Key-Point Based Tracking for Illegally Parked Vehicle Detection

Xing Gao

A thesis submitted to the University of Sussex For the degree of Doctor of Philosophy Nov 2021

Declaration

I hereby declare that this thesis has not been and will not be submitted in whole or in part to another University for the award of any other degree.

Signature: xing gao

Date: 26 July,2021

Abstract

This research aims to develop a target detection and tracking system that can realize real-time video surveillance. The purpose of the research is to realize a monitoring application that can run automatically and intelligently to detect and track illegally parked vehicles. Since the application scenario of the algorithm is a real traffic environment, it must be able to adapt to complex environmental interference, such as drastic changes in lighting conditions, frequent occlusion, and long-term stable tracking.

The thesis shows the detailed design process and test results of the system. This algorithm combines the target detection function based on deep learning network and the multi-object tracking algorithm based on key point matching. The method shown in the thesis focuses on detecting and tracking stationary vehicles in the no parking area. An object detection algorithm based on a deep learning network is used to recognize vehicles. Once the recognized vehicle is defined as an illegally parked vehicle through the determination of its motion state and location, an algorithm based on key-point matching is developed and tracked for this type of vehicle. If the target is still stationary in the no parking area after a period, the system will generate an alarm.

The method was tested in more than 20 hours of video. The video comes from public database and our own. They all show real surveillance scenes, including different time periods of the day and different locations. The test results show that the method achieves 100% in precision (also called positive predictive value), 95% in recall (also known as sensitivity) and 97% in F1 (a measure that combines precision and recall). The results

obtained also produce better detection and tracking compared to other comparable methods.

Publications

Some ideas and figures in this thesis have appeared in the following publications:

Conference Presentation:

Xing Gao, Philip M. Birch, Rupert C. Young, Chris R. Chatwin. "Occluded illegally parked vehicle detection and long term tracking (Conference Presentation)." Pattern Recognition and Tracking XXXI. Vol. 11400. International Society for Optics and Photonics, 2020.

Conference Paper:

X. Gao, P. M. Birch, R. C. D. Young and C. R. Chatwin, "Illegally parked vehicle detection using deep learning and key-point tracking," 9th International Conference on Imaging for Crime Detection and Prevention (ICDP-2019), 2019, pp. 7-12, doi: 10.1049/cp.2019.1160.

List of Acronyms

ASFP Actively selected feature point

AVMD Advanced video motion detection

BF Brute-force

BRIEF Binary Robust Independent Elementary Features

CNN Convolutional Neural Network

CRF conditional random field

DOG difference of Gaussians

DOH Determinant of Hessian

FCN Fully convolution all network

FPGA Field Programmable Gate Array

GMM Gaussian mixture model

HOG Histogram of oriented gradients

I-LIDS Imagery Library for Intelligent Detection Systems

IOU Intersection over union

KNN. K-nearest neighbours-based Background/Foreground Segmentation Algorithm

LOG Laplacian of Gaussian

LSTM Long Short Term Memory

MCMC Markov chain Monte Carlo

MOG Gaussian Mixture-based Background/Foreground Segmentation Algorithm

MOT Multi-object tracking

ORB Oriented FAST and rotated BRIEF

PCA Principal component analysis

PTZ Pan tilt zoom

R-CNN Region-Based Convolutional Neural Network

RNN Recurrent neural network

ROI region of interest

SHI Segmentation History Image

SIFT Scale-invariant feature transform

SORT Simple Online and Realtime Tracking

SSD. Single Shot MultiBox Detector

SURF speeded up robust features

SVM support vector machine

SVT Support Vector Tracking

VISOR Video Surveillance Online Repository

VMD Video motion detection

VTD visual tracking decomposition

YOLO You only look once

Acknowledgement

First of all, I would like to thank my supervisors Dr Philip Birch and Dr Rupert Young for their dedicated guidance in academics, strict requirements in work, and the meticulous care in life. They have provided me with a good and relaxed learning and working environment. They use extensive professional knowledge, rigorous academic attitude and selfless work sprit to lead me in all aspects. Here, I would like to extend my highest respect and heartfelt thanks to my supervisors.

I would also like to thank my parents and grandparents for your unconditionally support along the way to any decision I made, and to give me all the love. I would truly like to thank my girlfriend YingYing Zhang. We felt in love during my study. Her meticulous care and company made me feel at ease with my studies. At the same time, I would also like to sincerely thank my friends Zhang Haoran, Xi Runqi and Ethelbet Chinedu Eze for their encouragement and help.

Thanks to everyone around me for the encouragement, support, company and concern.

Content

List of Figures

List of Tables

Chapter 1

1.1 Introduction

In recent years, with the development of cities, the public's awareness of public safety and security has gradually increased. At the same time, the development of information technology, cloud computing and other technologies not only puts higher requirements on security products, but also shows a broad space for the application of video surveillance security products. More and more new technologies are gradually applied to urban construction and urban management, such as urban traffic management and urban emergency handling. This requires a video surveillance system that can image and truly reflect the characteristics of the monitored object. In particular, digital surveillance systems rely on networks and take intelligent and practical image processing methods as the core, occupying the commanding heights of today's surveillance technology (see Figure 1 for video surveillance system). At the same time, the rise of artificial intelligence can help systems detect, track, and analyse with minimal human intervention. However, there are still many shortcomings in fully automatic surveillance systems. Target detection, tracking, and analysis are influenced by multiple factors. Such as significant changes in weather and heavy occlusion by vehicles. And for long-term stationary vehicles under these influences, stable and effective tracking is also a huge challenge. Considering the complexity of the monitoring scenario, this thesis is devoted to developing a tracking system for cars. This system can be used to monitor, track and analyse the long-term parking status of vehicles under complex conditions.

1.1.1 Video Surveillance Systems

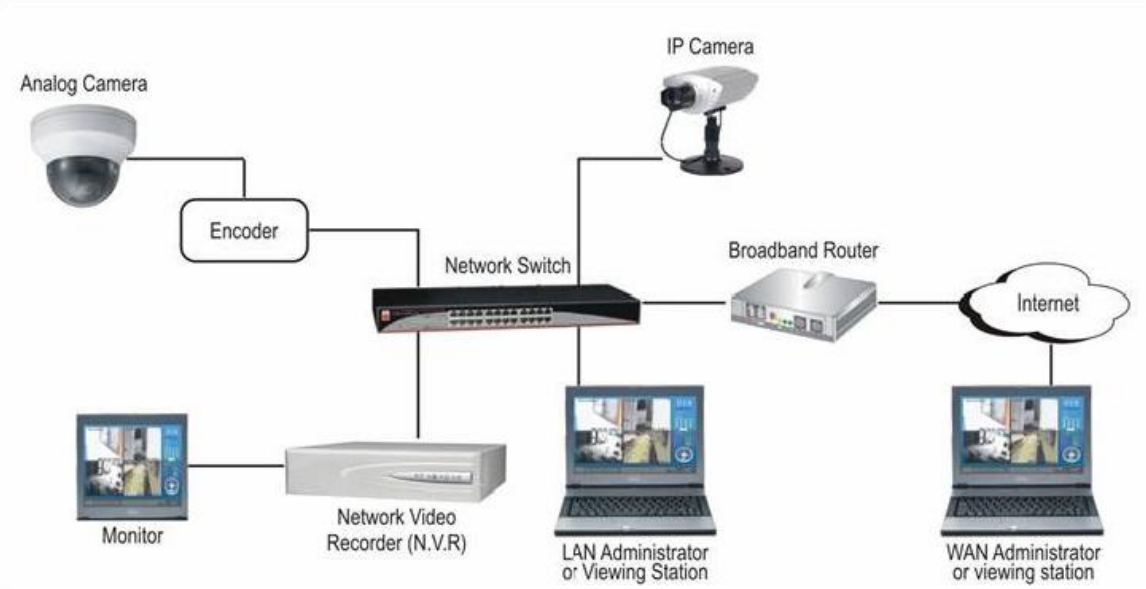

Figure 1 Video surveillance system which includes wired and wireless transmission methods.

Video surveillance is the physical basis for real-time monitoring of key departments or important places in various industries. The management department can obtain effective data, image or sound information through it, and timely monitor the process of sudden abnormal events, to provide efficient and timely command and dispatch, deploy police force, and handle cases.

The surveillance system is composed of five major parts: camera, transmission, control, display and recording. The camera transmits the video image to the control host through the coaxial cable, network cable and optical fibre. In addition, with the development of communication technology, wireless network technologies such as 5G are also used for data exchange between the camera and the host. The control host then distributes the video

signal to each monitor and video equipment, and simultaneously records the voice signal that needs to be transmitted into the video recorder. Through the control host, the operator can issue instructions to control the up, down, left, and right movements of the Pan-Tilt-Zoom (PTZ) camera and also switch between multiple cameras. The special video processing mode can be used to record, replay, and process images to make video surveillance system has a broader application scenario.

However, the biggest disadvantage of traditional video surveillance is that it relies on the 24-hour conscientious monitoring by the staff on duty. A slight negligence may miss important situations, which delays the alert and leads to incalculable consequences. As shown in Figure 2, the number of cameras is usually greater than the number of monitors. The use of round robin displays and multi-screen with small images is likely to cause security to miss abnormal phenomena.

Figure 2 Video surveillance platform.

Nowadays, with the rapid development and promotion of computer applications, a powerful wave of digitisation has been set off all over the world, and the digitisation of various devices has become the primary goal of security protection. The key performance characteristics of the digital monitoring alarm are: real-time display of the monitoring screen, video image quality adjustment function, recording speed can be set, fast retrieval, multiple recording methods, automatic backup, pan-tilt-lens control function, network transmission, etc. Some of these characteristics are also crucial to the study of this thesis.

1.1.2 The Evolution of Video Analytics

According to the theory of attention economics, most of today's security control centres and corresponding video surveillance systems present a large amount of information to security personnel, resulting in poor attention span. A study [1] has shown that the performance of operators shows a disturbing trend:

1. The performance of security operators dropped significantly after 20 minutes
2. Poor image quality accelerates this rate of decline
3. Doubling the number of cameras viewed will double the speed of descent

To improve the work efficiency of security personnel, video analysis technology is used in the security field. Video analysis technology uses computer image visual analysis technology to analyse and track the target in the camera scene by separating the background and the target in the scene. Security personnel can pre-set different illegal rules (for example, the

area is a no parking area at a certain time) in different camera scenes. Once the target violates the predefined rules in the scene, the system will automatically send out an alarm message, and the monitoring command platform will automatically pop up an alarm message and emit a warning sound. Through related equipment, users can click on the alarm information to reorganise the alarm scene and take relevant preventive measures.

According to the introduction of the white paper [2] released by the Avigilon Corporation, the evolution of video analytics has gone through the following three technologies:

1. Video motion detection: VMD is the standard function of most new surveillance cameras, video recorders and video management software. VMD focuses on detecting any pixel movement between scenes. VMD is the most effective method in a static environment, but it has a limited effect in a dynamic environment and has a high false alarm rate. For example, Axis Communications [3] has developed a VMD-based motion detection application that can be used for low-traffic scenarios.
2. Advanced video motion detection: In order to solve the limitations of VMD, the industry has developed from VMD to advanced video motion detection (AVMD). AVMD is based on background modelling and alerts for any changes that deviate from the established background model. This technology focuses on monitoring the scene and uses data collected through complex manual calibration to identify moving objects. After being properly set and calibrated, AVMD is quite efficient. For example, Huang [4] proposed a motion detection algorithm that integrates a background modelling module, an alarm trigger module, and an object extraction module to achieve 81.84% of the average F1 on the test video. However, when the background composition changes (such as

environmental, seasonal, and physical changes, etc.), its function will be limited, causing the false alarm rate to increase over time and requiring regular recalibration.

3. Advanced video pattern detection: The latest evolution of video analysis is advanced video pattern detection, which is based on a pattern modelling algorithm to alert any changes in the mode of known object types such as people or vehicles. This technology focuses on identifying objects in the field of view and using the object's motion information to accurately classify them. For example, Truong et al. [5] used Gaussian mixture model and a fuzzy c-means (FCM) algorithm to segment candidate fire regions, and then a support vector machine (SVM) was used to distinguish between fire and non-fire.

The characteristics of the above three types of video analysis technologies are highlighting relevant information of interest to help operators maintain their concentration. The application of these video analysis methods has completely changed the previous mode of monitoring and analysing surveillance pictures by security personnel. The intelligent video surveillance system can continuously analyse the monitored pictures, immediately notify the occurrence of abnormalities, reduce reliance on personnel, and improve work efficiency.

In addition, users can define multiple types of intrusions according to actual needs, allowing users to define the characteristics of threats more accurately, effectively reducing false positives and false negatives alarms, and reducing the amount of useless data. Finally, traditional video surveillance can only be used as evidence for subsequent queries, while video analysis methods can play an early warning role (for example, someone left suspicious objects in a public place, or someone stays in a sensitive area for a long time) for prompting

security personnel to pay attention to relevant surveillance pictures before security threats occur, effectively preventing accidents.

