re-identification feature signature matching efficient in multi target multi camera tracking has become a research problem.

This research introduces a trajectory based probabilistic search algorithm to reduce target search space and increase the efficiency of the MTMCT. The solution consists of a YOLO v4 based object detection module, IOU based single camera tracking module, OSNet based feature extraction module and a cross camera identification module using probabilistic target search algorithm. The system takes video streams in and outputs the global trajectory of each target target. The evaluation is done using identification F1 score and the efficiency was measured using the number of frames processed in a second.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
MTMCT	Multi Target Multi Camera Tracking

CHAPTER 1

INTRODUCTION

1.1 Prolegomena

Nowadays smart cities have become the talk of the town and it is no longer limited to science fiction movies. Countries like Singapore have already implemented most of the smart features in their cities. Smartness is usually measured under the topics of Mobility, Healthcare, Security, Water, Energy, Engagement and Community, Economic Development and Housing and Waste.

The concept of smart vision helps to improve the smartness of mobility and security in a city. In the work of Montemayor [1], they have explained the useability of computer vision in logistic management of a city. On the other hand, cognitive vision can be used to detect crimes and improve the security of the city.

One of the frequently faced challenges in this research area is inefficiency of vehicle re-identification feature signature matching in multi target multi camera tracking (MTMCT) in real time applications. Most of the current approaches, stream videos to a central server and then process. However, this causes many hindrances due to bandwidth limitations, need of high performance cloud servers, single point failure and data privacy issues.

This chapter gives an overall idea about the research we have done in the subject area of multi-target multi-camera tracking using hierarchical probabilistic target search. In doing so, we have structured the introduction chapter under several subheadings, aims and objectives, background and motivation, problem definition and our solution.

1.2 Aim and Objectives

The aim of this project is to develop a real-time multi target multi camera tracking system using Probabilistic Target Search.

1. Critically reviewing existing approaches for multi target multi camera tracking
2. In-depth study of relevant technology and tool frameworks

3. Develop a method to predict object probability in camera network and feature matching algorithm using a trajectory based probability hierarchy
4. Evaluate the developed approach using AI City Challenge 2021 dataset

1.3 Background and Motivation

With the advancement of smart surveillance concepts, research on tracking particular objects and people became significant. Different research parties carried out advanced research on areas such as object tracking, people/vehicle demographics, traffic monitoring and burglary detection. However while seeking solutions for above mentioned problems, researchers faced a major challenge that needed an accurate and efficient solution to continue advancements in surveillance based research. This problem was none other than tracking objects though non-overlapped multiple cameras.

1.4 Problem Definition

Multi target multi camera tracking is an important problem and many researchers have conducted research to improve the accuracy and performance of this problem. However after studying and reviewing previous work done, still there is room left for research in this area. Addressing how to make vehicle re-identification feature signature matching efficient in multi target multi camera tracking has become a research problem.

1.5 Novel Approach to MTMCT

This research focuses on a novel approach for improving multi target multi camera tracking efficiency using hierarchical probabilistic search. The solution consists of a three step process. First single camera tracking must be done and re-identification features must be extracted from each tracked object. The second step is to extract traffic condition information from each camera. The last step would be to search targets using a hierarchy constructed using the predicted trajectory of each target using extracted traffic condition information.

1.6 Resource Requirements

This research requires multiple camera streams without non-overlapping field of views. Ideally access to a real time surveillance camera system in a city would serve the purpose. However in this project the AI City Challenge 2021 dataset has been used to implement the solution. The feature extraction model training and real time re-identification using feature matching requires a computer with a powerful GPU.

1.7 Structure of the Thesis

Chapter 1 of this thesis provides an overview of the research. This covers a brief introduction to the background of the problem domain, identified problem, solution and resource requirements of the project. Chapter 2 consists of a critical review on previous related work and challenges identified in their solutions. Chapter 3 consists of deep analysis on related technologies and past work done related to the technology.

Chapter 4 explains the approach taken in intended research and inputs, outputs, process and features of the outcomes. Chapter 5 and chapter 6 includes design and implementation details of the project. Finally chapter 7 and chapter 8 discuss the evaluation of the proposed system and conclusion of the outcomes.

1.8 Summary

In this chapter we have discussed the roots behind selecting MTMCT domain, the identified problems in the area and our target milestones in brief. In addition to that this chapter provides the overall structure of this project.

CHAPTER 2

DEVELOPMENTS AND ISSUES IN MTMCT

2.1 Introduction

In Chapter 1, we provided an overall picture of the research presented in this thesis. This chapter gives a critical review of literature in Multi-Target Multi-Camera Tracking. In doing so, we have structured the literature review under several subheadings, namely, early developments, latest development, and challenges in MTMCT. This chapter also summaries challenges in MTMCT and defines the Research Problem.

2.2 Early Developments in MTMCT

Intelligent video surveillance is a multi-disciplinary subject area related to computer vision, algorithms, signal processing, pattern recognition, communication, embedded computing and cameras. Areas like multi-camera tracking, object re-identification, multi-camera activity analysis and intelligent video surveillance are usually discussed under this field.

In early developments tracking objects in a field of view and a similarity evaluation algorithm [2] has been used to match objects from different fields of views to re-identify the same object across multiple cameras. In addition to that, methods like using a top-down view occupancy map for simultaneously tracking the bounding boxes and the global trajectory on the top-down map. Across most of the early works in MTMCT, intra-camera tracking frequently loses tracking and causes fragmented tracklets due to complication like occlusion in crowded environments. Due to this limitation most of the subsequent work has focused on solving fragmented tracklets issue in the camera view using mathematical approaches.

However, introducing advanced mathematical approaches has made these methods more complex and may need multiple passes to process the data. Also it has still been hard to achieve real-time object tracking using these methods. This was mainly due to use of exhaustive algorithms to extract features for identifying objects and feature extraction.

With the evolution of deep learning, it has been discovered the hindrances caused by above mentioned mathematical algorithms can be overcomed. Deep learning based method [7] was proposed for people detection to overcome above mentioned issues in using complex mathematical approaches. With the observation of benefits introduced by moving to deep learning, solutions with more advanced features like people counting [8] have also been developed on people tracking.

2.3 Breakthrough in MTMCT

As discussed in the previous topic, the history of MTMCT has evolved from object detection, object tracking and object re-identification. In recent researches MTMCT has been addressed as a single problem and researchers have provided complete solutions to MTMCT. However the significant contribution of such research can be categorized into 3 subtopics: Re-Identification Feature Extraction, Single Camera Tracking and Inter Camera Tracking. Following subsections discuss the recent significant contributions to each area mentioned above.

2.3.1 Re-Identification feature extraction

With the advancement of convolutional neural networks most of the recent developments [9, 19, 21, 24] have used different variations of CNNs to extract object features in a vector format. Modifications to CNNs used for feature extraction such as adding an unsupervised excitation layer to enhance representation learning. However some of the above work has blended some other technologies in extracting more comprehensive features. Vehicle metadata such as vehicle type, brand and color can be embedded to vehicle feature set [9] as an auxiliary features. MTMCT solutions for human tracking can be improved by using human pose information [28] and pose information can be further used to identify the occlusion level [26] to extract the confidence of features extracted. In single camera tracking the local neighborhood, the consecutive frames in the temporal domain are used. However in multi camera tracking this property is not considered and the matching solely depends on the visual features extracted. Including locality aware info to the feature vector [29] could solve this issue. While extracting visual features using the object tracked in single camera tracking, which instance of the track to be used for feature extracting could be a critical decision to make. Extracting the most discriminant feature in the track using temporal attention [31] could help to retrieve better feature representation. Using just the visual feature embedding of the whole object could be less accurate due to objects with similar appearance may occur in the system. i.e: vehicles with the same brand, model and color, people wearing similar clothes. More specific but commonly occurring in all objects areas such as face [34] in human MTMCT or number plates [55] in vehicle MTMCT could be used if high resolution camera streams are available.

2.3.2 Single camera tracking

Most of the MTMCT work has used single camera tracking as their base of tracking. Some of those work have contributed with novel algorithms for tracking while others have brought in improvements to the existing algorithms. Tracking by detection algorithms have been widely used [19] in single camera tracking in MTMCT solutions. Tracklet Plane Matching (TPM) [38] and DeepSort [39] like algorithms are highly utilized in many cases. However having 2 separate modules for detection and appearance embedding modules for association could lead to inefficiencies in tracking by detection. Using modified Joint Detection Embedding (JDE) Tracker [40] instead of state of art tracking by detection algorithms have shown significant improvements in some work [24].

Many MTMCT solutions [32, 41, 42, 43] have used graph models to solve the single camera tracking problem as a cost minimization problem. Graphs are usually built using detected objects as vertices[32, 41, 42]. However, assuming conditional independence of detections and high-dimensional affinity matrices can be considered as major limitations in detection-based graph models. This is due to missing conditional dependence due to ignoring temporal information introduced frame to frame transitions and inability to find a global minimum in a high-dimensional affinity matrix. On the other hand some research work [43] has focused on using tracklets as the vertices of the graph. Specifically Tracklet-Net-Tracker (TNT) [44] based single camera tracking has shown significant accuracy improvements [31] by utilizing short trajectory as vertices and associating them to generate accurate complete trajectories.

Tracklet splitting or fragmentation is one of the biggest problems in single camera tracking. Object detection issues, occlusion and inadequate frame rates are the common reasons for these issues. Pose detection can be used to identify occlusion [22, 26] in human tracking and incorporate such information in tracking algorithms. In addition to that some techniques have been introduced to refine tracklets with predefined zones and GPS information [20] to reconnect split tracklets caused by occlusion and eliminate invalid tracklets. While attempting to regroup related tracklets together, there could be noise tracklets which must be filtered. Single camera tracking filters have been introduced [23] to eliminate unreliable tracklets.

Traffic awareness [9, 27] is a novel concept introduced in MTMCT single camera tracking. utilizing the prior knowledge of usual traffic patterns, moveable directions, time taken to move and movement allowed areas are usually defined as traffic awareness in MTMCT. Using this information could contribute a lot in solving the tracklet splitting problem and filtering noise from trajectories. However this extra information comes with considerably labor intensive manual annotation.

2.3.3 Inter camera tracking

Inter camera tracking is the key component of the MTMCT problem. While feature extraction and single camera tracking comes with their own state of art solutions, multi camera tracking or inter camera tracking is unique to MTMCT. The significant difference between re-identification and MTMCT is the need to generate a global trajectory of the object in MTMCT where re-identification requires only association of similar objects. In other words MTMCT can be considered as an extension to re-identification where continuous identification and tracking is required.