1.1.3 Video Surveillance Application Field

In the past, video surveillance technology applications were mainly concentrated in finance, public security, transportation and electric power departments. For example, according to a survey by IDC (International Data Corporation) [6], the government is the largest video surveillance industry in China, accounting for 47.6% of total expenditures, and the Safe City (The project proposed by Ministry of Public Security of the People's Republic of China) project is the main market driver force. The goal of "Safe City" construction is to build a comprehensive urban early warning system and emergency command system. Such a comprehensive urban police management system that integrates comprehensive social security management, urban traffic management, and fire dispatching can not only meet the needs of public security management, urban management, traffic management, emergency command, etc., but also take into account disaster early warning, safety production monitoring and other aspects.

In the UK, video surveillance systems have played a key role in public security and fighting crime. With more than 6 million cameras all over the UK BSIA (The British Security Industry Association) [7], this surveillance system provides a deterrent to potential criminals and makes people feel safer. The British Department of Transport estimates that speed cameras

can reduce personal injury accidents by 22% and reduce the number of people killed or seriously injured at the camera scene by 42% [8].

In addition, in the field of transportation, the installation and implementation of intelligent traffic monitoring systems are essential for management departments. As motor vehicles are indispensable transportation for people to travel, effective management, punishment, and reduction of traffic violations of motor vehicles, rapid detection of traffic accident escapes and motor vehicle thefts have become more important for local governments and traffic control agencies. The transportation department can transmit the scene image of the surveillance area back to the command centre through the monitoring system, so that the management personnel can directly grasp the traffic conditions such as vehicle queues, jams, and signal lights. Video surveillance systems are also used in crowded places such as airports, stations, and shopping malls. High-definition intelligent monitoring adopts intelligent analysis technology to analyse the video content. By pre-setting different alarm rules in different camera scenes, it automatically alarms abnormal behaviours, and can generate various statistical information based on massive data to perform intelligent monitoring.

Although some systems have the defects shown in section 1.1.1, which is because the scenarios are usually in a complex environment and have huge data. These effects are serious challenges to the stability of the system. Hence, with the development of video analysis technology, a series of new technologies represented by deep learning, big data analysis and image processing have helped video surveillance systems be applied in multiple markets.

For example, in the process of early warning of wildfires, video surveillance has played a key role [9]. The front end of the wildfire monitoring system can be set in the wilderness without personnel on duty. With video analysis technology, the system can monitor and discover hidden fires 24 hours a day. In recent years, due to extreme heat and dryness, wildfires in 2020 have worsened by 13% compared to 2019 [10], which further emphasizes the urgency of video analysis in forest monitoring.

The supermarket retail industry is also an emerging market for video surveillance. Alibaba opened its first automated retail supermarkets in 2017. Since 2017, these supermarkets have been named Tmall supermarkets. Alibaba uses several technologies to automate Tmall supermarket. First, through image recognition technology, Tmall Supermarket will conduct rapid facial feature recognition and identity verification on consumers. Secondly, through item identification and tracking technology, combined with consumer behaviour identification, Tmall Supermarket can determine the consumer's settlement intention, and finally use the smart gate to quickly complete the payment. These intelligent analysis technologies are integrated into the video surveillance system to monitor shoplifters and payments. Even for traditional retail stores, smart video surveillance systems are necessary to reduce costs.

The fact that video surveillance technology is only used in the enterprise industry has gradually been broken, and the home has become a new market for video surveillance applications. In the home security market, video surveillance is mainly used for residential security and property monitoring. Home monitoring system can use network technology to connect the video, audio, alarm, and other monitoring systems installed in the home, and

save and send useful information to other data terminals, such as mobile phones, notebooks, 999 alarm centres, etc. AT&T announced a service called digital life in 2016 [11]. The video surveillance system and network infrastructure are linked together, which is now used by AT&T to prevent burglary and remote furniture control.

1.1.4 Parked Vehicle Tracking and Challenges

With the development of the world economy, the rapid increase in the number of motor vehicles has led to a series of problems such as traffic jams, among which illegal parking of motor vehicles is an important factor in causing traffic accidents and traffic jams. Illegal parking of motor vehicles has become a ubiquitous urban phenomenon. According to research by Aljoufie [12], illegal parking practice can hamper sustainability of transportation system. Illegal parking management is one of the important evaluation criteria for urban traffic management. At present, most illegal parking detections use manual monitoring to collect information at fixed points. When illegal parking is discovered through video surveillance, law enforcement officers manually control the pan-tilt camera to zoom in, take pictures of the vehicle, and manually identify and record the license plate, and manually restore the camera's pre-set position after completion.

For example, in 2007, California's Mountain Recreation and Conservation Administration (MRCA) installed the first stop sign camera in the United States [13]. The five cameras ware located in state parks such as Franklin Canyon Park and Temescal Gateway Park.

However, the detection process for illegally parked vehicles is complicated and cumbersome, the labour cost is high, and the management is complicated and inefficient. With a large increase in monitoring points, the workload of the monitoring personnel is getting heavier, and the accuracy of the work is greatly reduced, which greatly consumes the human, material, and financial resources related to traffic control.

Figure 3 Illegally parked vehicle monitoring [14].

Figure 4 Illegally parked vehicle monitoring [15].

If the manual monitoring mode is adopted, since the monitoring centre usually has multiple cameras and corresponding displays, some short-term violations may not be noticed by the operator [15]. The mode of on-site capture requires a large number of personnel [14]. If it is not discovered and investigated in time, it will cause derivative problems such as traffic jams. The above two methods will waste a lot of time and energy of traffic managers. Therefore, an automated parking management system is proposed to deal with the complex and changeable traffic environment. First, in spaces where parking is prohibited, such as bus stops, fire hydrants, sidewalks, and any marked areas, the system needs to automatically detect illegal vehicles and generate alarms. Secondly, track the captured target.

1.1.5 Research Gap

This thesis contains a literature review of current automated systems. Chapter 2 shows the technical background of various target tracking and illegally parking monitoring systems. The review provides evidence to prove that the monitoring system can be used extensively in the field of illegally parking monitoring. However, the current automatic monitoring system still cannot completely replace the manual monitoring mode because the related technology is still not reliable enough. Although research and technology are advancing, most of the illegally parking monitoring systems have not been optimized to achieve the following functions that are of interest in this research:

1. Accurate target vehicle detection capability. When multiple targets appear at the same time, the system should accurately distinguish them.
2. Stable target vehicle tracking ability. The system should be more robust to occlusion, changes in light, changes in vehicle appearance and long-term stationary.

The former means that the system needs to accurately detect the target vehicle from a congested road. However, many current detection methods are based on background modelling to obtain foreground information. This method is difficult to control the detection results without excessive detection or missing detection, which often causes misjudgements. In addition, multi-object tracking requires a digital ID to be assigned to each target. The latter means that the system needs to be able to adapt to complex environmental changes and provide effective long-term tracking, because sometimes the transportation department will delimit areas that prohibit long-term parking but allow

temporary parking. Only continuous tracking can set different time thresholds according to different requirements to realize intelligent monitoring of illegally parked vehicles.

1.1.6 Research Context and Scope

The content of this thesis is to propose an automatic monitoring algorithm to help road cameras automatically track illegally parked vehicles. This research does not involve other uses of vehicle monitoring, such as non-stationary target tracking, accident reporting and red-light enforcement. It should be noted that this thesis only discusses tracking solutions based on static cameras.

1.2 Thesis Plan

This thesis is dedicated to proposing a reliable automatic monitoring system for illegally parked vehicles, which can be used in various scenes and weather conditions in the city to relieve the monitoring pressure of traffic management departments. The existing detecting and tracking algorithms are discussed in Chapter 2. Chapter 2 also analyses the challenges and shortcomings of existing methods. The project started with research on a tracking algorithm that can automatically monitor and alarm, and then gradually developed to be suitable for more complex and strict environments.

This thesis is divided into two parts. The first part shows the combination of deep learning-based target detection and multi-object tracking to track illegally parked vehicles, while the second part is the improvement and development of the first part based on key point matching. The entire technology is developed and tested on a personal PC platform with an Intel Xeon(R) CPU E5-1620 V3 running at 3.5GHz, NVIDIA Quadro K4200 and with 15.6 GB RAM. The tracking system developed by this project is developed by python and runs on the Spyder platform of the Linux system. The image processing function in the system is provided by the OpenCV-Python library, and the detection weight of the deep learning algorithm YOLOV3 is provided by [103] and pre-trained. All functions and errors during the running of the software can be visually detected and improved.

Figure 5 depicts the overall framework of the system and shows the flow from video capture to the end of tracking. It shows that the target will be detected first, then the state will be judged (speed and location), and finally the eligible target will be tracked until it leaves the no parking zone. Specific system design issues, operation and configuration strategies are introduced in Chapters 3, 4, 5 and 6.

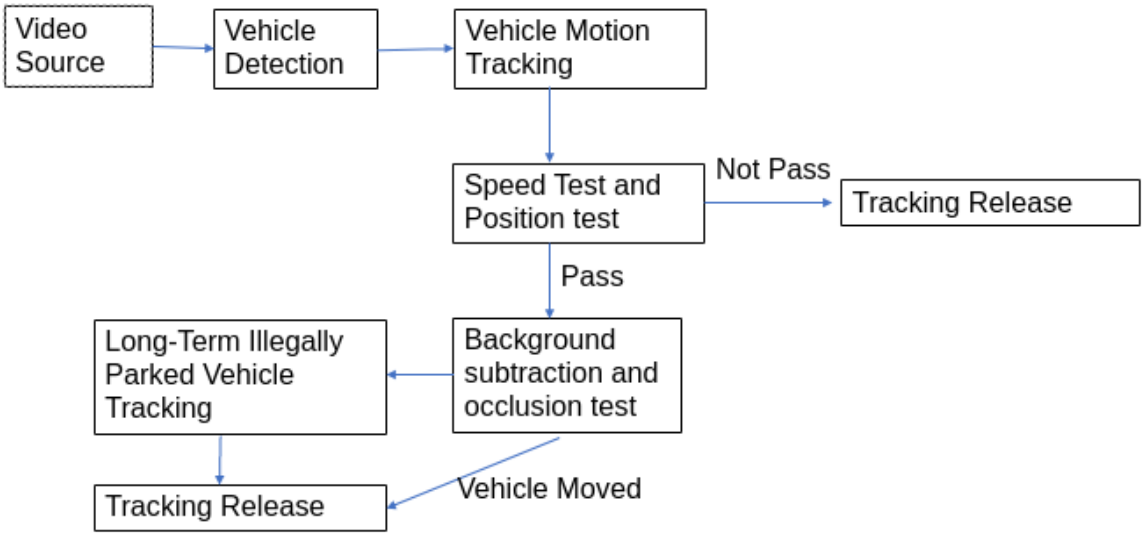

Figure 5 The proposed system framework. The proposed system has been divided into three part: vehicle detection, illegally parked vehicle judgment and target tracking. Only the illegally parked vehicles have been tracked.

The initial illegally parked vehicle tracking was developed in Chapter 3. The deep learning algorithm YOLO (You Only Look Once) [16] was used to detect vehicles in the scene, and the multi-object tracking algorithm SORT (Simple Online and Real-time Tracking) [17] was used to track stationary vehicles in the scene. The technology had been tested on i-LIDS dataset [18] and ViSOR dataset [19]. It can be found that although this method achieves a good precision performance, the algorithm generated many false alarms during the test, mainly due to the defect of the YOLO [16] algorithm itself in the detection of nearby small objects. Frequent occlusion and light changes were also causing of failure. Another problem is that the identity of the target is frequently changed due to the instability of the tracking component. In order to solve the above problems, the remaining chapters introduce more powerful solutions, in which feature point matching is used to track illegal objects in various complex environments.

Chapter 4 introduces an algorithm to keep the tracking from being dismissed. The feature point description of the vehicle was used to match, but it was still restricted by occlusion and environmental conditions. Due to the recognition errors of the deep learning algorithm in the target detection, some non-vehicle targets will be recognized as vehicles, which leads to the invalid tracking of the system. In addition, because the number of SIFT (Scale-Invariant Feature Transform) [20] feature points is too sparse on some blurred targets, the accuracy and reliability of matching are insufficient, which has a negative impact on tracking. This method had been tested on the i-LIDS dataset [18] and the results were significantly improved compared with the method in Chapter 3. In addition, unlike the multi-object tracking algorithm simple online and real-time tracking (SORT) [17], this method can keep the ID of the target until the target leaves the no parking zone instead of judging to base on the geometric prediction of the bounding box's position, which causes a higher error rate and ID switch.

In Chapter 5, the problem of the previous chapter is solved. A method based on Dense SIFT feature point extraction has been used, which significantly increases the number of feature points. At the same time, the increasing of feature points increased the robustness to blurred targets and light changes. Although the Dense SIFT based technology is better than the previous method which discussed in Chapter 4, it requires more computing power in calculation, and because the bounding box of the target contains the area outside the vehicle, in the process of feature point matching, feature descriptors which outside the vehicle will interfere the releasing of the tracker.

In order to overcome the aforementioned problems, a novel actively selected feature point (ASFP) method has been proposed in Chapter 6. Chapter 6 shows a complete parking tracking system, which includes the tracking process under strong light changes and long-term, high-frequency occlusion. This technology was tested in different scenarios and the results were compared with other methods. According to observations, the results of this method are more accurate.

Chapter 7 is the last chapter and conclusion of the thesis, including the advantages of our method, the overall direction of this research and the challenges it faces. Also, in the last chapter, a discussion of the future work and envisaged direction of the illegally parked vehicle monitoring system have been given.

Chapter 2 Literature Review

2.1 Introduction

This chapter reviews and analyses the current existing target detection and tracking methods and illegal vehicle monitoring methods. The essence of illegally parked vehicle tracking is to track stationary objects in a specific area, while stationary object tracking is generally based on object tracking algorithms. As one of the core topics in the field of vision research, the visual tracking of objects has a research history of nearly 20 years [21]. Visual tracking refers to the detection, extraction, recognition and tracking of the target in the video image sequence to obtain the target's motion parameters, such as target centroid position, speed, acceleration, and motion trajectory, etc., thereby carrying out further processing and analysis to realize the understanding of the target's behaviour to complete higher-level tasks. The visual tracking process generally includes several stages such as target detection, target feature extraction, and target tracking. Among them, target detection and feature extraction require prior knowledge, and different methods are selected according to different occasions. Target tracking can be understood as estimating the spatiotemporal state of the target based on the initial state of the target and the target's visual features obtained through feature extraction.

2.2 Chapter Organization

The arrangement of this chapter as follows. Since this research proposed a feature matching based tracking method, the image features detection and different feature descriptors are reviewed in Section 2.3. In Section 2.4, an overview of object tracking is given, which is divided into three parts: discriminative, generative and deep learning based object tracking method. Section 2.5 reviews the object detection algorithm based on deep learning, this is because the proposed method involved deep learning detection algorithms. Section 2.6 reviews some current illegally parked vehicle tracking methods and Section 2.7 reviews some semantic segmentation methods and background subtraction. A summary is given in Section 2.8.

2.3 The Feature Detection and Feature Descriptors

2.3.1 Feature Detection

Image feature detection is the premise of image analysis and image recognition, and it is the most effective way to express high-dimensional image data. The local feature point is the local expression of the image feature, it can only reflect the local characteristics of the image, so it is more suitable for image matching and retrieval applications, but not suitable for image understanding. The global features represent some global information, such as colour distribution, texture features, and the shape of main objects. Global features are susceptible to environmental interference. Unfavourable factors such as illumination,

rotation, and noise will affect the global features. In contrast, the local feature points often correspond to the structure where some lines cross or the light and dark changes in the image, and they receive less interference.

The blob and corner are two types of local feature points [22]. A blob usually refers to an area that is different in colour and grey from the surrounding area. It has stronger anti-noise ability and better stability than corner feature because the blob feature reflects the characteristics of the area. The corner feature comes from the corner of the object or the intersection between lines in the image.

Therefore, in this section, two commonly used methods of blob detection are reviewed, namely Laplacian of Gaussian (LoG) and Determinant of Hessian (DoH). LoG was proposed by Lindeberg in [23] and used to detect local extremum points in an image. This method used a Gaussian low-pass filter to convolve the image. The goal is to remove the noise in the image, and then use the LoG operator to convolve the image. When the blob structure in the image is close to the shape of the LoG operator, the response of the image convolution reaches the maximum.

Another classic blob detection is based on Determinant of Hessian (DoH). The value of the determinant of the Hessian matrix also reflects the local structural information of the image. Compared with LoG, DoH has a better inhibitory effect on the slender structure blobs [121].

2.3.2 Feature Descriptors

Once the distinguishing feature points are detected from the original image, various local feature descriptors can be used to describe the feature points, and finally compared with the object to be detected to obtain a point-to-point match. Choosing appropriate feature descriptors plays an important role in object detection. This is because the appearance of the object may change due to various factors, including viewing angle, occlusion and light. These disturbances have inspired researchers to develop different image feature description methods. This section reviews two representative feature descriptors and their applications.

In 1999, Lowe proposed Scale-invariant Feature Transform (SIFT) [20], which extracted feature points with invariable scale on the difference of Gaussian (DoG) and perfected the algorithm in 2004. The algorithm has certain affine invariance and perspective invariance, rotation invariance and illumination invariance, so it has been widely used in image feature description.

The algorithm can be summarised into three steps:

1. the construction of the Difference of Gaussian pyramid
2. the detection of feature points
3. the feature description

In the first step, the algorithm constructs a pyramid with a linear relationship so that the feature points can be searched on a continuous Gaussian kernel scale. It is better than LoG

because it uses the first-order difference of gaussian to approximate the LoG operator, which greatly reduces the amount of calculation.

In step of feature point search, the main purpose is the selection of extremum points. The extremum point is selected by comparing the detection point with the nearest-neighbour points. In addition, another key point in the second step is to delete the edge point, because the value of DoG will be affected by the edge.

The last step is the description of the feature points. The description of the direction of feature points requires histogram statistics on the gradient directions of points in the nearest-neighbour of the feature points, and the direction with the largest proportion in the histogram is selected as the main direction of the feature points. When calculating the feature vector, it is necessary to rotate the local image in the main direction, and then perform the gradient histogram statistics in the nearest-neighbour.

As a powerful algorithm that includes feature detection and feature description, SIFT algorithm is widely used in many fields. Piccinini et al. [24] proposed a method to detect and locate objects under extreme occlusion conditions. This method uses SIFT key point extraction and mean shift clustering to partition the correspondences between the object model and the image onto different potential object instances with real-time performance. For many targets that cannot be detected by appearance, this method has good results. Hashmi et al. [25] proposed a method that can detect copy-move forgery in images. In copy-move forgery, a part of the image is copied and pasted in another part of the same image to conceal an object or to duplicate certain image elements. In this method, the SIFT

algorithm has been used to describe the feature points of the potential area to help find the location of the forgery.

In 2006, Bay et al. [26] proposed the Speeded Up Robust Features (SURF). Aiming at the shortcomings of the SIFT algorithm that the speed is too slow and the amount of calculation is large, this method uses the responses of Haar wavelets to approximate gradient computation and then accelerate the SIFT operator.

Like SIFT, the SURF algorithm also includes the detection and description of features. SURF descriptors are widely used in the field of object detection. Pranata et al. [27] used deep learning networks and SURF to implement a computer-assisted method for detecting fracture locations in Computed Tomography (CT) images. The SURF algorithm has been used to extract the feature points in the CT image and match them with the reference image to locate the fracture. Zhao et al. [28] proposed a field-programmable gate array (FPGA) based traffic sign detection system. They used the parallelism and rich resources of FPGAs to implement a real-time (60 frames per second) detection method by SURF matching.

In summary, the SIFT and SURF algorithms are based on the feature detection method not only to achieve feature detection, but also to describe the detected feature points. Their feature descriptors are all invariant to rotation, scale, blur, Illumination, warping and noise area [29].

In addition to SIFT and SURF, the traditional feature descriptors also include Binary Robust Independent Elementary Features (BRIEF) [30] and Oriented FAST and rotated BRIEF (ORB) [31] descriptors. And deep learning has also been introduced into the field of feature

description, and many feature descriptors based on deep learning have been proposed, such as DeepCD [32], L2-Net [33]and HardNet [34]. These descriptors have their own characteristics and advantages.

According to the research from [35], it can be found that compared with traditional feature descriptors, deep learning-based descriptors usually have better performance, but with higher computation consumption and longer matching time. And because the descriptors based on deep learning are built on convolution neural networks (CNN), it is difficult to achieve lightweight operation among the traditional feature descriptors, SIFT descriptor has achieved very good performance and have good robustness to conditions such as illumination and occlusion [35],[36], although SIFT is not the fastest descriptor.

2.4 Tracking Methods

The above feature descriptors can be used for target tracking, however, the feature descriptor does not directly participate in the generation of the tracking result. For algorithms that directly perform target tracking without feature matching, the appearance of a changing target is always a challenge. Usually, the appearance changing of the target includes external changes and internal changes. Internal changes include changes in the shape or angle of the target. External changes are usually caused by changes in light, environment, occlusion and camera movement. These can only be handled by adaptive updating its representation gradually. Therefore, research on observation model and model update becomes very necessary. Before the 2010s, researchers still focused on traditional

tracking methods for visual targets [37], such as optical flow method, Kalman filter, particle filter, and Mean-Shift.

Lucas et al. [38] proposed the optical flow algorithm in 1981. Optical flow refers to the instantaneous velocity of the pixel motion of a moving object on the observation imaging plane. The principle of this algorithm is to use the changes in the time domain of pixels in the image sequence and the correlation between adjacent frames to find the correspondence between the previous frame and the current frame, thereby calculating the object's motion information between adjacent frames. Denman et al. [39] improved a tracking method based on optical flow. This method overcomes the shortcomings of detecting background flow and develops a system that can use tracking output to enhance optical flow detection.

Kalman and Bucy [40] published a paper about Kalman filtering in the 1960s and proposed an algorithm that can estimate the state of a dynamic system through a series of incomplete and noisy measurement values. The algorithm only needs to use the measured value of the current moment and the estimated value of the previous moment, and the estimated value of the current state can be derived from the model. This algorithm introduces state variables into the filtering theory, which can solve the filtering problem of time-varying, multi-variable, and non-stationary time series. This algorithm is easy to implement on a computer because it is a recursive algorithm, and does not need to store historical measurement values. The amount of calculation and storage is relatively small, and it is convenient for real-time online processing. With the rapid development of computer technology, the Kalman filter has gradually obtained extensive research and applications.

For example, Patel et al. [41] developed a method for tracking objects using Kalman filter. The method draws the trajectory of the target by predicting the position of the centre of mass of the target. Bewley et al. developed SORT [17] which uses Kalman filter algorithm to predict the position of the detected object in the next frame. The prediction is matched with the detection result from next frame to achieve vehicle tracking.

Particle filtering is also used for object tracking. The essence of particle filtering is monte-carlo simulation. It is an estimation of objects' location probability in the region of interest. By using this feature, Yang et al. proposed a multi-object tracking algorithm [42]. This method uses colour and edge direction histograms to characterise the tracked object. Another method based on particle filtering has been used in [43] to improve tracking performance. An unscented particle filter is used to generate sophisticated proposal distributions that seamlessly integrate the current observation. Cho et al. [44] deployed a particle filter-based moving object tracking system that can be used in FPGA. The system includes five modules: Camera Controller, Image Store Memory, Image Subtractor, Noise Reduction Filter and Particle Filters. The particle filter module uses the pixel differencing between the images of the previous and current frames generated by the Image Subtractor module to output the tracking image.

Mean shift is an important and classic algorithm in the field of visual tracking. It has been widely used in clustering, image smoothing, image segmentation and tracking. An enhanced mean shift tracking algorithm using joint spatial-colour feature was proposed by Hu et al. [45]. The algorithm uses kernel density estimation to model the target image and uses the estimated kernel density to develop a new similarity measure function. Through these two

similarity measurement functions, two similarity-based mean-shift tracking algorithms were developed. In addition, in order to solve the object deformation problem, the variance matrix is calculated to update the direction of the tracked object.

Another SIFT-based mean shift algorithm [46] is also used for object tracking in real scenes. The mean shift is used to perform a similarity search through the colour histogram and the SIFT feature is used to deal with cross-frame regions. The mean shift algorithm combined with colour distribution has been used in Comaniciu et al. [47]. Research has found that the solution is suitable for tracking various objects with different colour or texture patterns.

Since the 2010s, more and more people have paid attention to the use of machine learning methods in tracking. In recent years, the results in the field of target tracking have basically been using machine learning methods. At present, online learning tracking algorithms can be divided into two categories: generative model and discriminative model. In addition, tracking algorithms based on deep learning are currently very popular research directions.

2.4.1 Generative Tracking Methods

The target tracking method based on generative models can be defined as: The first step is extracting the target features to learn the appearance model to represent the target, and then searching the image area for pattern matching, finally find the area in the image that best matches the pattern to locate the target.

The target information carried by the generative model is richer, and it is easier to meet the evaluation criteria of target tracking and the real-time requirements when processing a large amount of data information. It is the first step in applications such as intelligent video surveillance, human-computer interaction, and intelligent transportation. In recent years, it has been widely used in emerging fields such as military guidance, medical diagnosis, meteorological analysis, and astronomical detection.

The core of the generative method is the target representation method and target model. For example, probabilistic principal component analysis (PCA) is very effective for target tracking, because its representational power can capture the generation process for high-dimensional image data. Ross et al. [48] proposed an incremental algorithm model based on PCA, which can effectively online adapt the changing of appearance of the target. This algorithm avoids the failure of many algorithms that use the fixed appearance model of the target. A new online object tracking algorithm with sparse prototypes was proposed by Wang et al. [49]. The classic principal component analysis (PCA) algorithm and the latest sparse representation scheme are used to learn effective appearance models. Taking advantage of the subspace representation, the algorithm can handle higher resolution image observation.

The following methods are also belonging to generative tracking. Kwon et al. [50] proposed a generative tracker named visual tracker sampler. The sample used in this method includes tracker and target state, and tracking is achieved by selecting a most likely tracker and a highly possible state. Experiments show that their method can accurately and robustly track targets in a real-world tracking environment. Lee et al. [51] proposed a tracking algorithm

based on visual tracking decomposition (VTD). The algorithm decomposes the observation model into multiple basic observation models which construct by sparse principal component analysis (SPCA). Multiple basic trackers designed by associating multiple basic observation models, and then integrate multiple basic trackers into one compound tracker through the Markov Chain Monte Carlo (IMCMC) framework.