Most common approach to multi camera tracking is utilizing appearance feature vectors for identifying the same object across multiple cameras. Appearance feature distance based tracklet association [19] is the most common method used in inter camera tracking. Multiple distance matrices such as Euclidean distance, Cosine distance and Mahalawanabis distance have been used to calculate the feature distance. Combinatorial optimization algorithms such as Kuhn-Munkres algorithm (Hungarian Algorithm) have been used [26] to solve the assignment problem.

Feature grouping [25] and hierarchical clustering have also been used [27, 30] for inter camera tracklet association. One of the disadvantages of using hierarchical clustering is the need to wait till features of all detections are retrieved. However the actual object transition happens between adjacent cameras. This property can be utilized to improve the clustering process by introducing sub-clustering for adjacent cameras [24].

In contrast to the common method of matching tracklets by matching targets, Tracklet-to-target matching [10, 32] has been tested by several works. When complete single camera trajectories are compiled using local tracklets and then form global trajectories using those complete single camera trajectories, there could be performance hindrances due to multiple matching requirements. In tracklet-to-target assignment, unique targets are identified and local tracklets are assigned to relevant targets. Global trajectories can be generated straight away by using this method. Restricted Non-negative Matrix Factorization (RNMF) [10] and TRACklet-to-Target Assignment (TRACTA) algorithm [32] have been used to solve this assignment problem.

Apart from efforts to identify objects across multiple cameras using state-of-are methods, There have been several efforts to constrain the search space to increase performance. Quality of extracted features can play a major role in accuracy of matching outcomes. Occlusion can directly affect the quality of features extracted since the parts of the intended object to track may be covered by other objects. If the occlusion can be identified, the obstacle occluded features can be discarded [22] to improve accuracy.

MTMCT is usually carried out for a camera network without overlapping field of views and distributed in a limited geographical space. In such a network, object transition between any camera pair in the network is not possible due to camera adjacency. In expected operation object transitions happen between neighboring cameras. This property can be used to improve inter camera tracking accuracy and performance by applying constraints on possible matching pairs. The concept of Trajectory based camera link model is introduced in some MTMCT solutions [9, 27, 31] to incorporate this advantage. In addition to that, applying constraints on association using spatio-temporal information such as traveling time, road structures, and traffic rules [19, 22, 35, 36] has also been done in multiple works.

While the majority of the work has been focusing on associating already seen tracklets, multi-camera trajectory forecasting (MCTF) [22] has been introduced to predict in which camera a particular object will appear next. Even though this method has not been incorporated to the MTMCT or re-identification in the above work, this information carries a massive potential to improve MTMCT.

2.4 Probability Based Target Search

One of the primary applications of MTMCT is real time object detection. When re-identifying objects across multiple cameras the search space becomes enormously large. This hinders the performance of MTMCT drastically. Ha and Cho [14] have suggested a hierarchical probabilistic target search(HPTS) algorithm to decrease search time in target search of drone systems. In this work they have used multi layered drones in different altitudes. higher level drones give an idea about the probability of finding the target in each lower level drone. Search is performed on the descending order of probabilities.

2.5 Challenges in MTMCT

Above section has given an understanding about the recent research developments in the MTMCT domain. However in some cases there are some common limitations identified in this domain. Most of those issues and limitations are inherited by the challenges in video surveillance. Wang [15] has noted several challenges faced in multi-camera video surveillance. One common issue in multi camera tracking is different cameras having different lighting conditions and also they might come from different brands which will output video streams with different qualities. This can put up a huge impact on the re-identification feature vector. Another common challenge is occlusion of objects. Target objects being covered by another object for some time is a frequent incident in surveillance. This becomes a challenge in even manual surveillance using CCTV footage.

Apart from the common vision based challenges, MTMCT has a limitation with performance. When the target count increases, the search space grows exponentially and becomes resource expensive in feature rematching for re-identification.

2.6 Problem definition

Multi target multi camera tracking is an important problem and many researchers have conducted research to improve the accuracy and performance of this problem. However after studying and reviewing previous work done, still there is room left for research in this area.

The primary problem of multi target multi camera tracking is the enormous amount of feature matching to re-identify an object across cameras. Though existing research has focused on trajectory based camera link models to minimize this problem, majority of it is focussed only on adjacent camera tracking. However in practical scenarios it is not acceptable to assume there will be only one adjacent entry camera for a particular camera exit. In addition to that it should be viable to handle missing the tracking from the adjacent cameras but re capturing in subsequent cameras.

The secondary problem of MTMCT is generating a unique feature signature for re-identification. As mentioned above many of the previous work have struggled with this specially when it comes to vehicle re-identification. One of the fundamental issues with CCTV surveillance footage is low resolution. Due to this and occlusion it is nearly impossible to detect vehicle license plates. Therefore the most common way is to extract visual features of the entire vehicle. However the solutions can not rely on such features since there could be many vehicles from the same model and same color. Nevertheless, lighting conditions can put a huge burden on this. Especially at night most of the surveillance cameras work in infrared mode and provide only a grayscale image.

In addition to that in practical scenarios streaming videos to servers can cause high bandwidth demands and also can cause serious privacy issues due to data transmission and storing in cloud servers. Therefore addressing how to make vehicle re-identification feature signature matching efficient in multi target multi camera tracking has become a research problem.

2.7 Summary

This chapter presented a critical review of literature in MTMCT and identified achievements in the field and research challenges. In the next chapter, we give a detailed discussion on the technology adopted for implementing MTMCT.

CHAPTER 3

TECHNOLOGIES USED FOR MTMCT

3.1 Introduction

In chapter 2 we have discussed and critically reviewed the literature of multi-target multi-camera tracking and limitations and challenges of them. We have defined our research problem as how to make vehicle re-identification feature signature matching efficient in multi target multi camera tracking. In order to solve this problem, we have identified time taken for feature matching over multi camera streams must be more efficient. In this chapter we will discuss the technologies used to extract re-identification features and theory behind hierarchical probabilistic search.

3.2 Technologies Used In Object Detection

A vast amount of research has been done to identify objects in an image frame. The primary goal in this task is to identify the availability of the target object in the frame and if it does the location of it. In early stages this was done using mathematical algorithms.

HOG [16] is one of the popular ways of deriving a descriptor for a bounding box of an object. HOG is based on identifying intensity gradients of object shape within an image. To calculate the HOG descriptors of an image, the image is first divided into small regions and for each region histogram of gradient directions or edge orientations will be computed. Combination of all these histograms forms the descriptor at last. However HOG feature is not sensitive to variance of light, direction and size whereas it will not work well in scale invariant rotation invariant scenarios.

Haar [17] is another feature extraction method. Unlike HOG features, Haar features are independent of illumination variation. HOG features are much heavier for computation compared to Haar features. Three categories of Haar features are edge features, linear features and center features. Haar features can be computed fast because integral images are being used. Haar features are computed similar to the coefficients of Haar wavelet transform. A template connecting black and white rectangles, and the irrelative coordinates to the origin of the search window is what form Haar like features. For The Detection of horizontal, vertical, and symmetric structures, Haar-like features are found to be well suitable.

However with the advancement of deep learning, convolutional neural network based object detection has out-performed most of the mathematical algorithms used in object detection. Jiao Et. el [18] has carried out a deep review on deep learning based object detection. During the early days of CNN based object detection, two consequent stages were involved in detecting objects in an image frame. R-CNN (Region based CNN detector) [45], Fast R-CNN (faster version of R-CNN) [46], Faster R-CNN (Uses novel RPN) [47] and Mask R-CNN (Extension of Faster R-CNN) [48] are classic examples of two stage detectors. Two stage detectors are famous for their high accuracy and usually used for offline precision systems.

On the other hand single stage detectors are comparatively lagging in accuracy domain when compared to two stage detectors. However Single stage stage detectors outperforms two stage detectors in detection speed. YOLO (You Only Look Once) [49] and SSD (Single Shot Detector) [50] are classic examples of single stage detectors. Over the years single stage detectors have also grown in both accuracy and speed domains. In this work we have used a YOLO detector to detect objects given its ability to detect objects in different sizes over SSD.

YOLO uses a single neural network to output both spatial object bounding boxes and their class probabilities in one go. *figure 3.1* illustrates how YOLO processes images.

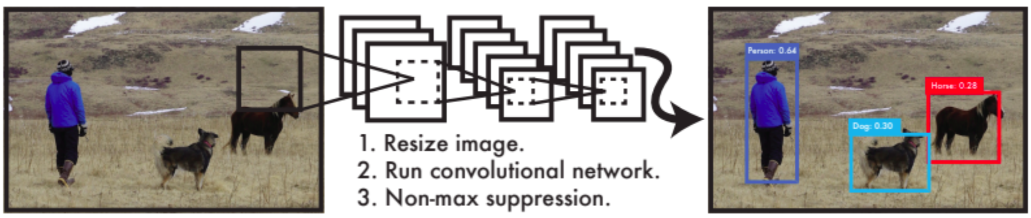

Figure 3.1 - The YOLO Detection System

YOLO has evolved over the years in many versions. YOLO v2 [51] or YOLO9000 was an improvement to YOLO that also could detect over 9000 object classes. YOLO v3 [52] was also an improvement to YOLO where they have made YOLO more faster (3x faster than SSD) while achieving the same accuracy in detection.

This version of YOLO clearly boosted the possibility of running real time object detection applications on affordable hardware. Recently YOLO v4 [53] and v5 were also released with performance improvements to enhance real time useability. *Figure 3.2* illustrates YOLO v4 performance compared to other state-of-art object detectors with MS COCO dataset on Tesla V100.

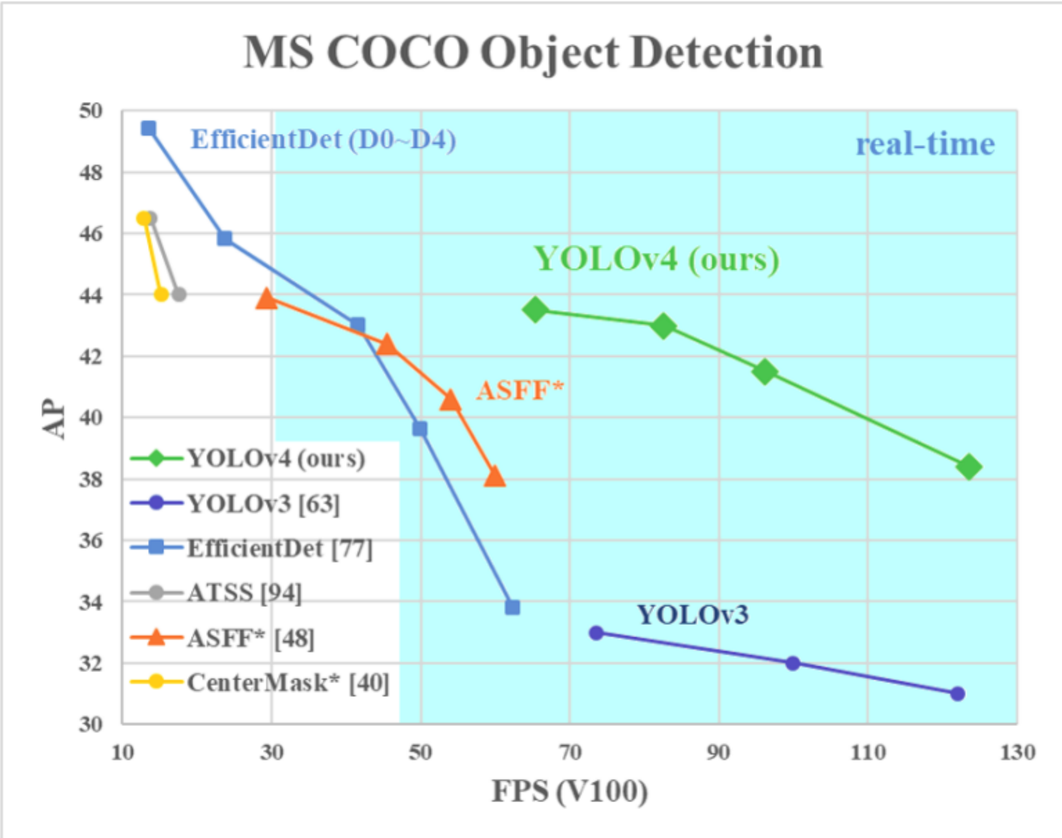

Figure 3.2 - YOLO v4 performance illustration compared to other state-of-art object detection methods

In our work we use YOLO v4 pre-trained weights to transfer-learn the model to detect vehicles accurately.

3.2.1 Transfer Learning

In real life applications, the usage of object detection would mostly be on detecting a particular object class or few classes. However generic pre-trained weights are commonly available for most models. In this case restricting the number of classes instead of using the generic weights would lead to a lesser amount of false detections. In addition to that there could be a large amount of knowledge learned by the pre-trained weights which could be used in our specific detection. Learning this knowledge from scratch not only requires a large dataset related to our application, but will consume a massive amount of time as well. In such scenarios, transfer learning comes in smart where time constrained, limited application dataset availability and pre-trained weights of the model are available for related domain situations. We have used YOLO v4 pre-trained weights which are trained on MS COCO dataset to detect 80 classes and transfer learned with custom vehicle dataset to detect vehicles.

3.3 Technologies used in Single Camera Tracking

In chapter 2 we have discussed the different methods used for single camera tracking in MTMCT solutions. In this work we have used a simple Intersection-over-union(IoU) based object tracking with Kalman filter to track vehicles in a single camera view.

3.3.1 Intersection over Union (IOU)

IOU is a frequently used concept in the computer vision domain. The most common usage of IOU is to measure the quality of object detection. As shown in *figure 3.3*, the green box represents the ground truth and the blue box represents the detected bounding box provided by the detection algorithm/model. Maximum accuracy of detection is achieved when the two bounding boxes are the same. This metric can be mathematically represented by taking the ratio between intersection and union of the bounding boxes. By using this method a normalized metric can be derived regardless of the sizes of the bounding boxes which will be 0 when there is no overlap and 1 if the boxes are the same. *Figure 3.4* illustrates the IOU metric calculation.

3.3.2 IOU for simple object tracking

The concept of IOU can be utilized to implement a simple resource efficient object tracker when the video frame rate is sufficient. spatial information or the object bounding box is given by object detection method and the temporal tracking is done by calculating IOU between bounding boxes of adjacent frames. If a bounding box B_T appeared at t = T frame and a bounding box B_{T+1} appeared at frame t = T+1, we can say B_T and B_{T+1} belong to the same object if the IOU between two boxes exceed a given threshold value α. However there are several drawbacks introduced by the simplicity of the IOU tracker. Major drawback of IOU is inability to handle object path crossovers. Once 2 object paths cross each other, there is a high chance that tracklets may switch identities. Another issue is inability to handle tracklet splitting due to object detection issues.

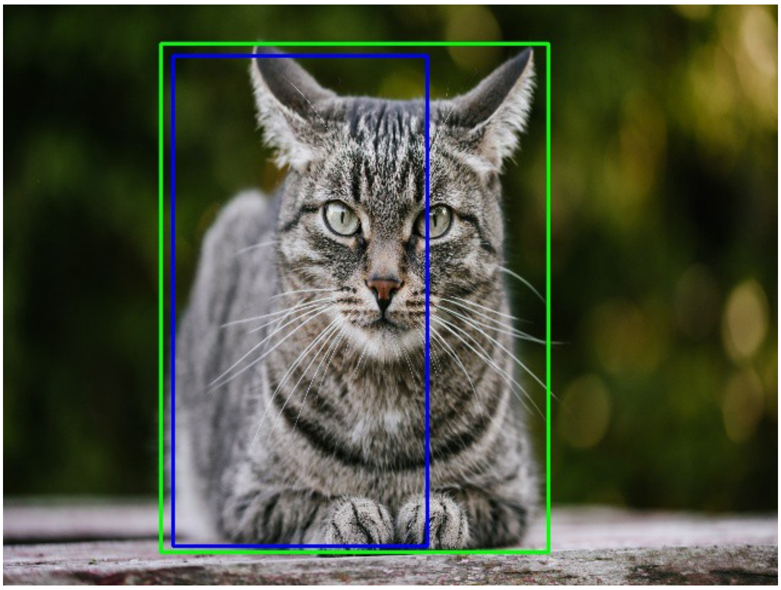

Figure 3.3 - Ground truth and detection bounding boxes of object detection

$$IOU = \frac{Area\ of\ Intersection\ of\ two\ boxes}{Area\ of\ Union\ of\ two\ boxes}$$

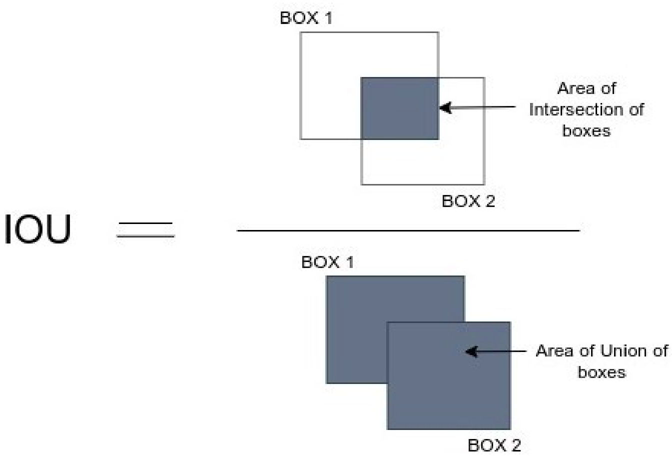

Figure 3.4 - IOU metric calculation

3.4 Technologies used in Re-identification Feature Extraction

In chapter 2 under recent work on re-identification feature extraction, multiple state-of-art methods were discussed. In this work we are using openvino vehicle-reid-0001 pre-trained model distributed under MIT license. The vehicle-reid-0001 model has been developed on the OmniScaleNet backbone. This model takes the complete vehicle image as the input and outputs a 512 float point feature embedding. The model has been trained using VeRi-776 [56] dataset and claims rank1 accuracy over 95%.

3.4.1 Feature distance calculation

In object re-identification using the appearance feature vector, the similarity is measured using a feature domain distance metric. In our work we have evaluated Euclidean and Cosine distances and the Cosine distance metric outperforms Euclidean since Cosine distance has been used in the training process of vehicle-reid-0001 model. Cosine distance is simply defined as the cosine angle between 2 vectors and can be calculated as follows.

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

3.5 Technologies used in Probabilistic Target Search

Once the features are extracted, the next challenge is to perform feature matching of targets detected across multiple cameras. Previous work done on MTMCT has used different technologies to perform this task. In our work we are evaluating the possibility of using probabilistic target search to fulfill this requirement.

3.4 Summary

In this chapter we have discussed the technologies which will be utilized in this work. In the next chapter we will discuss our approach to solve the research problem identified in chapter 2 by using technologies specified in chapter 3.

CHAPTER 4

APPROACH

4.1 Introduction

In chapter 3, we discussed the technologies that can be used to solve our research problem of inefficiency of vehicle re-identification feature signature matching in multi target multi camera tracking (MTMCT) in real time applications. We have come up with a Probabilistic Target Search solution to address the above problem. Rest of the chapter has been structured with the subheading hypothesis, input/output, process, users and features. Figure 4.1 illustrates a high level diagram of the approach.

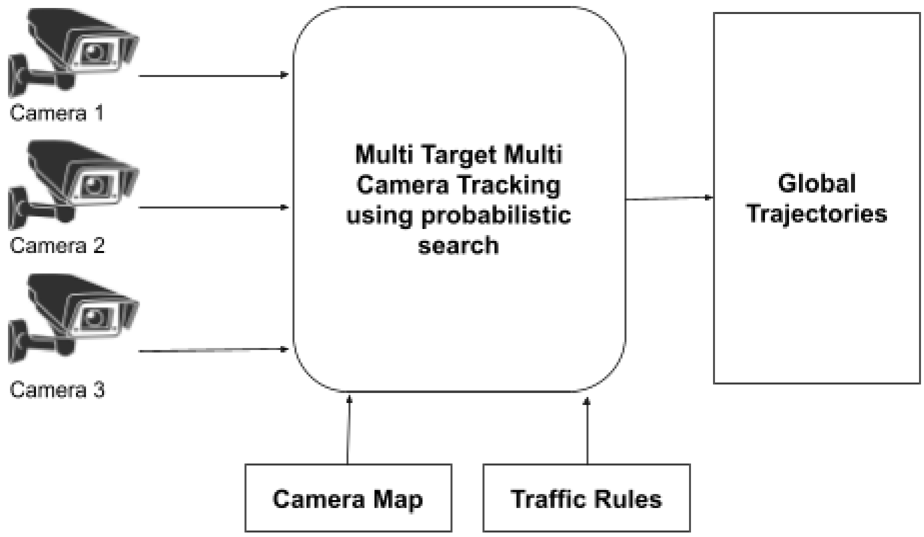

Figure 4.1 - Approach

4.2 Hypothesis

Inter camera object reidentification can be efficiently done by forecasting single camera tracked objects' possible appearance in other cameras using a dynamic camera link model and restructuring the search space in a probability descending order.