Regardless of whether the generative model uses global features or local features, its essence is to find the closest candidate target to the target model in the high-dimensional space of target representation. The disadvantage of this type of method is that it only focuses on the target information and ignores the background information.

2.4.2 Discriminative Tracking Methods

The discriminative method turns the tracking problem into a classification problem and trains the classifier to distinguish the target and the background. In the current frame, the target area is the positive sample, and the background area is the negative sample. A machine learning method is used to train a classifier online to distinguish the target and the background. And this trained classifier will be used to find the optimal region in the next frame.

The discriminative tracking method was first proposed by Collins and Liu [52]. This method is also called tracking-by-detection. This method adaptively selects the most distinguishing

features for the current background and target, and then the feature evaluation mechanism is embedded in the mean shift tracking system.

[53]-[60] shows the applications of support vector machine (SVM) in the tracking field. A SVM is a learning technology developed by Boser and his team [53] that is a popular and powerful classifier. In the context of target tracking, it can be converted into a binary classification problem of target and background. The feature information of target and background is used as positive and negative samples to train the SVM to obtain the overall classifier. Then the overall classifier can be used to distinguish the target and background in the next frame.

Osuna et al. [54] developed a SVM for face detection and demonstrated the feasibility of their method on the face detection problem involving a dataset of 50,000 samples. By using the powerful background segmentation capability of SVM, Avidan [55] proposed a vehicle tracking method called Support Vector Tracking (SVT) that combines optical flow and SVM. SVT combines the computational efficiency of optical flow-based tracking with the function of the general classifier SVM, thereby extending the functions of the tracker and the classifier.

However, a SVM cannot effectively solve the online classification problem, because when a new sample is added, the classifier must be fully retrained. In response to this situation, Tian et al. [56] proposed a tracking algorithm based on the linear SVM classifier. The algorithm uses the key frame of the target to online update the SVM classifier and uses historical information to effectively process the target appearance variation. Another SVM

based online tracking method was used in [57] which combined with PCA. Unlike the conventional SVM method, their method directly learns and predicts the state of the object, instead of tracking the two-dimensional transition in the process. Sharma et al. [58] proposed another online SVM based tracking algorithm. The SVM parameter vector is online learned to construct a discriminative classifier. Then the learned SVM parameter are used for object likelihood model construction.

In [59], the ranking SVM was trained for tracking. The tracking process has been formulated as a weakly supervised sorting problem to incorporate information into object in the next frame. Ning et al. [60] deployed dual linear structured support vector machine (SSVM) to achieve fast learning and execution during tracking. This method formulates object tracking as a linear SSVM detection problem to achieve rapid model update in its original form.

SVMs are very powerful models, which perform well in the various tracking algorithms shown above. However, if the amount of data is very large, they may face challenges in terms of runtime and memory usage.

2.4.3 Deep learning Tracking Methods

Recently, discriminative tracking methods have gradually occupied the mainstream position, and discriminative methods represented by Correlation Filter and Deep Learning have also achieved good results. However, because the method based on correlation filtering is not very adaptable to scale changes, the solution to scale changes will decrease the tracking

speed. At the same time, it is not robust to fast moving objects or low frame rate video. The model update strategy and update speed also affect the tracking of occluded objects. Considering the above reasons, the correlation filtering shows a low correlation with this project, only deep learning tracking is reviewed here.

The human eye can easily follow a specific target within a period of time. But for the machine, this task is not simple, especially in the tracking process. There are various complicated situations such as severe deformation of the target, occlusion by other targets, or interference from similar objects. Over the past few decades, the research on target tracking has made considerable progress, especially since various machine learning algorithms were introduced. Since 2013, deep learning methods have begun to develop rapidly in the field of target tracking and have gradually surpassed traditional methods in performance and achieved huge breakthroughs.

Since 2015, there has been a new trend in the application of deep learning in the field of target tracking. That is, directly use the CNN network trained on a large-scale classification database such as ImageNet [61] to obtain the feature representation of the target, and then use the observation model to classify and obtain the tracking result. This approach not only avoids the dilemma of insufficient samples for direct training a large-scale CNN during tracking, but also makes full use of the powerful representation capabilities of deep features.

For example, as a representative work of applying CNN features to object tracking, FCNT (Visual Tracking with Fully Convolutional Networks) [62] has done an in-depth analysis of the tracking performance of pre-trained CNN features on ImageNet and designed

subsequent network structure based on the analysis results. Based on the analysis of CNN features, the research constructs a feature screening network and a prediction network, which prevents the tracker from drifting while being more robust to the deformation of the target. Another concise and effective way to implement tracking using deep features was proposed by Ma et al. [63]. The main idea is to extract deep features, and then use correlation filters to determine the final bounding-box.

These methods are all successful cases of applying pre-trained CNN network to extract features to improve tracking performance, which shows that using this idea to solve the lack of training data and improve performance is highly feasible. However, the pre-trained CNN network for the classification task pays more attention to the objects between the classifications and ignores the differences within the classes. The classification task divides similar objects into one category, and the tracking task takes the different appearances of the same object as one category, which makes these two tasks very different.

Realising that there is a huge difference between image classification tasks and tracking, Nam et al. [64] proposed a network called MDNet to directly use tracking video to pre-train CNN to obtain the general representation. MDNet uses video data which closer to the essence of tracking for training and proposes a novel multi-domain training method. The idea of cross-application of training data solves the difficulty of distinguishing foreground and background in all training sequences with the same CNN, but the speed of this method is still slow because the data of the first frame is used to train the bounding box regression model.

Hence, recurrent neural networks (RNN), especially LSTMs (Long Short-Term Memory) with gate structure, GRUs (Gated Recurrent Units), etc. have shown outstanding performance on timing tasks. Many researchers have begun to explore how to apply RNNs to solve the problems in existing tracking tasks.

Cui et al. [65] proposed a novel tracking method using multi-directional recurrent neural network to model and identify reliable target parts which are useful for overall tracking. This method generates a confidence map of the entire candidate region through RNN training on a two-dimensional plane. The confidence map will be used to train correlation filters to obtain the final tracking results. Compared with other correlation filter algorithms based on traditional features, this algorithm has a greater improvement, which shows that RNN's exploration of association relationships and constraints on filters are indeed effective.

Different from the trend of deep learning in the field of vision such as detection and recognition that far surpass traditional algorithms, the application of deep learning in the field of target tracking still faces challenges. The main problem is the lack of training data: one of the capabilities of the deep model comes from the effective learning of a large number of labelled training data, and target tracking only provides the bounding-box of the first frame as training data. In this case, training a deep model for the current target at the beginning of tracking is difficult.

The above shows several current ideas to solve this problem based on deep learning tracking algorithms. However, the existing deep learning target tracking methods are still difficult to meet the real-time requirements (because the deep learning need GPU and

powerful computing power). There is still a lot of research space for designing the network and tracking process to achieve speed and effect improvement.

2.5 Deep Learning Target Detection

Object detection belongs to the field of computer vision, which existed before the concept of deep learning was proposed. Before deep neural networks, the main way to achieve target detection was still based on statistics or knowledge. In view of the amazing performance of deep neural networks, the industry's research on target detection has shifted in the direction based on deep neural networks. In view of the powerful logic abstraction ability and feature extraction ability of deep neural networks, the current target detection is almost all based on the deep learning framework. Target detection based on deep learning has a wider range of detection objects, covering various objects in life, such as bicycles, animals, people, cars, and so on.

Target detection based on deep learning can be divided into two categories:

1. Deep learning classification algorithm.
2. Deep learning regression algorithm.

The classification algorithm is also known as the two-step method. This type of algorithm first extracts the Region Proposal (candidate area) of the detected image. The difference with sliding windows is that Region Proposal uses the texture, colour and other information in the image, so this type of algorithm produces fewer candidate windows and higher

quality. After Region Proposal extracts the candidate windows, the next step is to use a deep neural network to automatically extract features and classify these candidate windows. Then merge the areas containing the same target, and finally output the target area to be detected.

This type of algorithm was first proposed by Girshick et al. [66]. An R-CNN (Region Based Convolutional Neural Networks) has been used to generate candidate regions before detection, which reduces information redundancy and improves detection speed. And this algorithm overcomes the poor robustness of traditional manual feature extraction, which is limited to low-level features such as colour and texture. However, the overlap of the candidate regions causes the R-CNN algorithm to repeatedly convolves the same region, which increases the storage space occupation and reduces the training speed. In addition, the cropping and zooming of the picture will lose the original information of the picture, resulting in poor training effects.

He et al. [67] discovered these problems and developed SPP Net to solve them. SPP Net chooses to convolve firstly and then generate the area to reduce storage and speed up training. A special pooling layer is used before the fully connected layer to break the constraint of fixed size input.

Based on the SPP Net, Girshick changed R-CNN from a serial structure to a parallel structure and regressed the bounding box while classifying [68]. This change not only speeds up the prediction, but also improves the accuracy. Ren et al. [69] proposed a concept of Region Proposal Networks, which uses neural networks to learn by themselves to generate

candidate regions. These neural networks can learn more high-level and abstract features, and the reliability of the generated candidate regions are greatly improved.

The detection methods introduced belong to a two-stage scheme, which includes two steps: candidate region generation and region classification. Compared with traditional target detection algorithms, this type of algorithm does not completely abandon the design concept of Region Proposal, but the use of corresponding algorithms greatly reduces the number of Region Proposal.

With the advent of the You Only Look Once (YOLO) [16] algorithm, deep learning target detection algorithms began to have a single-stage method. Different from the two-step detection algorithm represented by the R-CNN series, YOLO discards the region proposal stage, and directly completes feature extraction, region proposal regression and classification in the same convolutional network, making the network structure simple, and the detection speed is nearly 10 times faster than Faster R-CNN. This enabled the deep learning target detection algorithm to meet the needs of real-time detection tasks under the current computing power. The algorithm scales the detected image to a uniform size.

In order to detect targets at different locations, the image is equally divided into grids. If the centre of a target falls in a grid unit, the grid unit is responsible for predicting the target and outputting its bounding box and confidence. The confidence here refers to what object is contained in the grid and the accuracy of predicting this object.

YOLO is indeed very fast (78fps for YOLOv3), but the detection quality of small targets is not good, and it is not easy to distinguish when multiple targets appear in a grid cell. Liu et al.

[70] proposed a single-shot detector (SSD) algorithm based on YOLO, which combined anchors from the Faster RCNN. Feature maps of different sizes are combined for prediction, thereby improving the recognition accuracy. And the high-resolution (large size) feature map contains more information about small objects, so SSD can better identify small objects. In addition, the biggest difference from YOLO is that SSD does not use fully connected layer to reduce many parameters and increase speed.

2.6 Illegally Parked Vehicle Detection and Tracking

Illegal parking affects the overall environment of urban road traffic and leads to chaotic urban traffic order. As shown in Chapter 1, illegally parked vehicle detection and tracking systems are essential for many scenarios. In order to replace or help the human operators of the surveillance system, researchers have developed a variety of methods to detect and track illegally parked vehicles in urban traffic scenes using the various visual tracking and target detection technologies shown above.

Lee et al. [71] proposed the use of background segmentation and consecutive frame subtraction to detect illegal parked vehicles. Image projection is used to convert two-dimensional data into one-dimensional data, which helps reduce the complexity of background segmentation and target tracking. Bevilacqua et al. [72] proposed target detection by background difference and then calculated its centroid position. They determined whether the target is stationary by tracking the position of the target centroid. Alkhawaji et al. [73] proposed a system for detecting and tracking illegal vehicles using a

Gaussian Mixture Model (GMM) and Kalman Filter. The GMM is used in this system as a reliable background model construction and foreground extraction method. In the tracking component of the system, the Kalman filter is used to track the detected target.

Another algorithm using GMM for background construction was proposed by Mu et al. [74]. The relatively complete foreground target is processed and tracked by the morphology filter, and the shape of vehicle and roundness of wheel are used to identify the vehicle. Sarker et al. [75] proposed the Illegal Parking Detection method based on GMM. The foreground targets extracted by an adaptive GMM are analysed and detected through time sequences of pixel-level features. The grey value of a pixel point of the no parking zone is counted on a time sequence to distinguish whether the object stops in the Region of Interest (ROI), and the seed filling algorithm is used to distinguish the target type.

Hassan et al. [76] proposed a method for detecting a stationary target based on image segmentation. They used the high update rate of GMM for foreground extraction of the video, as this can accommodate lighting changes, but it means that stationary object to be part of the background quickly. To solve this problem, they introduced Segmentation History Images (SHI) to post-filter the results of GMM and determine when the object was when stationary. [77] shows a machine learning tracking method that combines Haar-Cascade and background subtraction models. Gaussian mixture model and canny method are used to separate foreground objects and edge detection respectively.

A vehicle tracking method that combines SVM and background subtraction is proposed by Jo et al. [78]. The HOG (Histogram of Oriented Gradients) feature was introduced to train a

SVM classifier from the vehicle database, and this classifier is used to classify the foreground objects. An online learning tracking algorithm using the frame-to-frame subtraction was proposed by Zhao et al. [79]. This algorithm uses a method based on texture and colour histograms. Online learning methods are used to update the weights of these features in order to achieve the best solution. The approach proposed by Maddalena et al. [80] is to automatically generate models by self-organising methods based on the background and foreground without prior knowledge of the pattern classes. Albiol et al. [81] used the Harris algorithm to detect corner points on the street, and they combined the corners with time to create a spatiotemporal map. The detection of stationary vehicles was achieved by analysing the spatiotemporal map. This method does not require subtracting the background or tracking any targets.