4.3 Input

There are three main inputs expected by the system. Namely non-overlapping multi camera surveillance video streams of a geographically bounded camera network where the same object has a chance in reappearing subsequent cameras, a camera map with their geographic location and orientation and general traffic rules applied in relevant field of views.

4.4 Output

Global trajectories of each vehicle in the captured surveillance are given as outputs of the system. In and out times from each camera for the same object can be retrieved by the system.

4.5 Process

Probabilistic target search is a common algorithm for increasing the efficiency when search space is large. This can be adopted in the MTMCT problem by predicting the trajectory of the vehicle. A probability based search graph can be synthesized using prediction of the next potential appearance of a particular target. However this can be further improved by considering traffic volumes in each camera. If the first priority search candidates have different traffic volume through their cameras, a secondary prioritizing can be done since more traffic flows through a particular camera means more probability that a given random vehicle will go through that camera.
The first requirement is predicting the possibilities of the next camera appearance for a given target. However since the appearance cannot be exactly predicted, a probability will be associated with the prediction. Then the matching should be carried out in the decreasing order of probability of appearance.

4.6 Users

The solution can be used for store/city wide security surveillance systems by government and private security frames. Additionally can be utilized for customer journey analysis in retail stores.

4.7 Features

The system can provide real time object tracking and provide the global trajectory of an object. This information can be used in applications such as identifying vehicles or people in city wide surveillance systems and identifying interesting statistics such as the amount of time spent by people in different departments of retail stores.

4.8 Summary

After critically reviewing the MTMCT literature and studying potential technologies usable for solving MTMCT in an efficient manner, This chapter we have discussed our approach to solve the identified research problem. In the next chapter we will be discussing the design of our system.

CHAPTER 5

DESIGN

5.1 Introduction

In Chapter 4 we have presented our approach, MTMCT optimization using probabilistic search. This chapter presents the design of each module used to implement this solution. In order to perform probabilistic target search, there are 2 process sequences to be carried out. The first part is extracting features from the objects appearing in individual cameras. The second part is object matching across multiple cameras. *Figure 5.1* provides a top level diagram of the design.

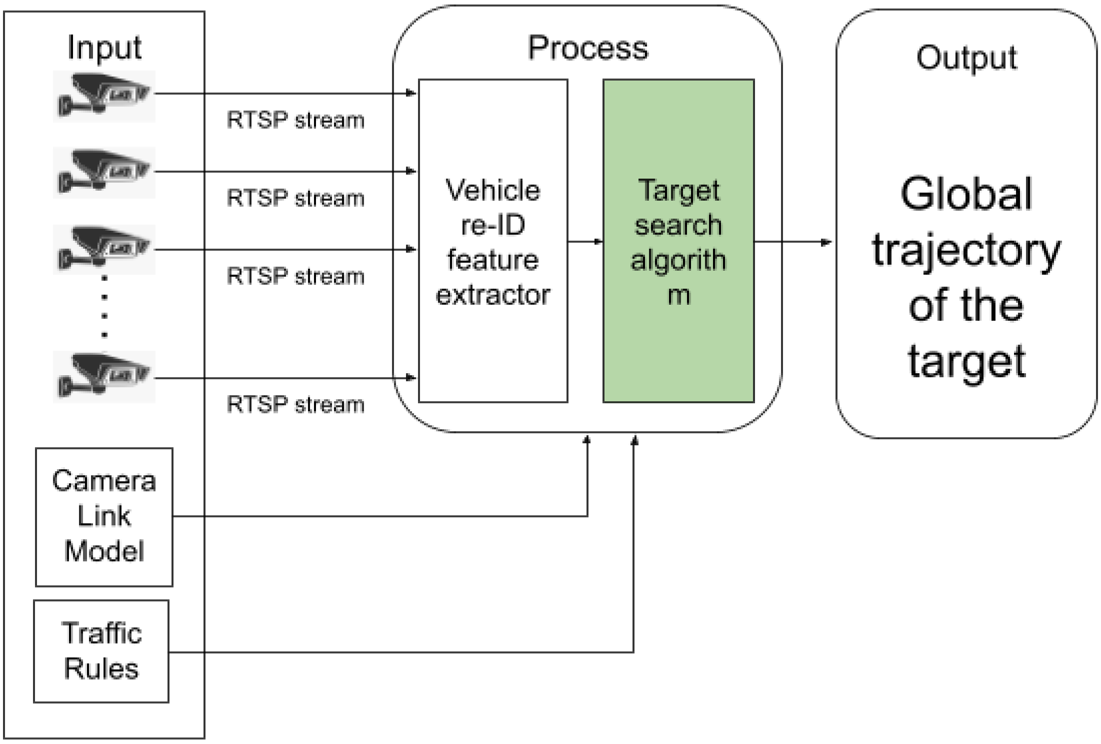

Figure 5.1 - Top level design of the solution

5.2 Vehicle Re-ID Feature Extraction

In this solution, object detection and single camera tracking are also considered as sub modules of feature extraction since the contributions of those modules are for the final feature extraction. Each frame received by a camera stream must go through a sequence of processes in order to generate an accurate feature set for an object.

Figure 5.2 illustrates the process sequence of feature extraction. Expectation of each module of this sequence is discussed in the following sub headings.

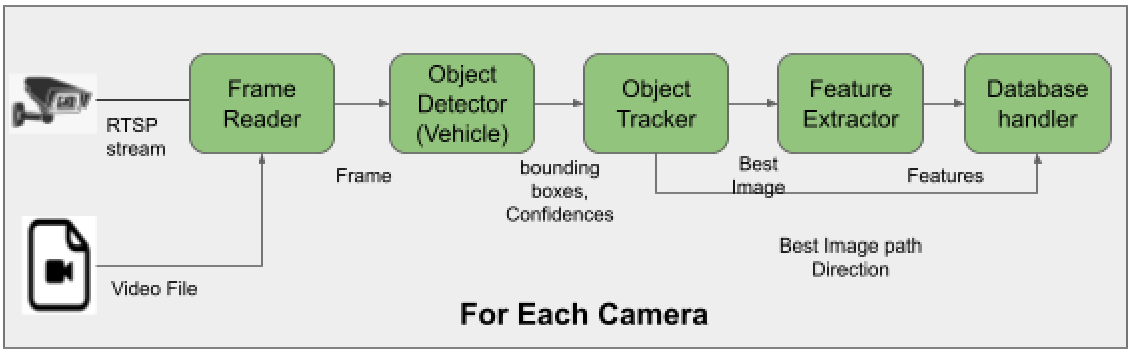

Figure 5.2 - Modules involved for feature extraction

5.2.1 Frame Reader

The system requires capturing image frames from a realtime RTSP stream or a pre-recorded video. A frame reader is required to provide this service as acquiring frames and providing to the next components of the process sequence.

5.2.2 Object Detector

As discussed in the previous chapters, accurate object detection is the key foundation to MTMCT. This module is responsible for receiving a frame from the frame reader, running an object detection method and outputting the bounding boxes of objects to the following modules. However in addition to the bounding box of the object, providing detection confidence will also benefit the later modules in the sequence.

5.2.3 Object Tracker

Object tracker is the key component of the feature extraction sub system. Object tracker is responsible for taking detection bounding boxes from the object detector sub module and providing a unique id throughout the camera field of view for a particular object. In addition to generating the local tracklet, determining objects travel direction out of given possible travel directions as shown in *Figure 5.3* is also a responsibility of this module.

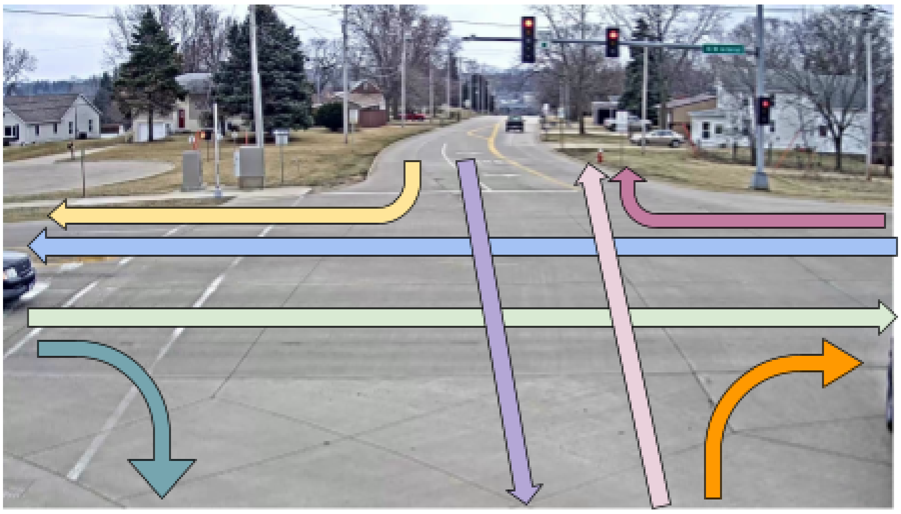

Figure 5.3 - Pre-defined directions of possible object movements

5.2.4 Feature Extractor

Once the tracklets are generated by the object tracker, the appearance feature vector can be extracted using cropped object images. However a tracklet of an object may contain multiple image instances of the object. Which image out of these to be used to extract the appearance feature vector is a tricky question to be answered. In this work we have used the detection with maximum confidence for feature extraction.

5.2.5 Database Handler

After the object goes out of the vicinity of a particular camera view, retrieved data such as detection images, temporal information, object direction, potential next appearances with probabilities, trajectory and the feature vector must be stored in-order to be used by the target search algorithm.

5.3 Target Search Algorithm

By the end of the feature extraction process sequence multiple data points will be stored into a database and available for the target search algorithm. The sole responsibility of the target search algorithm is to compare newly appearing targets with already seen objects and identify whether they are referring to the same object. The high level diagram of the target search algorithm is given in *Figure 5.4*.

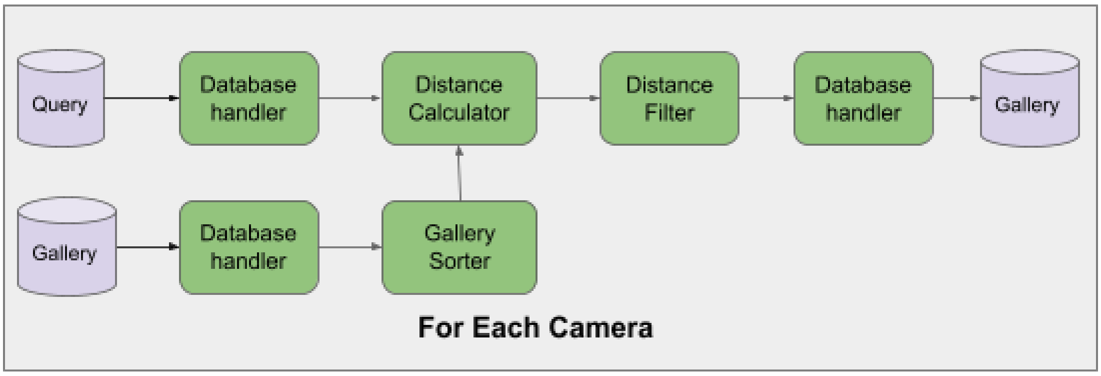

Figure 5.4 - Modules involved for target search

5.3.1 Gallery Sorter

Each camera will have their own Query and Gallery databases. In gallery databases, each entry contains the probability of the given object appearing in the considered camera. Before calculating distances between a query entry and gallery entries, the gallery entries are ordered in descending order of appearance probability. This will allow the process to evaluate most probable galleries first in order to decrease matching time.

5.3.2 Distance Calculator

As we discussed under chapter 3, the similarity between 2 images can be determined using the closeness of the instances in the appearance feature domain. This closeness can be calculated using the vector distance between 2 feature vectors. This module is responsible for calculating feature distance between 2 feature vectors.

5.3.3 Distance Filter

Conventional method of matching 2 entries in a gallery and a query is calculating all the distances with gallery entries and assigning the shortest distance pair as a positive match. However if the target is not in the gallery, the theory query will be matched with the entry with shortest distance and it may lead to a chain of false positives. To avoid this situation we are introducing a classifier to identify if it is a correct match using the calculated distance metric.

5.4 Summary

In this chapter we have discussed the design of the solution to implement our hypothesis. The implementation of the design is discussed in the next chapter

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

In chapter 5 we have discussed the design of our system to implement the approach discussed in chapter 4. In this chapter we will discuss the implementation of the design. As explained in the previous chapter we have designed our solution in a modularized manner. In the implementation also we are planning to maintain the modularity so switching between algorithms where there is more than one method to implement a particular module.

We have selected python as our programming language due to the ease of prototyping and available support. Following diagram in *Figure 6.1* shows the high level representation of implemented modules.

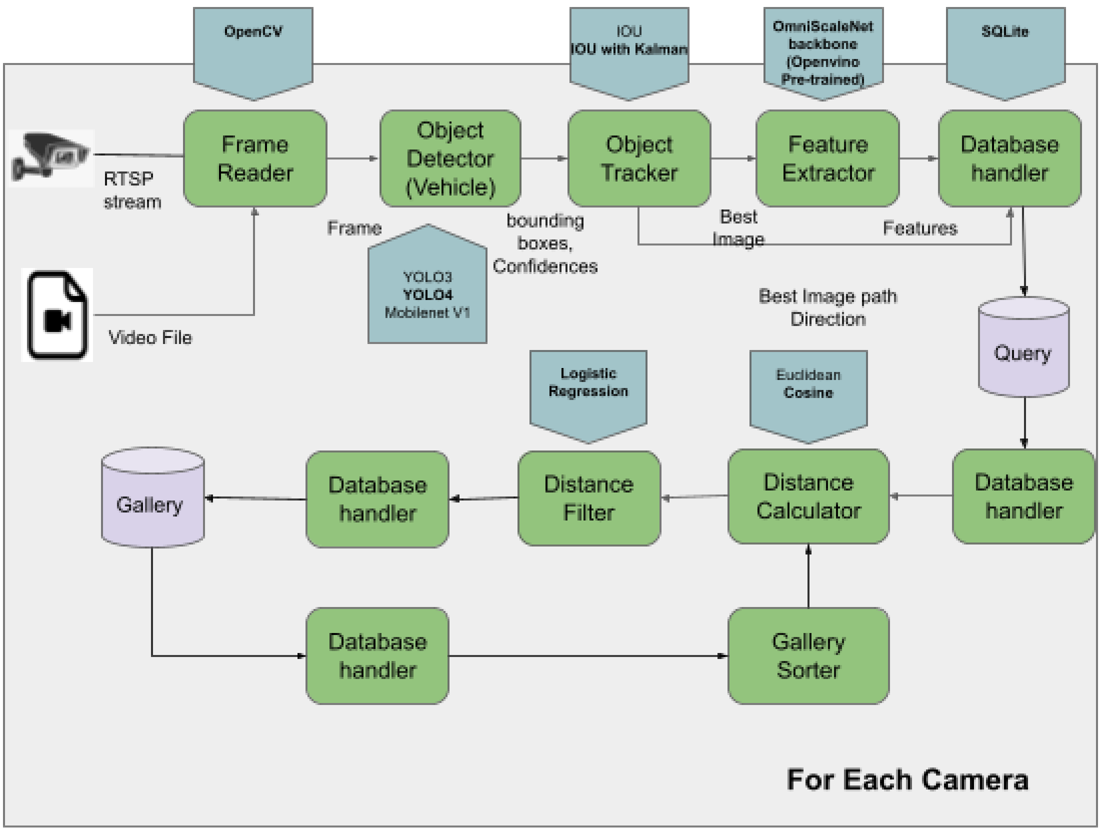

Figure 6.1 - Implementation of modules of the proposed system

6.2 Implementation of Feature Extractor Modules

6.2.1 Frame Reader

Frame reader was implemented using OpenCV in-built functions to read a frame from a video file or real time RTSP stream. In the video frame reader initially video is listed to a queue and each frame is provided for the system for processing on request one after another. However this becomes challenging in real time scenarios since the RTSP stream frame rate and the processing frame needs to be matched for smooth operation. A frame suppressor is introduced to match the supply and demand between the stream and the application.

6.2.2 Object Detection

We have done our implementation as a vehicle MTMCT solution. Once a frame is acquired by the system it is sent to the object detection module as a RGB matrix and the object detection modules take this input to detect objects.

In this work we have tried out 3 object detection models YOLO3, YOLO4 and MobileNetV1. YOLO versions were transferred to detect vehicles using darknet(https://pjreddie.com/darknet/yolo/) and Cars dataset introduced in 3D Object Representations for Fine-Grained Categorization[XXX]. An openvino pre-trained vehicle detection model based on MobileNetV1 is also evaluated for the Object detection model. However after testing the models for several videos, the transfer learned YOLO v4 model using Cars dataset was selected for object detection.

6.2.3 Object Tracker

As we discussed in the design chapter, Object tracking carries the major weight of the feature extraction sub system. 2 main responsibilities are handled by the object tracker, namely,

1. Tracking detected objects throughout a camera field of view
2. Determining the object travel direction

From the object detection module, the bounding boxes of detected objects are received to the object tracking module. A simple Intersection over Union(IoU) tracker is implemented to perform the intra-camera object tracking. Once T = t+1 bounding box coordinates are received, IoU between each bounding box pair in T = t and t+1 are calculated using the equation shown in figure 6.2.

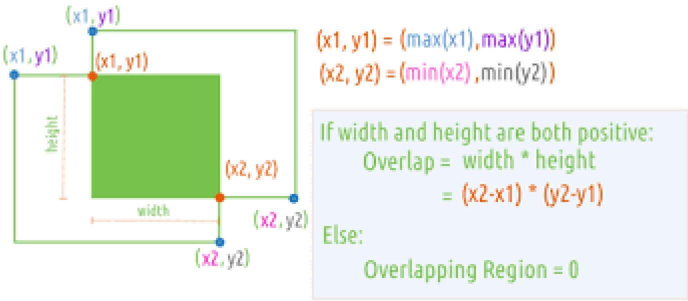

Figure 6.2 - IoU calculation between 2 bounding boxes

If the IoU value is exceeding a certain threshold we relate the 2 boxes in the temporal domain. However when the region of interest is crowded, identity switching could happen due to object trajectory overlapping. We have introduced a Kalman filter to avoid this situation. In this case we predict the location of bounding boxes appearing at T = t + 1 using the bounding boxes at T = t and the Kalman filter. Then the IoU is calculated between the predicted box and the actual box. This method helps to improve accuracy over identity switching issues.

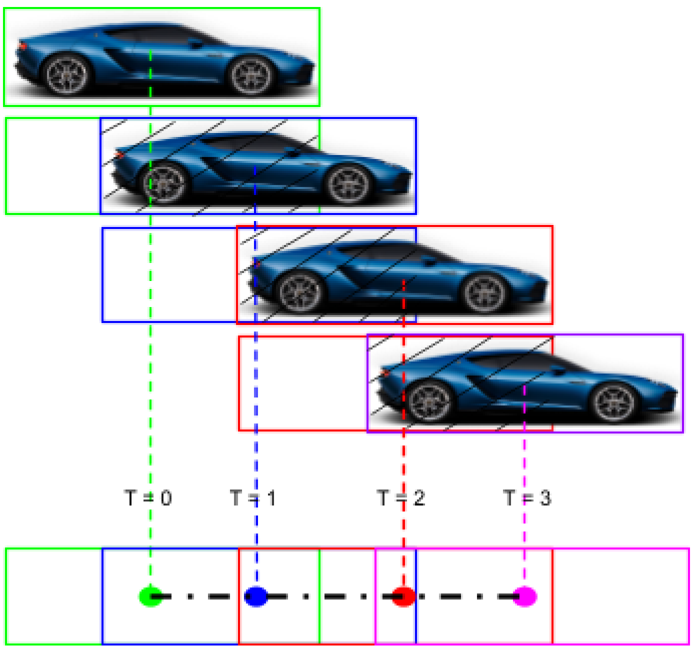

Figure 6.3 - IoU tracking mechanism

The next responsibility of the object tracker is to determine the direction of the object to predict the cameras it will appear next. This is done using the trajectory obtained in the above process and a pre demarcated map on possible object movements according to the traffic rules. Figure 6.4 illustrates an example of direction demarcation and a simple case of direction detection.

PE1 → PE5 (Direction 1)
PE4 → PE0 (Direction 2)
PE2 → PE6 (Direction 3)
PE7 → PE3 (Direction 4)
PE2 → PE0 (Direction 5)
PE4 → PE3 (Direction 6)
PE7 → PE5 (Direction 7)
PE1 → PE6 (Direction 8)

```
If in_polygon = PE1 and out_polygon = PE5 then:
        probability of being direction 1 = 1
        Probability of being other direction = 0
Else if in_polygon = PE1 and out_polygon = UNKNOWN:
        Probability of being a direction starting with PE1 = 1 / No of direction starting with PE1
        Probability of being other direction = 0
Else if in_polygon = UNKNOWN and out_polygon = PE5:
        Probability of being a direction ending with PE5 = 1 / No of direction ending with PE5
        Probability of being other direction = 0
```

Figure 6.4 - Direction identification

6.2.4 Feature Extraction

Before extracting the features, most suitable detection out of all seen instances must be selected. For this work we have used detection with highest detection confidence as the feature extraction image.