Compared with traditional target detection methods based on background subtraction, deep learning-based target detection algorithms also have broad application prospects. Xie et al. [82] proposed a real-time illegal parking detection system that uses a combination of deep learning algorithms and motion detection. The system uses the SSD algorithm as the target detection module, and the target detected as a vehicle will be judged as a parking vehicle and tracked through motion analysis and location analysis. Even in complex weather, the deep learning based SSD network was still robust.

Tang et al. [83] proposed a real-time illegal parking detection algorithm based on Contextual Information Transmission. This research improves the performance of the SSD algorithm in small target tracking and feeds deep layer information back to the shallow layer through

deconvolution. According to the test, the algorithm is 1.5% higher than the benchmark SSD in the detection of small targets.

Other deep learning-based algorithms also have many applications in the field of illegally parked vehicle detection. Ng et al. [84] proposed a CNN-based illegal parking detection technology. The technology is divided into two steps. The ROI extraction module extracts the latest no parking zone and then a sliding window search module performs window-based searching on the ROI. The second step uses a pre-trained ALexNet to check whether there is an illegally parked vehicle in the search window. The method proposed by Chen et al. [85] is to perform target detection processing on video using a deep learning network YOLOv3, and then template matching is applied in motion detection of a vehicle.

2.7 Semantic Segmentation and Background Subtraction

Semantic segmentation is a technique that associates tags or categories with each pixel of a picture. It is used to identify the collection of pixels that constitute a distinguishable category. For example, self-driving cars need to recognise vehicles, pedestrians, traffic signals, sidewalks, and other road features. Therefore, semantic segmentation can be used in many situations, such as autonomous driving and medical imaging

Before deep learning methods became popular, semantic segmentation methods such as TextonForest [86] and Random Forest Classifiers [87] were used more frequently. However,

after the popularity of deep convolutional networks, deep learning methods have improved a lot over traditional methods, so traditional methods will not be discussed in detail here.

The current deep learning semantic segmentation models are basically developed based on FCN (Fully Convolutional Network) [88]. FCN classifies the image at the pixel level, thereby solving the problem of image segmentation at the semantic level. This technology can accept input images of any size and retain the spatial information in the original input image.

For example, Zhang et al. [89] proposed a method based on FCN to solve the problem of vehicle counting in city cameras. The algorithm is a novel method called FCN-rLSTM that utilises the strengths of FCN for dense visual prediction and strength of LSTM for modelling temporal correlation. It can be found from the test on the UCSD dataset that the algorithm has performance far exceeding baseline methods.

However, the results obtained are insensitive to details in the image due to upsampling. Hence, some new methods have been developed to improve this problem. For example, another idea of semantic segmentation is based on the method of improving feature resolution. The purpose of this type of method is to restore the resolution dropped in the deep convolutional neural network, so as to obtain more context information.

The DeepLab [90] architecture developed by Google combines a deep convolutional neural network and a fully connected Conditional Random Field (CRF) was applied to the task of semantic segmentation, and the purpose is to do pixel-by-pixel classification. Liu et al. [91] applied semantic segmentation to the research of underwater scenes. The algorithm uses DeepLab as the basic framework, and introduces a module called unsupervised colour

correction method (UCM) into the encoder structure of the framework to improve image quality. Compared with the original method of Deeplab, this method improves the segmentation accuracy by 3%.

In addition to the above two architectures, there are many novel methods based on structures such as SegNet [92], RefineNet [93] and PSPNet [94], which can apply semantic segmentation to multiple scenarios. For instance, Audebert et al. [95] proposed a deep learning-based pre-detection segmentation method. This method was used to segment and classify various wheeled vehicles in high-resolution remote sensing images.

The methods introduced above all implement a semantic segmentation method based on deep learning. However, in order to achieve state of the art performance, semantic segmentation algorithms based on deep learning usually have many parameters and high computational complexity, which can easily overwhelm the resource capacity and capabilities of the video surveillance platform. Some studies such as [96] and [97] have implemented certain real-time applications, but these systems were all running on high-speed CPUs and powerful GPUs. For example, the [96] runs the program on the Nvidia Titan X, and so does [97]. Compared with some lightweight moving object detection algorithms, such as GMM (Gaussian Mixture Mode), semantic segmentation occupies a lot of computing power, which leads to this technology not being used in the target detection of this research. Considering that this project needs to implement a illegally parked vehicle monitoring system that can run in real time, the semantic segmentation algorithm is not conducive to reducing computational power consumption. At the same time, the model of using cloud computing capabilities can indeed solve the problem of lack of computing

power in local devices, but in the face of data transmission from millions of cameras, signal delay and congestion will become a huge challenge. Therefore, a lightweight detection method that can be run locally is the best solution.

As a traditional image processing method, background subtraction is the main pre-processing step in many computer-vision based tasks. For example, customer statistics need to use a static camera to record the number of people entering and leaving the room, or a traffic camera that needs to extract transportation information. If there is a complete still background frame, the foreground object can be obtained by calculating the pixel difference by the frame difference method. However, there will be no such image in most cases, so the background needs to be extracted from any obtained image. When a moving object has a shadow, the situation becomes more complicated because the shadow is also moving. For this reason, the background subtraction algorithm is introduced to separate the foreground of the moving object from the video, so as to achieve the purpose of target detection.

OpenCV has implemented several very easy to use algorithms, namely KNN [98], GMG [99], MOG [100] and MOG2 [101]. The results and comparison of these four background subtraction methods can be seen in Chapter 6.3.

2.8 Conclusion

This literature review has covered the major advances in the field of visual tracking and illegally parked vehicle tracking which supported the feasibility of automated and intelligent parked vehicle monitoring systems. The findings include visual tracking technology, multi-object tracking algorithms, and methods for monitoring illegally parked vehicles. It should be noticed that today's mainstream illegally parked vehicle tracking algorithms are based on the fusion of deep learning and image processing method. Through the research of relevant literature, it can be found that the deep learning based tracking usually requires the data association processing to achieve tracking by detection.

Therefore, the detection and data association of objects becomes crucial. There are many detection algorithms, for example, the famous Faster-RCNN algorithm is used to detect targets, and traditional methods can also be used to detect multiple targets. There are also many data association methods. One of the most popular methods is the Hungarian algorithm used in SORT [17] and deep SORT [102]. In addition, traditional image processing methods still play an important role in the tracking process.

Despite these achievements [71]-[85], there is still a lot of work to be done to optimise the illegally parked vehicle monitoring system to improve performance. These reviews and various techniques help advance this research. Based on the information in the literature review, no one has proposed a method that integrates tracking algorithms based on deep learning detection and feature point matching for use in an illegally parked vehicle monitoring system. Specifically, traditional image processing techniques such as

background segmentation cannot adapt to complex environmental changes. At the same time, the detection and tracking methods based on deep learning such as semantic segmentation and deep learning descriptors provide good detection and tracking results but consume a lot of computing power and cannot be used outside the laboratory. Therefore, this thesis proposes a method based on deep learning detection and key point matching to achieve a low power consumption and high-efficiency illegally parked vehicle monitoring system. Specifically, the proposed method uses a one-step method to reduce the consumption of computational power, and the feature point matching ensures the tracking effect of the target.

The Chapter 1 shows the challenges of illegally parked vehicle monitoring. This research provides a novel and effective method to solve these problems. A tracking system based on the combination of YOLO algorithm and SORT algorithm has been proposed in Chapter 3. Then in Chapter 4 and 5, the tracking system has been modified. A tracking module based on feature point matching has been added to improve the tracking effect, which involved the changing of feature point extraction. Finally, the actively selected feature points (ASFP) has been integrated on the method discussed in Chapter 5 to achieve robustness to light changes, occlusion and long-term tracking. The effectiveness of this system has been proven through extensive tests.

Chapter 3 Deep Learning Detection and MOT Tracking on Illegally Parked Vehicles

3.1 Introduction

Automatic detection and tracking are essential for modern illegally parked vehicle monitoring systems. An illegally parked target can be defined as a vehicle that is stationary in a no parking zone at a specific time and parked for a period of time. The monitoring of this type of target means that the system needs to track the target in a continuous video sequence and keep the target's digital ID tag unchanged. The following situations will affect the monitoring process: similar targets appear in the scene; the target is partially or completely occluded; the background of the object is changing, such as light changes and background movement. Thus, for the case where a plurality of objects in the scene with the same fixed background, tracking solution can be converted to the multi-object tracking system of a particular time and location.

On the basis of the existing detection and multi-object tracking algorithms, this chapter proposes a tracking-by-detect method for illegally parked vehicles monitoring, in which the detection module uses YOLOv3 [103] algorithm. Then the tracking module will be implemented by the SORT algorithm. If the tracked target is identified as an illegally parked vehicle in the state judgment module, an alarm will be generated.

The technology has been tested on the i-LIDS dataset [18] and ViSOR dataset [19]. The experimental results show that although the proposed method for monitoring illegally

parked vehicles makes mistakes in the case of frequent occlusions, and there is an ID switch situation caused by data association failure, it can be used when occlusion do not occur.

3.2 Chapter Organisation

The arrangement of this chapter is as follows. The following section briefly describes the test environment of the system. Section 3.4 introduces the YOLOv3 algorithm used in this chapter to detect objects from video sequences. Section 3.5 introduces a method to track the vehicle. Section 3.6 describes how the tracked object is determined to be illegally parked vehicles. The results are presented and discussed in Section 3.7 and a summary is given in Section 3.8.

3.3 Overall Test Environment and Software Implementations

In order to test whether the multi-object tracking algorithm SORT can be used to track illegally parked vehicles, the ViSOR dataset [19] and i-LIDS [18] datasets have been used in this chapter. The ViSOR dataset provides the simplest monitoring scene. Although there are multiple objects in the scene, there are no more than 2 illegally parked vehicles at the same time. At the same time, there are no occlusion or complicated light changes, and all moving targets move slowly.

The i-LIDS dataset is closer to the real traffic environment. High-speed moving vehicles, drastic changes in light and shadows, multiple targets appearing in the no parking zone at the same time, frequent occlusion or even complete occlusion, the above conditions will interfere with the tracking process and generate false alarms. The detected vehicles enter the no parking zone in several different ways, including slow taxiing, fast passing and illegal parking, each of them needs to be classified. In addition, various bad-weather conditions can also lead to tracking failures due to the overexposure of the picture. Taking into account the differences in the scenes of different datasets, the no parking zone is user selected as the region of interest (ROI) as shown in Figure 6.

The main processing has been shown in Figure 7. All objects appearing in the ROI will be automatically classified by the state judgment module, and the targets determined as illegally parked vehicles will be tracked and alarms will be generated. The video sequence has been first detected by the YOLOv3 algorithm, and all non-vehicle detection results have been removed by a category filter before being sent into the target tracking module. In the process of all objects being tracked, the state judgment module will classify the objects according to their position and speed, thereby reducing false alarms. Otherwise, these false alarms will be generated by vehicles passing by the no parking zone. The alert can be generated in the first frame where it is defined as an illegally parked vehicle.

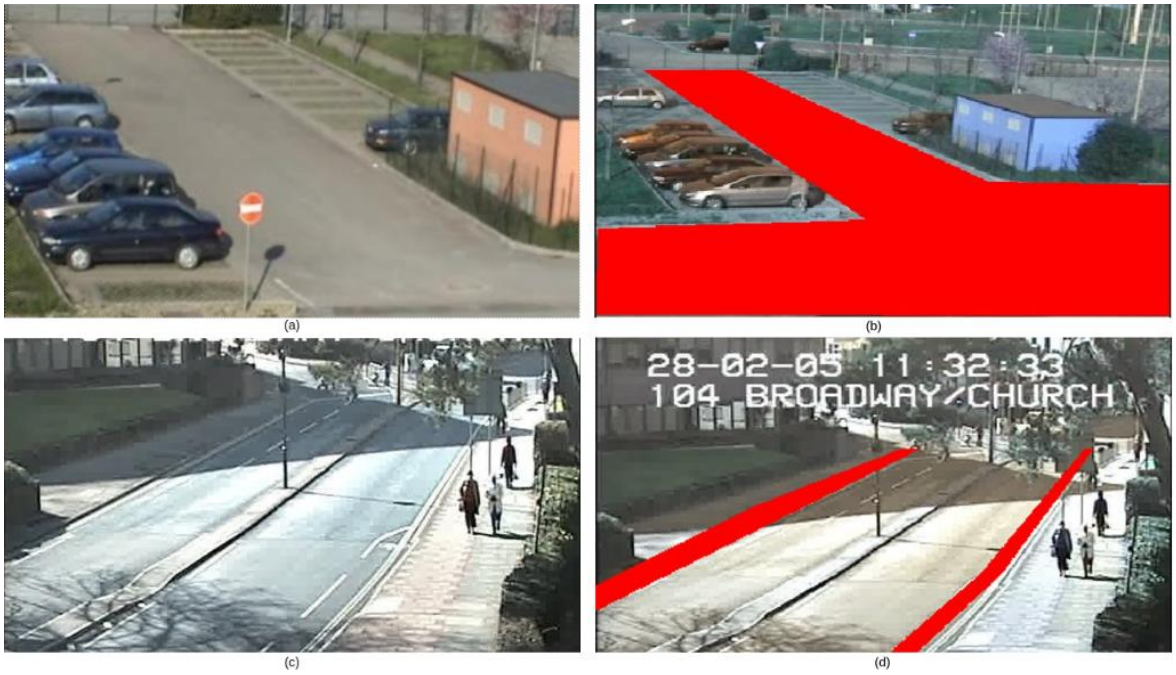

Figure 6 (a)(c) Typical traffic environment from ViSOR dataset and i-LIDS dataset. (b)(d) Manually marked no parking zone.