After selecting the image, an OmniScaleNet based Openvino pre-trained model(https://docs.openvino.ai/latest/omz_models_model_vehicle_reid_0001.html) was used to extract features from the images. Input and outputs of the models are shown in *Figure 6.4*.

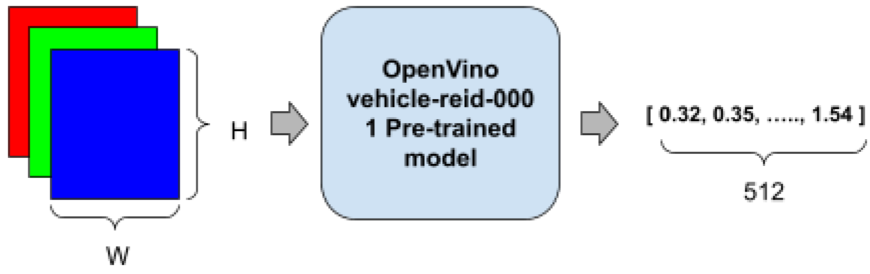

Figure 6.5 - vehicle-reid-0001 model inputs and outputs

6.2.5 Database handler

Typically any re-identification related problem consists of 2 databases, namely the gallery and the query. Gallery database includes past seen data where Query database includes newly seen data which needs to be matched with the gallery. Most common method is having a global gallery and queries for each camera. However in our implementation we restrict the gallery for each camera by determining the possibility of a particular object appearing in the considered camera. Figure 6.6 illustrates the traditional gallery query structure and the proposed gallery query structure.

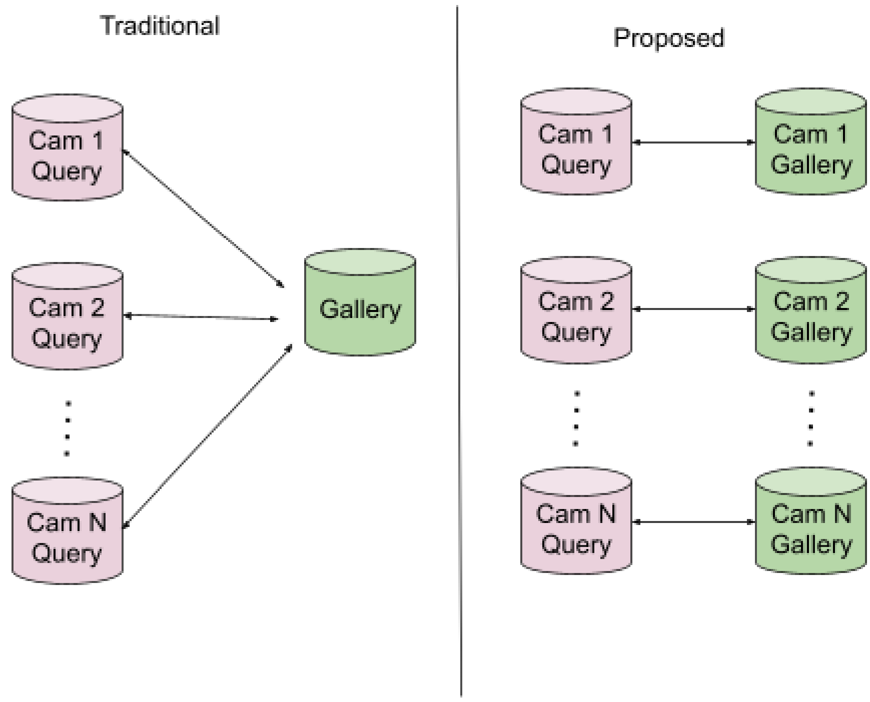

Figure 6.6 - Database structure implementation

The implementation of databases was done using SQLite 3 and each galley and query was implemented as a separate database table.

6.3 Implementation of Target Search Modules

6.3.1 Distance Calculator

The similarity between 2 feature vectors are calculated using cosine distance. In the initial stages both Cosine and Euclidean distance metrics were evaluated. However, considering the fact that the feature extraction model has also used cosine distance metric in the training and more accurate linear separability in distance filter implementation.

6.3.2 Distance Filter

As we discussed in the design chapter, pairing the lowest distant pair from gallery and query alone cannot guarantee a positive matching. A distance filter can be implemented to avoid this issue to a considerable extent. When this is implemented, 2 conditions must be satisfied for a pair to be considered as a positive match.

1. The pair must have the lowest cost in all the possible pairs
2. The distance must be pass the minimum true positive threshold from the filter

A logistic regression model was trained using distance metric as the input and whether the distance is a possible true positive or not as the output.

6.4 Datasets Used

5th AI City Challenge - City-Scale Multi-Camera Vehicle Tracking Dataset (https://www.aicitychallenge.org/)

The dataset consists of annotated traffic surveillance camera video dataset. The dataset consists of 6 locations with multiple cameras in each location to get the maximum coverage. Figure 6.7 shows a high level understanding on locations and camera placements.

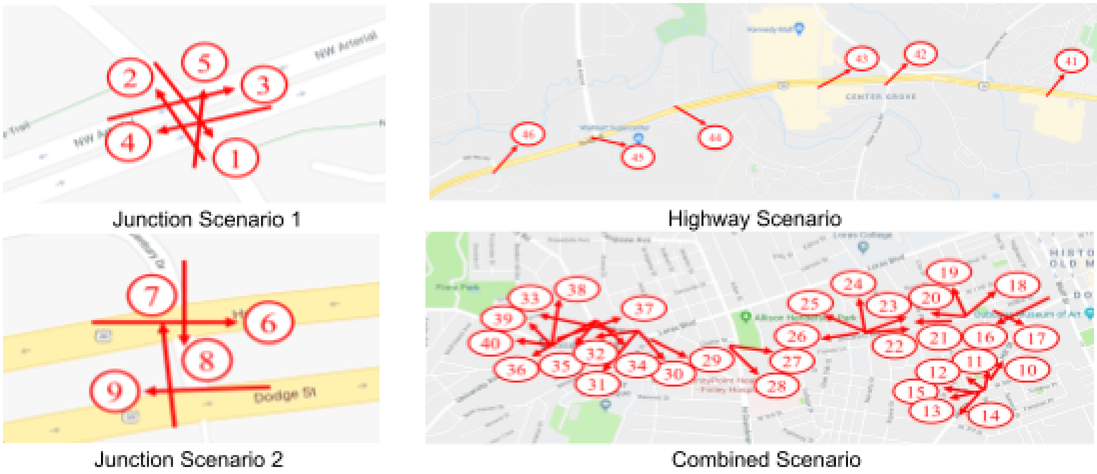

Figure 6.7 - City-Scale Multi-Camera Vehicle Tracking Dataset Overview

Cars Dataset (http://ai.stanford.edu/~jkrause/cars/car_dataset.html)

VeRi Dataset (https://vehiclereid.github.io/VeRi/)

6.5 Summary

In this chapter we have discussed the implementation of our proposed system. The source code of this implementation can be found in the Appendix. In the next chapter we will discuss the evaluation process and the results of the proposed system.

CHAPTER 7

EVALUATION

7.1 Introduction

In the previous chapter we have discussed the implementation of our design using various technologies and frameworks. In this chapter we will discuss the outcomes of our system.

The MTMCT can be evaluated in 2 major domains,

1. Re-Identification accuracy
2. System performance

Given that our research problem was "how to make vehicle re-identification feature signature matching efficient in multi target multi camera tracking" our target was to develop a system with increased performance. Usually the metric to measure the accuracy in object re-identification is the IDF1 score and the metric to measure the performance is processable frames per second using the proposed system.

7.2 Evaluation Strategy

7.2.1 Experiment Setup

Evaluation experiment is done using the City-Scale Multi-Camera Vehicle Tracking Dataset - Junction Scenario. The system takes 4 camera video inputs and processes each in a separate sub-process. Vehicle global tracklets are saved as the best image captured in each camera. Figure 7.1 shows the structure of experimental setup.

7.2.2 Accuracy Testing Strategy

As mentioned earlier the identification F1 score or the IDF1 score [54] has become the most widely used metric for evaluating the accuracy of MTMCT solutions.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

TP	=	Number of True Positives
FP	=	Number of False Positives
FN	=	Number of False Negatives

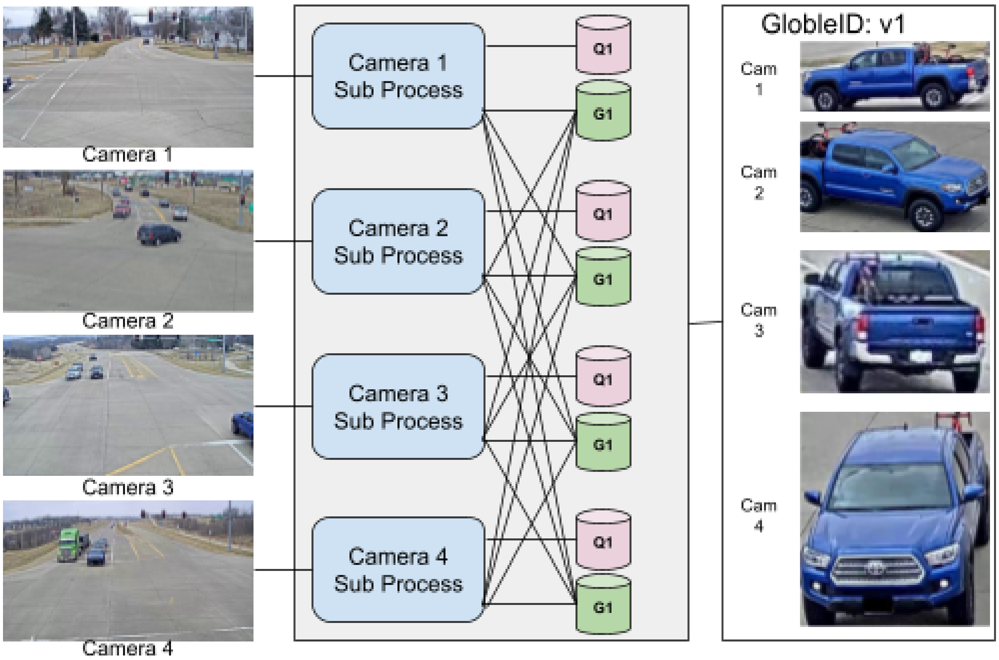

Figure 7.1 - Experimental Setup

In the proposed system, outputs are recorded into a folder structure and each folder name by the global id of a vehicle. Inside the folder the relevant vehicle images are stored with the appeared camera IDs. However inside this folder, False Positives can be seen and missed identifications can be considered as False Negatives. Since our evaluation is carried out using a junction setup, all 4 cameras should see every vehicle. Therefore each camera must have an identification on each vehicle in the ground truth. Table 7.1 shows a few examples in the output and these values can be used to calculate the IDF1 score for the system

Table 7.1 - System Output Examples

Vehicle ID	TP	FP	FN
v1	   	-	-
v5	  		-
v22		-	Missed detection by Cam 2

	 		

7.2.2 Performance Testing Strategy

The performance of the proposed system can be measured by determining the maximum attainable frames processed by the system for a unit time. However this metric will heavily depend on the traffic density since increasing the number of vehicles in the camera vicinity will increase the size of the search space. Because of this property the evaluation is carried out to find processed FPS per camera using single core performance of an Intel i9 9th generation processor and time taken to search a target in traditional database structure and proposed database structure as shown in *Figure 6.6*.

7.3 Results

7.3.1 Accuracy Testing

City-Scale Multi-Camera Vehicle Tracking Dataset - Junction Scenario has 3 minute and 15 second video recordings from each camera. During this time, the system was capable of capturing 71 distinct vehicles. The output of the system is given in Appendix I. A summary of results can be given as follows.

Table 7.2 - Accuracy test result summary

Method	True Positive	False Positive	False Negative	F1 Score
Using Probabilistic Target Search	**150**	**72**	**62**	**0.6912**
Without Using Probabilistic Target Search	**182**	**59**	**43**	**0.7811**

7.3.2 Performance Testing

The 4 camera videos used, have a maximum of 3 minute and 15 seconds overlap in recordings and the videos are recorded with 10 FPS. Therefore around 1950 frames are available for processing per video. Each video is processed by a separate python process which acquires a single dedicated core for each. However the processing FPS may vary due to available queries for each process. To rule out this variance, we have taken the average of 4 cores processing 1950 x 4 frames by using the equation below. Results obtained during the experiment are shown in Table 7.3.

Table 7.3 - Performance test result summary

Method	No of videos $\mathbf{N_v}$	Video FPS $\mathbf{FPS_{rec}}$	Video length(s) **T**	Total Time Elapsed(s) $\mathbf{T_{elp}}$	No of Cores $\mathbf{N_c}$	Average FPS per core $\mathbf{(N_v \times FPS_{rec} \times T) / (T_{elp} \times N_c)}$
Using Probabilistic Target Search	4	10	195	291.04	4	6.7
Without Using Probabilistic Target Search	4	10	195	331	4	5.89

7.4 Summary

In this chapter we have formulated the evaluation criterias of the newly proposed system for solving MTMCT and gathered the results using a junction scenario in City-Scale Multi-Camera Vehicle Tracking Dataset using an experimental design. The next chapter consists of the over discussion on the results obtained in this chapter and the over conclusion of the project.

CHAPTER 8

CONCLUSION AND FURTHER WORK

8.1 Introduction

In this research we have addressed the MTMCT problem using probabilistic target search in-order to increase the efficiency of target matching algorithm. Our hypothesis and the approach was to use probabilistic target search to make the target matching process faster. The design of the system, Implementation and the results of the experimental setup has been discussed in the previous chapters. In this chapter the overall conclusion regarding the achievements of the project will be discussed. In addition to that the limitations and further work to this project have also been discussed in this chapter.

8.2 Conclusion

8.2.1 Achievement of Project Objectives

This research process relied on achieving objectives defined in the introduction chapter to develop a MTMCT solution using probabilistic target search. All the mentioned objectives were covered in the process followed.

A critical literature study and review done to identify past and recent approaches to solve MTMCT problems using different techniques. During this step the advantages and disadvantages of each approach is studied in depth and identified alternative solutions given by subsequent research. At the end of the critical review of MTMCT literature, inadequate research on increasing MTMCT performance has been identified as a research problem.

MTMCT target matching carried out for the entire detection space was identified as a common implementation across most of the past work. This was identified as a major permanence hindrance factor. Probabilistic target search was identified as a potential alternative to reduce the search time while retaining the accuracy. The theories required for using probabilistic target search for MTMCT were studied in depth to fulfill the second objective.

Then an MTMCT using a feature matching algorithm using a trajectory based probability hierarchy was designed and implemented as discussed in chapter 5 and 6. Finally an experimental system was designed to evaluate the proposed solution using the AI City Challenge 2021 dataset.

8.2.2 Overall Conclusion

The target of this research was to find a method to improve the target matching efficiency while retaining the re-identification accuracy. The newly proposed system does this by predicting a detected object's next appearance camera, matching in sequenced manner with descending probability of appearance order and a distance filtering process. A typical database structure of MTMCT where queries are cross matched with a single gallery with all past seen tracklets was considered as the baseline for the evaluation.

The proposed system has shown increased average processing FPS value which is 6.7 FPS/core compared to the baseline value 5.89 FPS/core. This is a 13.75% improvement from the baseline model. A trade-off between accuracy and performance is expected since the proposed solution terminates the search when the distance filter classifies a pair as a positive match. However due to an increased number of False Positives and False Negatives the IDF1 score has decreased from the baseline value 0.78 to 0.69. This is basically due to the search termination due to false classification from the distance filter. Therefore we have lost 11.54% of the IDF1 score to gain a 13.75% gain in the performance.

However by considering the overall evaluation we can conclude that our hypothesis, Inter camera object reidentification can be efficiently done by forecasting single camera tracked objects' possible appearance in other cameras using a dynamic camera link model and restructuring the search space in a probability descending order is proved.

8.3 Limitations and Further Works

The target behind increasing the efficiency of MTMCT is to use this for real-time applications. Even Though we have gained the FPS/core value by 13.75%, still the attainable FPS value on latest hardware is 6.7 FPS/core. However, given that this is used for tracking vehicles which are moving relatively fast, 6.7 FPS might not be sufficient for real-time processing for smooth tracking.

8.2 Summary

With this chapter we have concluded our work by evaluating the completion of objectives, interpretation of results and limitations of the work.

REFERENCES

[1] Arce-Ruiz, Rosa & Baucells, Neus & Moreno Alonso, Concepcion. (2016). Smart Mobility in Smart Cities. 10.4995/CIT2016.2016.3485..

[2] Quin Cai and Jake K Aggarwal. Automatic tracking of human motion in indoor scenes across multiple synchronized video streams. InSixth International Con-ference on Computer Vision (IEEE Cat. No. 98CH36271), pages 356–362. IEEE,1998.

[3] J. Kang, I. Cohen and G. Medioni, "Tracking objects from multiple and moving cameras," IEE Intelligent Distributed Surveillance Systems, 2004, pp. 31-35, doi: 10.1049/ic:20040094

[4] Laura Leal-Taixe, Gerard Pons-Moll, and Bodo Rosenhahn. Branch-and-price global optimization for multi-view multi-target tracking. In2012 IEEE Confer-ence on Computer Vision and Pattern Recognition, pages 1987–1994. IEEE, 2012

[5] Longyin Wen, Zhen Lei, Ming-Ching Chang, Honggang Qi, and Siwei Lyu.Multi-camera multi-target tracking with space-time-view hyper-graph. International Journal of Computer Vision, 122(2):313–333, 2017.

[6] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people track-ing via hierarchical trajectory composition. InProceedings of the IEEE Confer-ence on Computer Vision and Pattern Recognition, pages 4256–4265, 2016.

[7] Pierre Baqu ́e, Franc ̧ois Fleuret, and Pascal Fua. Deep occlusion reasoning for multi-camera multi-target detection. InProceedings of the IEEE InternationalConference on Computer Vision, pages 271–279, 2017.

[8] Qi Zhang and Antoni B Chan. Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns. InProceedings of the IEEE Conference onComputer Vision and Pattern Recognition, pages 8297–8306, 2019.

[9] H.-M. Hsu, J. Cai, Y. Wang, J.-N. Hwang, and K.-J. Kim, “Multi-Target Multi-Camera Tracking of Vehicles using Metadata-Aided Re-ID and Trajectory-Based Camera Link Model,” IEEE Trans. Image Process., vol. 30, pp. 5198–5210, 2021, doi: 10.1109/TIP.2021.3078124.

[10] Y. He, X. Wei, X. Hong, W. Shi, and Y. Gong, “Multi-Target Multi-Camera Tracking by Tracklet-to-Target Assignment,” IEEE Trans. Image Process., vol. 29, pp. 5191–5205, 2020, doi: 10.1109/TIP.2020.2980070.

[11] Liu, Chong & Zhang, Yuqi & Luo, Hao & Tang, Jiasheng & Chen, Weihua & Xu, Xianzhe & Wang, Fan & Li, Hao & Shen, Yi-Dong. (2021). City-Scale Multi-Camera Vehicle Tracking Guided by Crossroad zones.

[12] Peng Li, Jiabin Zhang, Zheng Zhu, Yanwei Li, Lu Jiang, Guan Huang (2019) “State-aware Re-identification Feature for Multi-target Multi-camera Tracking”

[13] Chen, Junru & Geng, Shiqing & Yan, Yongluan & Huang, Danyang & Liu, Hao & Li, Yadong. (2021). Vehicle Re-identification Method Based on Vehicle Attribute and Mutual Exclusion Between Cameras.

[14] Ha, Ilkyu. (2018). A Probabilistic Target Search Algorithm Based on Hierarchical Collaboration for Improving Rapidity of Drones. Sensors. 18. 2535. 10.3390/s18082535.

[15] Xiaogang Wang,Intelligent multi-camera video surveillance: A review,Pattern Recognition Letters, Volume 34, Issue 1,2013, Pages 3-19,ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2012.07.005.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

[17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[18] Jiao, Licheng & Zhang, Fan & Liu, Fang & Yang, Shuyuan & Li, Lingling & Feng, Zhixi & Qu, Rong. (2019). A Survey of Deep Learning-Based Object Detection. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2939201.

[19] J. Ye et al., "A Robust MTMC Tracking System for AI-City Challenge 2021," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 4039-4048, doi: 10.1109/CVPRW53098.2021.00456.