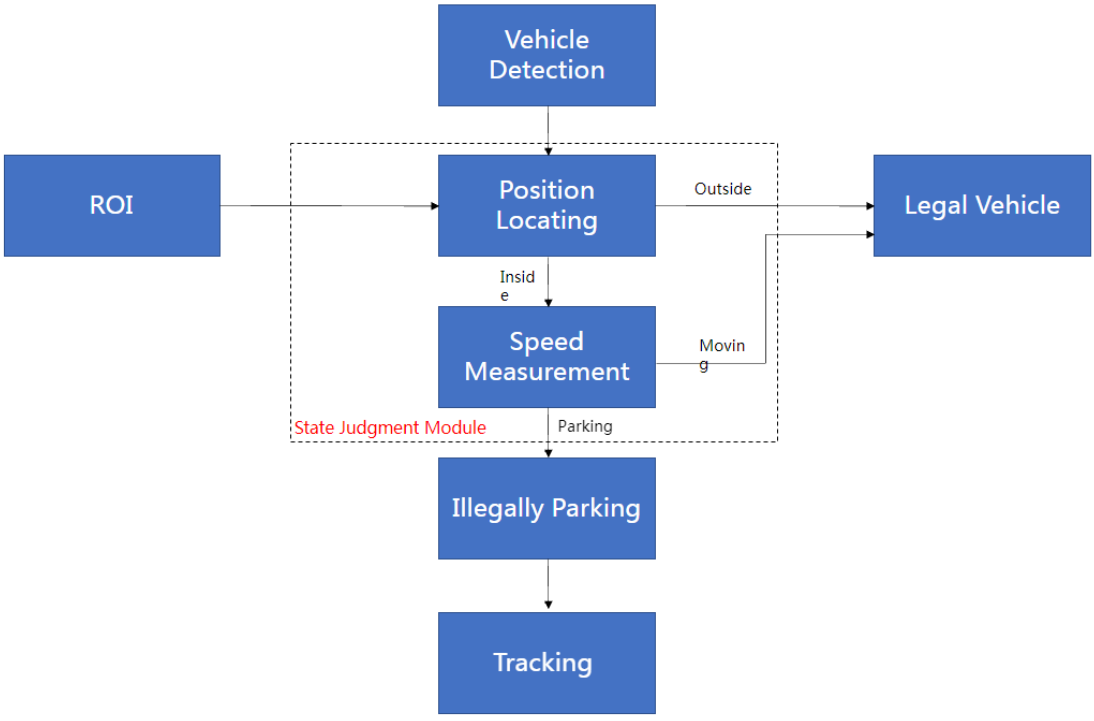

Figure 7 The framework of this system. The overall framework can be divided into three parts: vehicle detection, state judgment module and illegally parked vehicle tracking. All the detected vehicle should be judged by their speed and position. The vehicle belonging to illegally parking will be tracked.

The entire technology is developed and tested on a personal PC platform with an Intel Xeon(R) CPU E5-1620 V3 running at 3.5GHz, NVIDIA Quadro K4200 and with 15.6 GB RAM. The tracking system proposed in this chapter and the improvements in subsequent chapters are all developed by python and run on the Spyder platform of the Linux system. The image processing function in the system is provided by the OpenCV-Python library, and the detection weight of the deep learning algorithm YOLOV3 has been trained on the COCO dataset [124]. The judgment module of illegally parked vehicles is developed by the author on the basis of the original SORT algorithm [17]. The tracking program based on feature point matching and background subtraction is also developed by the author.

As the most commonly used evaluation index in object detection, mean Average Precision (mAP) represents the accuracy of the detection algorithm. The mAP is used to compare the ground truth bounding box with the detected box and return a score. The higher score indicates the higher accuracy of detection.

The calculation of mAP involves the following steps. First of all, several important indicators about the detection results need to be counted:

1. True Positive Alarms (TP): The system generates alarms for real events
2. False Positive Alarms (FP): The system generates an alarm but no real event occurs
3. False Negative Alarms (FN): The real event exists but the system does not generate an alarm

This data allows the following values to be calculated:

The Recall (detection rate):

$$R = \frac{TP}{TP + FN} \qquad Equation\ 3-2$$

and the Precision (probability of an alarm being genuine):

$$P = \frac{TP}{TP + FP} \qquad Equation\ 3-3$$

Recall, *R*, is the ratio between true positive events and the total events in the ground truth. Precision, *P*, is the ratio between true positive events and all events obtained from the experiment. In addition, the combination of recall and precision which known as *F1* measure is also used to evaluate the performance of the method. The calculation method for *F1* is as follows:

$$F_1 = \frac{2PR}{P + R} \qquad Equation\ 3-4$$

The second step is to rank the results according to the confidence of each detection. Draw the PR curve (precision-recall curve) of the precision and recall of each test result in turn by ranking, and then the area under the PR curve is the AP (Average Precision) score. The area can be obtained by Equation 3-5:

$$\mathrm{AP} = \int_0^1 p(r)dr \qquad Equation\ 3-5$$

In the test, the YOLOV3 algorithm obtained a mAP-50 (the IoU=0.5 according to COCO dataset's request) result of 57.9 on the COCO dataset. Figure 8 shows a series of examples of vehicles, buses and trucks in the COCO dataset.

Figure 8 Some examples of cars, buses and trucks from COCO dataset. The YOLOV3 algorithm has been trained to detect various types of vehicles, such as cars, buses and trucks that were marked in the images.

The method proposed in this paper is repeatedly tested on more than 10 hours of video to ensure the accuracy of the results. All functions and errors during the running of the software can be visually detected and improved. All functions and errors during software operation can be observed and analysed through the visual window, and all output data are stored locally for analysis.

3.4 YOLOv3 Object Detection Algorithm

The object detection method used to provide the potential targets to be tracked within this chapter, and also applies to this thesis, is based on the YOLOv3 algorithm presented in [103]. The basic idea of the YOLO algorithm has been introduced in the literature review. Compared with the original display, the third version has better detection capabilities for small objects. The basic framework of the deep learning algorithm YOLOV3 is a convolutional neural network with 53 layers. The image is input into the classification model with a size of 256x256 and features are extracted from different scales. Through weighted voting on all features, a tensor containing all feature information is formed, and this tensor will become the basis for classifying the target. Finally, the result of this classification is transformed into detection.

Therefore, the YOLOV3 algorithm detected objects in the video, and the identified objects are marked by bounding boxes. In order to verify the tracking performance of the target tracking module for illegally parked vehicles, only the detection results with vehicle-related tags will be kept, which reduced the interference of other types of targets in the multi-object tracking and speeds up the calculation. The following Figure 9 shows a typical scene from the i-LIDS dataset and YOLOV3 detection results. It should be noted that the detection weights of the YOLOV3 algorithm used in this project are trained on the COCO dataset [124] to obtain the detection ability for a variety of vehicles.

Figure 9 The vehicles detection results by YOLOv3. All detected targets are represented by blue bounding boxes.

3.5 SORT Tracking

As shown in Figure 9, the output of the YOLOv3 algorithm contains all the objects that appear in the video scene. All objects that do not belong to the target of interest has been deleted through the tags of the objects. As shown in Figure 10, only the targets classified as cars, buses, and trucks will be tracked. This change helps the multi-object tracking module focus on the illegally parked vehicle monitoring. As a classic algorithm for multi-object tracking, SORT can provide high-speed and efficient tracking for the illegally parked vehicle tracking.

SORT has a Tracking-by-Detection framework, which has four basic components: target detector, state prediction, data association and track management. These are also the basic

components of many multi-object tracking algorithms that follow the Tracking-by-Detection framework. The original SORT algorithm uses Faster R-CNN of VGG16 architecture [122] as the target detector. Section 3.4 introduces the YOLOv3 algorithm as the target detector in this system to provide faster detection speed, which has 3.8 time faster than Faster R-CNN [103].

SORT uses a Kalman filter [40] to actively predict the state of the target, and matches the predicted result with the actual detected target frame. The relationship between track and detection is regarded as a bipartite graph (also called a bi-graph, is a set of graph vertices), and the weight of each edge of the bipartite graph is defined by the Intersection over Union (IOU) of its two endpoints (a track and a detection respectively). SORT uses the Hungarian algorithm [123] to find the best match in this bipartite graph and sets a minimum IOU threshold for the match to reduce the number of false matches. Figure 10 shows some example of SORT tracking.

Figure 10 The vehicle tracking results by SORT. All detected targets are represented by colored bounding boxes. The number in the upper left corner represents the digital ID of the target.

3.6 State Judgment and Tracking

Once input the detected objects to the multi-object tracking module, these objects can be regarded as potential illegal targets. Considering the purpose is to focus on tracking stationary targets in the no parking area, a state judgment module and tracking module should be added to run in parallel. Chapter 2 reviews some methods based on such as time sequences of pixel-level features or window-based searching ([78]-[81]), but these methods require complex analysis processes, which consume a lot of computing power. In order to judge the state of the tracked target, the proposed judgment method is based on the following classification of all vehicles. According to the analysis of the behaviour of all vehicles, the vehicles in the video scene can usually be divided into the following categories:

1. The vehicle is driving outside the no parking zone
2. The vehicle is parking outside the no parking zone
3. The vehicle is moving in the no parking zone
4. The vehicle is parking in the no parking zone

Therefore, the state judgment should include vehicle position information and speed information. For categories 1 and 2, the proposed method can easily identify by the location information of the target. For categories 3 and 4, apart from using position information to determine whether the target is in the ROI area (no parking zone), it is also necessary to measure the speed of the target.

By observing and analysing the trajectory of vehicles in a large number of videos, firstly measuring the speed of all tracked vehicles will help delete vehicles that have no intention of parking. This allows the illegally parked vehicle monitoring system to screen out vehicles in category 1 and 3. In the field of surveillance video speed measurement, there are two ways to calculate vehicle speed. One is the combination of video image and field measurement. The vehicle movement distance is matched with the monitoring screen through field measurement, and the movement time is obtained by video frame rate calculation. The second is to use vehicle technical parameter measurement. In the video image, the vehicle's movement distance and time are respectively obtained by calculating the vehicle's technical parameters and the video frame rate.

Considering that the proposed method needs to be applicable to a variety of road scenes, and the actual distance measurement conflicts with the concept of the automatic monitoring method, the former method is not applicable to this thesis.

Therefore, a speed test on the basis of the second method has been conducted. Since the purpose of speed detection is to distinguish whether the vehicle is stationary, that is, to determine whether the target belongs to category 2 or 4. Hence, the actual vehicle movement distance is not needed in the detection processing. The relative movement distance of a vehicle can distinguish the movement state. Whether the vehicle is moving can be judged by calculating the distance between the same position of the target on the pixels in two adjacent frames. Here the Euclidean distance of the target in two adjacent frames has been chosen to calculate.

Euclidean distance is a commonly used distance definition that refers to the true distance between two points in m-dimensional space, or the natural length of the vector. The Euclidean distance in two-dimensional and three-dimensional space is the actual distance between two points. In the two-dimensional space, such as the scenes in this thesis, the Euclidean distance can be calculated by the following equation:

$$P = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad Equation\ 3-1$$

In Equation 3-1, x1, y1 indicate the position of YOLO bounding box‘s centre point in the current frame, and x2, y2 indicate the position of the centre point in the next frame. If the distance *P* between two points can equal to 0, it indicates that the target is in a static state.

In the test, each position of the bounding box has been tried to calculate the moving distance, but due to camera noise and partial occlusions, the bounding box did not remain completely still for stationary objects. This leads to the moving distance between frames cannot be equal to 0 even for the completely stationary target. Therefore, the centre point of the bounding box has been chosen from many distance calculations schemes which gives the smallest bounding box drift result. Figure 11 shows the result of distance measurement by selecting the centre point of the bounding box.