[20]K. -S. Yang, Y. -K. Chen, T. -S. Chen, C. -T. Liu and S. -Y. Chien, "Tracklet-refined Multi-Camera Tracking based on Balanced Cross-Domain Re-Identification for Vehicles," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 3978-3987, doi: 10.1109/CVPRW53098.2021.00449.

[21]K. Shim, S. Yoon, K. Ko and C. Kim, "Multi-Target Multi-Camera Vehicle Tracking for City-Scale Traffic Management," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 4188-4195, doi: 10.1109/CVPRW53098.2021.00473.

[22]A. Specker, D. Stadler, L. Florin and J. Beyerer, "An Occlusion-aware Multi-target Multi-camera Tracking System," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 4168-4177, doi: 10.1109/CVPRW53098.2021.00471.

[23]Y. -L. Li, Z. -Y. Chin, M. -C. Chang and C. -K. Chiang, "Multi-Camera Tracking By Candidate Intersection Ratio Tracklet Matching," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 4098-4106, doi: 10.1109/CVPRW53098.2021.00463.

[24]Liu, Chong & Zhang, Yuqi & Luo, Hao & Tang, Jiasheng & Chen, Weihua & Xu, Xianzhe & Wang, Fan & Li, Hao & Shen, Yi-Dong. (2021). City-Scale Multi-Camera Vehicle Tracking Guided by Crossroad Zones.

[25]Xu, Jian & Bo, Chunjuan & Wang, Dong. (2021). A novel multi-target multi-camera tracking approach based on feature grouping. Computers & Electrical Engineering. 92. 107153. 10.1016/j.compeleceng.2021.107153.

[26]S. You, H. Yao and C. Xu, "Multi-Target Multi-Camera Tracking With Optical-Based Pose Association," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 8, pp. 3105-3117, Aug. 2021, doi: 10.1109/TCSVT.2020.3036467.

[27]Hsu, Hung-Min & Wang, Yizhou & Hwang, Jenq-Neng. (2020). Traffic-Aware Multi-Camera Tracking of Vehicles Based on ReID and Camera Link Model.

[28]P. Li, J. Zhang, Z. Zhu, Y. Li, L. Jiang and G. Huang, "State-Aware Re-Identification Feature for Multi-Target Multi-Camera Tracking," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1506-1516, doi: 10.1109/CVPRW.2019.00192.

[29]Hou, Y., Zheng, L., Wang, Z., & Wang, S. (2019). Locality Aware Appearance Metric for Multi-Target Multi-Camera Tracking. ArXiv, abs/1911.12037.

[30]Li, P., Li, G., Yan, Z., Li, Y., Lu, M., Xu, P., Gu, Y., Bai, B., & Zhang, Y. (2019). Spatio-temporal Consistency and Hierarchical Matching for Multi-Target Multi-Camera Vehicle Tracking. CVPR Workshops.

[31]Hsu, H., Huang, T., Wang, G., Cai, J., Lei, Z., & Hwang, J. (2019). Multi-Camera Tracking of Vehicles based on Deep Features Re-ID and Trajectory-Based Camera Link Models. CVPR Workshops.

[32]Y. He, J. Han, W. Yu, X. Hong, X. Wei and Y. Gong, "City-Scale Multi-Camera Vehicle Tracking by Semantic Attribute Parsing and Cross-Camera Tracklet Matching," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2456-2465, doi: 10.1109/CVPRW50498.2020.00296.

[33]N. Peri et al., "Towards Real-Time Systems for Vehicle Re-Identification, Multi-Camera Tracking, and Anomaly Detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2648-2657, doi: 10.1109/CVPRW50498.2020.00319.

[34]Ye, Shiping & Bohush, Rykhard & Chen, Huafeng & Zakharava, I. & Ablameyko, Sergey. (2020). Person Tracking and Reidentification for Multicamera Indoor Video Surveillance Systems. Pattern Recognition and Image Analysis. 30. 827-837. 10.1134/S1054661820040136.

[35]LIU Guanqun, LI Ting. Research on Multi-camera Multi-target Tracking Method Based on Hierarchical Relational Model of Trajectory Tree[J]. Journal of Frontiers of Computer Science and Technology, 2020, 14(6): 1036-1044.

[36]X. Zhang and E. Izquierdo, "Real-Time Multi-Target Multi-Camera Tracking with Spatial-Temporal Information," 2019 IEEE Visual Communications and Image Processing (VCIP), 2019, pp. 1-4, doi: 10.1109/VCIP47243.2019.8965845.

[37]Styles, Olly & Guha, Tanaya & Sanchez, Victor & Kot, Alex. (2020). Multi-Camera Trajectory Forecasting: Pedestrian Trajectory Prediction in a Network of Cameras.

[38]Peng, Jinlong & Wang, Tao & Lin, Weiyao & Wang, Jian & See, John & Wen, Shilei & Ding, Erui. (2020). TPM: Multiple Object Tracking with Tracklet-Plane Matching. Pattern Recognition. 107. 107480. 10.1016/j.patcog.2020.107480.

[39]Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. 2017 IEEE International Conference on Image Processing (ICIP), 3645-3649.

[40]Wang, Zhongdao & Zheng, Liang & Liu, Yixuan & Wang, Shengjin. (2019). Towards Real-Time Multi-Object Tracking.

[41] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person reidentification. In Proceedings of the IEEE Conference on Computer Vision and pattern Recognition, pages 3539–3548, 2017

[42]Anton Milan, Konrad Schindler, and Stefan Roth. Multitarget tracking by discrete-continuous energy minimization. IEEE transactions on pattern analysis and machine intelligence, 38(10):2054–2068, 2016

[43] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In CVPR Workshop (CVPRW) on the AI City Challenge, 2018.

[44] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity:Multi-object tracking with trackletnet. arXiv preprint arXiv:1811.07258, 2018.

[45]R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, June 2014.

[46]R. Girshick, "Fast r-cnn," in 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448, Dec 2015.

[47]S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 1137–1149, June 2017.

[48]K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask r-cnn," in 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980– 2988, Oct 2017.

[49]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, June 2016.

[50]W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in Computer Vision – ECCV 2016 (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 21–37, Springer International Publishing, 2016.

[51]J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525, July 2017.

[52]J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," CoRR, vol. abs/1804.02767, 2018.

[53]Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao , "YOLOv4: Optimal Speed and Accuracy of Object Detection" arXiv preprint arXiv: 2004.10934, 2020 - arxiv.org.

[54]Ristani, Ergys & Solera, Francesco & Zou, Roger & Cucchiara, Rita & Tomasi, Carlo. (2016). Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking.

[55]Y. Sun, X. Zhou, L. Sun and S. Chen, "Vehicle Activity Analysis Based on ANPR System," 2014 12th IEEE International Conference on Embedded and Ubiquitous Computing, 2014, pp. 89-96, doi: 10.1109/EUC.2014.22.

[56]Liu, Xinchen & Liu, Wu & Mei, Tao & Ma, Huadong. (2016). A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance. 9906. 869-884. 10.1007/978-3-319-46475-6_53.

APPENDIX

Appendix I: Results

Results with probabilistic target search

Vehicle ID	Ground Truth	True Positives	False Positives	False Negatives
v1	4	4		
v2	4	1		3
v3	4	2		2
v4	4	1		3
v5	4	1	3	
v6	4	1	2	1
v7	4	2	2	
v8	4	4		
v9	4	1		3
v10	4	1	2	1
v11	4	1	2	1
v12	4	2		2
v13	4	2	2	
v14	4	3	1	
v15	4	2	2	
v16	4	1	3	
v17	4	2	1	1
v18	4	2	1	1
v19	4	4		
v20	4	3	1	
v21	4	2	2	
v22	4	2	1	1
v23	4	2	2	
v24	4	1	3	
v25	4	2	2	
v26	4	4		
v27	4	1	3	
v28	4	1	1	2
v29	4	4		
v30	4	2	2	
v31	4	2	1	1

v32	4	1		3
v33	4	2	2	
v34	4	2	2	
v35	4	2		2
v36	4	4		
v37	4	1		3
v38	4	4		
v39	4	2	2	
v40	4	3	1	
v41	4	3	1	
v42	4	2	2	
v43	4	4		
v44	4	2	2	
v45	4	2	1	1
v46	4	2	2	
v47	4	4		
v48	4	2		2
v49	4	1		3
v50	4	2	2	
v51	4	4		
v52	4	2	2	
v53	4	3	1	
v54	4	2	2	
v55	4	4		
v56	4	2	1	1
v57	4	2	2	
v58	4	1		3
v59	4	1		3
v60	4	2	2	
v61	4	2	1	1
v62	4	3		1
v63	4	2	1	1
v64	4	3		1
v65	4	2	2	
v66	4	2	2	
v67	4	1		3
v68	4	1		3

v69	4	1		3
v70	4	1		3
v71	4	1		3
Sum	**284**	**150**	**72**	**62**

Results with standard method

Vehicle ID	Ground Truth	True Positives	False Positives	False Negatives
v1	4	4		
v2	4	4		
v3	4	3		1
v4	4	1		3
v5	4	3	1	
v6	4	1	2	1
v7	4	2	2	
v8	4	4		
v9	4	1		3
v10	4	1	2	1
v11	4	3		1
v12	4	2		2
v13	4	2	2	
v14	4	3	1	
v15	4	2	2	
v16	4	4		
v17	4	2	1	1
v18	4	2	1	1
v19	4	4		
v20	4	3	1	
v21	4	2	2	
v22	4	3		1
v23	4	3	1	
v24	4	3	1	
v25	4	2	2	
v26	4	4		
v27	4	1	3	
v28	4	1	1	2
v29	4	4		

v30	4	2	2	
v31	4	2	1	1
v32	4	1		3
v33	4	2	2	
v34	4	2	2	
v35	4	2		2
v36	4	4		
v37	4	4		
v38	4	4		
v39	4	2	2	
v40	4	3	1	
v41	4	3	1	
v42	4	2	2	
v43	4	4		
v44	4	2	2	
v45	4	2	1	1
v46	4	2	2	
v47	4	4		
v48	4	2		2
v49	4	1		3
v50	4	2	2	
v51	4	4		
v52	4	2	2	
v53	4	3	1	
v54	4	2	2	
v55	4	4		
v56	4	2	1	1
v57	4	4		
v58	4	1		3
v59	4	1		3
v60	4	2	2	
v61	4	2	1	1
v62	4	3		1
v63	4	2	1	1
v64	4	3		1
v65	4	2	2	
v66	4	2	2	

v67	4	1		3
v68	4	4		
v69	4	4		
v70	4	4		
v71	4	4		
Sum		182	59	43