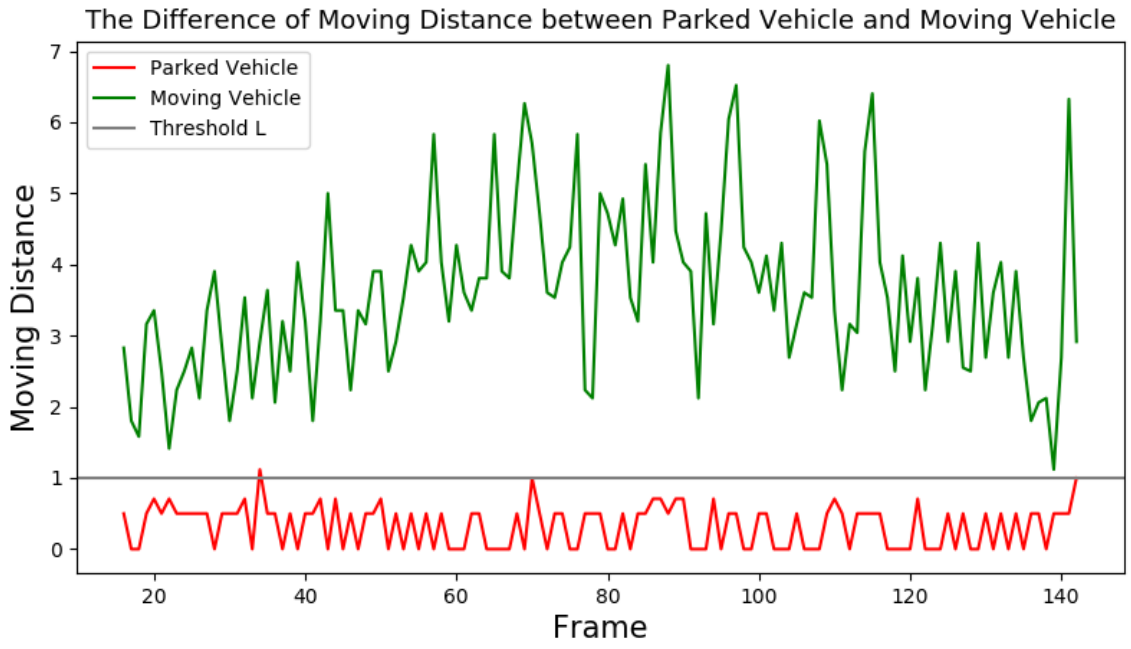

Figure 11 The line graph shows moving distance of the different vehicles. It can be found that there is a clear distinction between moving vehicles and stationary vehicles.

As shown in the Figure 11, the movement state of the vehicle can be judged by using some threshold *L*. According to the measurement of a large number of datasets, ideally, the

moving distance of the illegally parked vehicle is 0, but the bounding box generated by the system will be slightly shifted so that the distance cannot be constant at 0.

However, since the moving distance of the driving vehicle will not be less than 1 with Euclidean Distance, the threshold *L* can be set to 1 to distinguish the movement state of the vehicle. At the same time, the location of the vehicle also needs to be detected. The bounding box generated by the detector was used to determine whether the target is in the ROI. If the target completely enters the ROI, the bounding box will have a huge intersection with the no parking zone. In addition, considering the difference between the 3D viewing angle of the monitoring screen and the 2D viewing angle of the detected image, the target passing at the edge of the ROI will also have a bounding box that partially overlaps the no parking zone. Therefore, a ratio *R* is given to calculate the ratio of the intersection of the bounding box and the no parking zone to the total area of the bounding box. If the intersection value *R* of the target is less than a given threshold *S*, it means that the target is not in the ROI. Figure 12 shows the relative positions of the vehicle and the no parking zone under different intersection values *R*.

Figure 12 The intersection between bounding box and the red zone has been calculated and represented by *S*. The position status of vehicles can be judged by *S*. If the intersection is greater than *S*, it means that at least part of the vehicle has entered the red zone (a) *S*=5.7% (b) *S*=18% (c) *S*=51% (d) *S*=26%.

In a 2-dimensional image, the driving direction of vehicles with the same volume will affect the intersection area. For example, for the four targets in Figure 12, it is obvious that the intersection in (a) and (b) for the vehicles driving towards the camera is small, while (c) and (d) are large. Therefore, the threshold *S* should not be too large to filter out all targets driving towards the camera. However, the vehicle still can be clearly classified in the above four situations by combining with the target moving distance information. Therefore, through testing on the i-LIDS and ViSOR datasets, *S*=10% is an appropriate threshold, and this value is also selected in the following chapters.

In summary, the target will be defined as an illegally parked vehicle if the moving distance is greater than *L* and the overlap area is greater than *S*.

3.7 Discussion and Results

The proposed method has been tested on the ViSOR dataset and i-LIDS dataset which provides two different test scenarios: empty parking lots and roads with complex environmental changes. The ViSOR dataset consists of 4 video sequences in 2-3 minute and contains multiple events. All events occurred in a parking lot and the illegally parked vehicles were not blocked. The I-LIDS dataset was opened by the Centre for Applied Science and Technology (CAST, formerly HOSDB) to researchers for using in parking vehicle monitoring. I-LIDS contains 3 sets of video sequences with a total length of more than 10 hours and different lighting conditions and weather conditions (day, night, rain, lightning, rapidly changing shadows, etc.). In the test of these two datasets, the ViSOR requires an alarm to be generated within 10 seconds when the vehicle is stationary in the no parking zone. The i-LIDS dataset requires the alarm to be generated within 10 seconds after the target is stationary in the no parking zone for more than 60 seconds (In order to clearly show the sensitivity of the proposed method to parking vehicles, In the presentation section of this thesis, the program will be temporarily modified to generate an alarm when the target is stationary in the no parking zone. In all the statistical results shown in this thesis, the alarms are generated strictly in accordance with the official guidelines of i-LIDS.).

Figure 13 shows all the events in the 4 video sequences of the ViSOR dataset. The proposed method detects all events and avoids two interference events. In (a), (b) and (c), the alarm event is generated immediately when the vehicle is stationary in the no parking zone. Since the tracking module is based on a multi-object tracking algorithm, as shown in the Figure 13, multiple bounding boxes are generated to locate the vehicle. In order to show the results of the proposed method the word “alarm” is displayed on the image above the problem vehicle. The information is also written to file. Visually and by comparing the results to the ground truth data the accuracy of the tracker can be measured. For (d), there are two vehicles entering the video scene, but the vehicles passing through the no parking zone and finally stop in the area where parking is allowed, (case 14 and 12 in (d)), so the alarm is not generated.

Figure 13 The detection and tracking results from VISOR dataset. The word ‘alarm’ above the vehicle represent the tracking of the target. (a) Case 26 is an illegally parked vehicle entered the no parking zone. (b) Case 19 is an illegally parked vehicle entered the no parking zone. (c) Case 26 is an illegally parked vehicle entered the no parking zone. (d) No illegally parked event occurred, case 12 and 14 are two interference events.

The test video sequences in the i-LIDS dataset are divided into three groups based on three different viewing angles and scenes. Frequent occlusions, sunny days, rainy days, shadow movement caused by cloudy, overexposure caused by night and lightning are all included in the test scene. Table 1 below shows the detailed test results of the proposed algorithm. All event alerts are strictly in accordance with the official requirements of the i-LIDS dataset.

File Name	Duration	Total Event	True Positive	False Positive	False Negative
PVTEA101a	01:21:29	26	16	0	10
PVTEA101b	00:24:40	8	1	0	7
PVTEA102a	00:51:56	18	11	0	7
PVTEA103a	01:03:54	16	8	0	8
PVTEA201a	00:18:50	4	2	1	2
PVTEA202b	01:05:23	17	4	2	13
PVTEA301a	00:15:29	3	1	0	2
PVTEA301b	00:28:07	7	5	0	2
PVTEA301c	00:27:09	6	3	0	3
Total	**06:16:57**	**105**	**51**	**3**	**54**

Table 1 Result from the proposed method on videos containing illegally parking events from i-LIDS dataset

Table 1 shows that in all 105 events, the method based on deep learning detection (YOLOV3) and multi-object tracking (SORT) successfully tracked 51 events (see Figure 14 for tracking examples) and lost 54 events. The overall recall is only 49%. At the same time, the method generated 3 false alarms (precision 94%). These results are significantly lower than other authors' methods of illegally parked vehicle tracking (94% in [104], 89% in [78]). According to the observation and analysis of each event, it can be found that the systematic flaws of this method led to a large number of false negative events. First, as described above, the tracking module of this method is based on the multi-object tracking algorithm SORT.

According to the framework of the SORT algorithm, in the process of target tracking, Kalman filter is used to establish the following prediction model:

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \qquad Equation\ 3-5$$

Where u and v represent the horizontal and vertical coordinates of the centre of the target, s and r represent the size and proportion of the bounding box of the target. And the last three values indicate the prediction of next frame.

The detected bounding box is used to update the target state, in which the velocity component is optimised through the Kalman method. If no detection is associated with the target, the state will be updated by the linear velocity model. Therefore, as shown in the model, no exterior features of the tracked target are used, but only the position and size of the detected bounding box have been used for the motion estimation and data association of the target, and there is no re-recognition algorithm. So, when the target is lost, it cannot be tracked again.

Secondly, the Hungarian algorithm is used for data association. The cost matrix is calculated by the intersection-over-union (IOU) distance between the detected targets and their predicted position. IOU can solve the problem of short-term occlusion of the target. However, for the long-term occlusion as shown in Figure 14, the tracking is difficult to re-establish, because if the IOU matching between the predicted position of the tracked target and the detected target is not achieved for consecutive *T* frames, the target is considered to disappear.

Finally, the tracker will be deleted if the predicted position of the target is not associated with the detection in the *T* frame (*T* was set by original SORT), which caused by occlusion or environment. After a period of time, although the original target may be tracked again, the target ID has been switched. Therefore, in the test, although some targets were tracked from the first frame until they left the ROI, which was also regarded as a false negative event because their ID had changed. During the test, *T* was increased to 20 (1 in original SORT) to give more processing time to the data association, but there were still at least 355 ID switches in the test. Figure 14 shows a typical monitoring process involving a successful monitoring event and a failed event.

For the 2450 frames from (a) to (h), the illegally parked vehicles with ID 4 and 11 are two real illegally parking events. In (a), case4 (see the vehicle with ID 4 in Figure 14) is detected and defined as an illegally parked target through position and speed detection, and the word ‘alarm’ is generated as marked on the image above the problem vehicle. After 4 seconds in (b), case 11 and 13 appeared at the same time, case 4 is still being monitored because the occlusion has not occurred yet. With a few frames later in (c), case4 is occluded by the case11 (illegally parked event) and the case 13 (fast passing vehicle), at this moment the tracking fails due to the occlusion. In the next second, as shown in (d), the tracking is regenerated because the detection result is successfully linked with the predicted bounding box.

However, it is limited by the insufficient detection capabilities of the YOLOv3 algorithm for small objects, and in the YOLOv3 detection framework, each grid can only output one prediction result. Therefore, the biggest shortcoming of this algorithm is that the detection

effect of some nearby small objects is not very good, such as the two illegal vehicles nearby in (d). As shown in the figure, the bounding box of case4 cannot stably represent the target position, causing the system to incorrectly judge the target as a moving target. Figure 14 (e) and (f) show that the tracking is unstable due to detection defects. Figure 14 (g) and (h) show the scene when the target leaves the ROI. The alarm of case 11 continues when the vehicle slowly moving in no parking zone, because the target is still in the ROI and the speed is less than the threshold *L*. The alarm will not be released until the target leaves the illegal zone, as shown in (h).

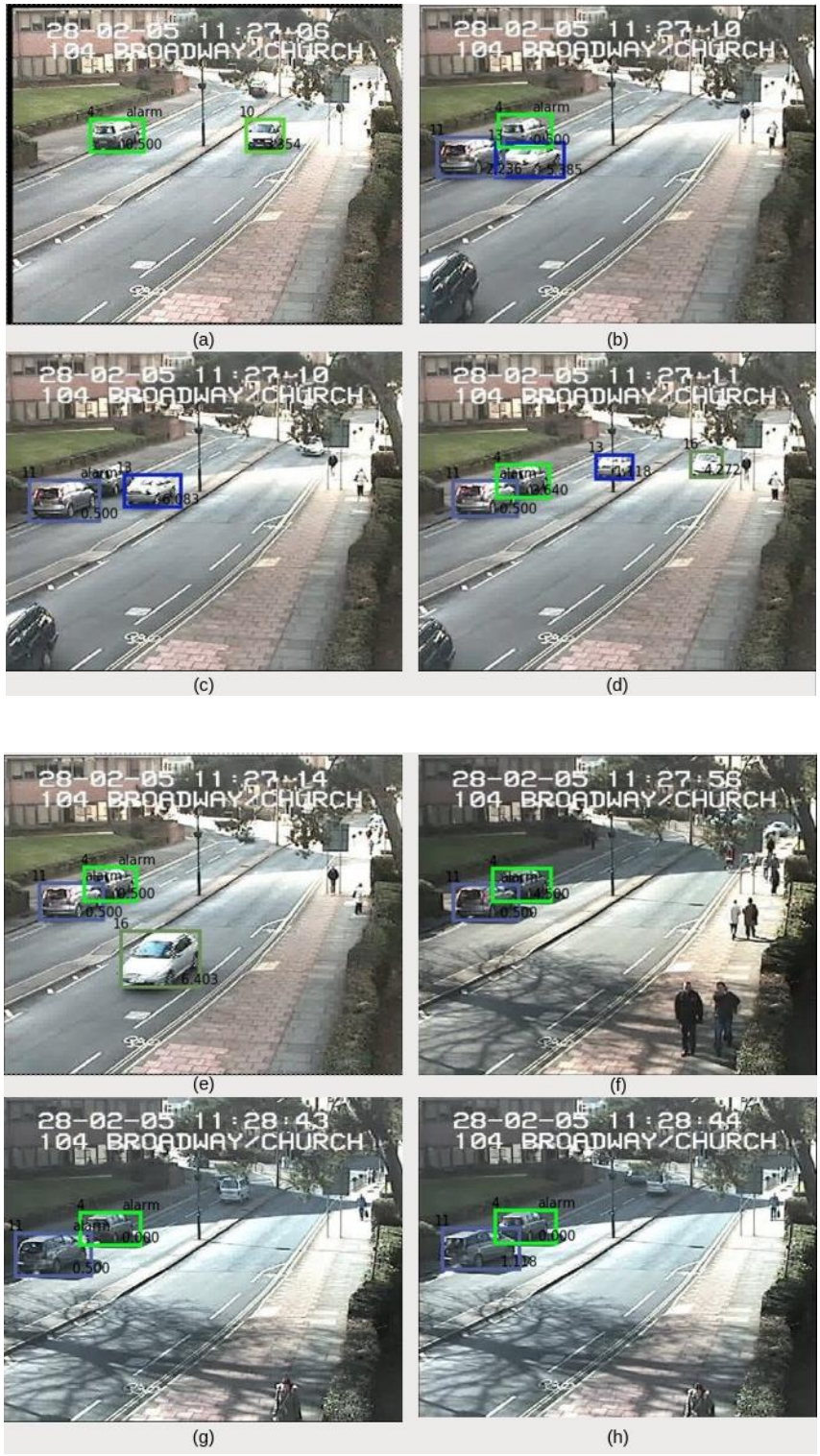

Figure 14 The failure tracking caused by proposed method. (a)The illegally parking event occurred. (b)Two vehicles moving closer to the target. (c)The tracking lost because of the occlusion. (b)The tracking re-generated again. (e)-(g)Two tracking running stable because there is no effect of occlusion. (h)One of the target left the no parking zone and tracking released.

Figure 15 shows more test results. Successful alarm events are shown in (a)-(c) because there is no serious occlusion affecting the generation of the tracker. Figure 15 (d)-(f) represent typical target loss caused by detection failure. The movement of people obviously affects the YOLO algorithm's choice of the target's bounding box position, which leads to drastic fluctuations in the position and size of the bounding box. The passing truck completely includes the target from the viewing angle of the two-dimensional image, resulting in the loss of detection of the target vehicle.

Figure 15 (a)-(c) The success tracking achieved by proposed method with raining. The system can working well without serious occlusions. (d)-(e) The failure tracking at night. The tracking failure because pedestrians interfered with speed measurement (f) The tracking lost because of the detection failure. When the truck passes by, the target vehicle cannot by recognised by YOLOV3 because it was completely covered by the truck.

3.8 Conclusion

This chapter discusses a technology based on the combination of deep learning detection and multi-object tracking to determine whether the automatic detection and tracking of parked vehicles can be achieved in actual scenes. The technology has been tested on video sequences from the ViSOR and i-LIDS datasets, and the recall and precision of the technology have been obtained. The experimental results show that the method based on the combination of the YOLOv3 algorithm and the SORT algorithm achieves 100% precision and recall rate for the ViSOR dataset, but the precision rate drops to 94% and the method fails to track on 54 events in a total of 105 illegally parked events (49% recall) for i-LIDS dataset.

The failure of the method with the i-LIDS dataset does not mean that the method is completely unable to detect and track the illegally parked target. By manually checking all failed events, it can be found that all failed events were caused by changes in the environment such as occlusion, lighting, and multi-target overlap. In other words, the system failures are due to the long-term tracking of the parked vehicle. In order to overcome these difficulties, a more powerful solution is needed to maintain the tracking ID in the first frame of the event and establish the inter-frame connection of the illegally parked vehicle in the subsequent frame sequence. Another observation about illegal vehicle monitoring in this chapter is to determine whether the proposed method of measuring the position and speed of the target can efficiently adapt to the complex traffic environment. Through testing, a suitable threshold has been found for distinguishing the position and

speed of illegal vehicles and legal vehicles. In addition, it is easy to be observed that the detection will affect the speed calculation, and the fluctuation of the bounding box is caused by occlusion, which gives wrong speed data. Chapter 4 introduces a technique based on SIFT feature descriptor matching, where the focus is to keep tracking of the illegally parked target in the case of SORT tracking failure.

Chapter 4 Tracking with Key Points

4.1 Introduction

Chapter 3 shows a method based on the combination of deep learning detection and multi-object tracking for illegally parked vehicle monitoring. The main drawback is that this method lacks robustness to occlusion, light changes, and multi-target overlap. However, since the proposed method provides good detection result of the first frame when the target is illegally parked in the test, other methods can be used to replace the tracking method proposed above for continuous tracking after the first frame. Therefore, a method based on SIFT feature descriptor matching has been proposed to continuously track the target after the target is judged as an illegally parked event. Different from global features such as colour features, texture features, and shape features, SIFT features are local features. The local image features have the characteristics of rich content in the image, small correlation between features, and the detection and matching of other features will not be affected by the disappearance of some features in the case of occlusion. In addition, SIFT features have good stability and invariance, can adapt to rotation, scale changes, and brightness changes, and cannot be interfered by viewing angle changes and affine transformations to a certain extent. These invariances first come from the gaussian pyramid established during SIFT feature extraction. The SIFT algorithm obtains scale invariance by extracting extreme points in the gaussian pyramid. The rotation invariance is obtained by collecting the gradient direction of the 4*4 area around the extreme point. Finally, the algorithm avoids affine changes in illumination by normalizing the gradient histogram

In order to explore whether SIFT feature descriptors can be used to track illegally parked vehicles under occlusion and light changes, this chapter proposes a method for tracking illegally parked vehicles based on SIFT feature point matching, where the feature points are matched by brute-force matching. The matching result will be used to keep track when the target is detected and is defined as an illegally parked vehicle. The method was tested on the video sequence of the I-LIDS dataset. Experimental results show that tracking will not be disturbed when most occlusions occur. However, some of the video scenes contain long-term large-area occlusions, small targets, and blurr targets. The test shows that the extraction of SIFT feature points will be affected by the above conditions and cause tracking failures.

4.2 Chapter Organization

This chapter is organised as follows. Section 4.3 introduces the overall framework of the SIFT algorithm for illegal vehicle detection and suggests how to make up for the shortcomings of the previous chapter. The section 4.4 shows the extraction of SIFT feature points and how to replace the SORT algorithm for continuous target tracking. This includes the characteristic analysis of the SIFT feature descriptor to deal with the tested traffic environment and the method based on BF (Brute-Force) matching to maintain the inter-frame link of the target. In Section 4.5, the proposed method is tested on the i-LIDS dataset and the results are discussed. Finally, a summary is given in section 4.6.

4.3 The Framework of Key-Point Matching in Tracking

Detecting and tracking stationary objects such as illegally parked vehicles is one of the important functions of all public video surveillance systems. As the number of cameras has increased dramatically and with the development of technology, a large number of high-definition monitoring images can be provided to the central processing platform, a reliable automated analysis and alarm generation system is required to deal with such complex situations. Many datasets were published (i-LIDS, Sherbrooke, ViSOR, Sussex Daytime) to help researchers test such detection and tracking algorithms. Hassan [104] summarised a general framework for solving such problems. The framework consists of two main steps: detection and tracking. Among them, detection is to separate the foreground target of interest from the static background by background segmentation technology. Traditional background segmentation techniques include frame difference method and GMM-based background subtraction method. Nowadays, with the development of deep learning technology, semantic segmentation technology based on high-performance GPUs is also widely used. Then once these foreground targets stay in the ROI for more than a certain time, the alarm will be triggered and these events will be tracked. In Chapter 2, some methods and researches in this area also have been reviewed. However, the problem is that in the detection process, the background segmentation technology cannot adapt to light changes well and incorrectly identifies background objects as foreground targets. Another serious problem is that in the tracking process, colour, texture, and shape features are usually used to track objects. However, the global features mentioned above are not applicable to the situation of lighting changes and occlusion.

Considering that with the development of deep learning technology, one thing for sure is that the achievements in the field of multi-object tracking can help to establish the detection and tracking of stationary targets, especially for illegally parked vehicles. The general workflow of the MOT (Multiple Object Tracking) algorithm includes the following steps:

1. grab the original frame of a given video
2. run the object detector to obtain the bounding box of the object
3. calculate different features for each detected object, usually the visual and motion features
4. calculate the probability that the two objects belong to the same target
5. assign a digital ID to each object by data association

In the previous chapter, the proposed method used the YOLOv3 algorithm and the SORT algorithm to detect and track illegally parked vehicles. The potential and shortcomings of the multi-object tracking system in the field of illegal vehicle monitoring have been shown in Chapter 3. Through analysis, it can be found that, compared with the general MOT algorithm, the tracking of illegally parked vehicles can be achieved by modifying step 3 and 4 on the basis of the above framework. Therefore, a tracking method based on SIFT feature matching has been proposed in this chapter. The framework of this method is shown in Figure 16.

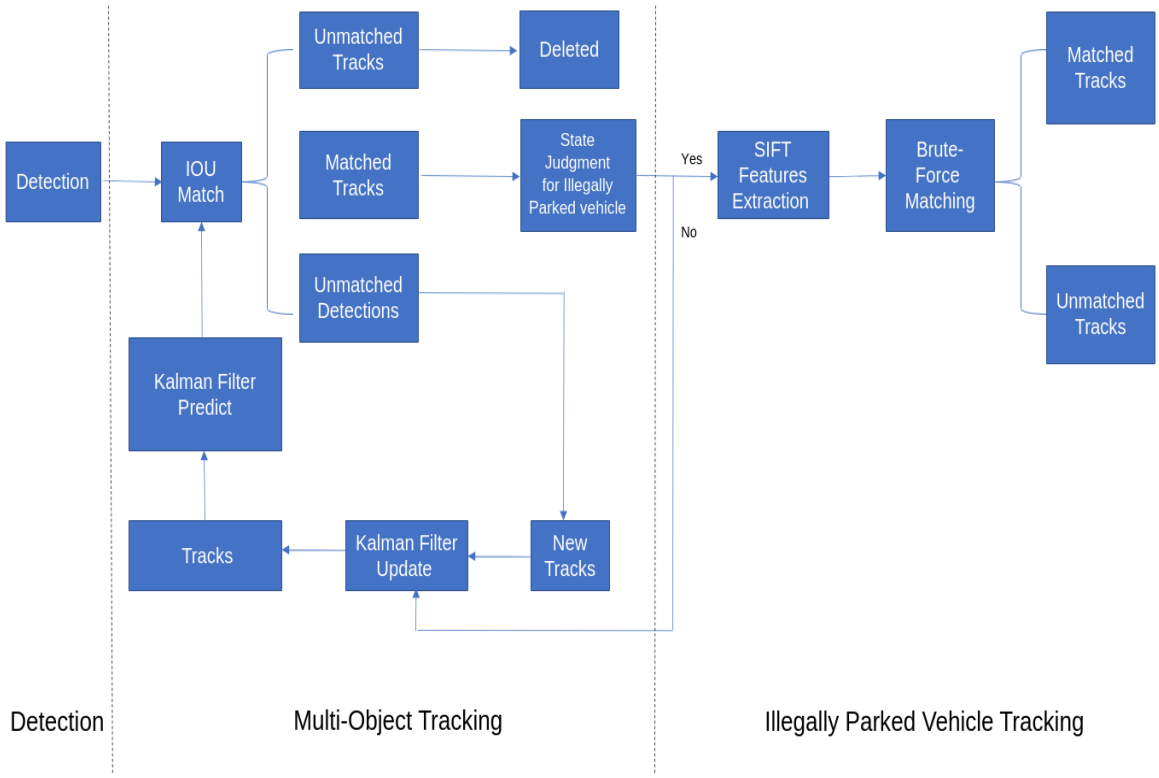

Figure 16 The framework of proposed method in this chapter. The detection module is based on YOLOV3 algorithm. The multi-object tracking includes the SORT algorithm and the illegally parked vehicle judgment module. The last module build a key-points matching based illegally parked vehicle tracking.

The framework is divided into three parts. The first and second parts are still based on the YOLOv3 algorithm and the SORT algorithm as shown in the Figures 9 and 10 (see Chapter 3 for the specific process). Different from the general method summarised by Hassan [104], the proposed method uses the YOLOv3 algorithm to replace the background segmentation algorithm in target detection, which ensures the accuracy of the target classification (background segmentation technology usually needs to integrate other algorithms to identify the category of foreground objects).

Secondly, the deep learning detection algorithm is robust to occlusion, illumination changes and target appearance changes than background segmentation. At the end of the second part, the target state judgment module (see Section3.3) will output two results. Vehicles judged to be driving legally will not be considered in the next step. For a vehicle that is defined as an illegally parked vehicle, starting from the first frame of the event, the proposed method will use the SIFT algorithm to extract the key points of the target, and this frame will also be set as the initial frame. In the next frame, since the position of the stationary vehicle will not change, the target will no longer be predicted and detected by SORT. A new SIFT descriptor extraction was used in the same position as the previous frame.

After the key point features of the target area are obtained in the initial frame and the current frame, a feature matching will be built by using the key points of the two frames. If the number of successfully matched key points is greater than a given threshold *G*, it means that the target remains in place. If the number of key points for successful matching is less than the threshold *G*, it means that the target has left the ROI and therefore cannot obtain enough key points for successful matching.

4.4 SIFT Descriptor Extraction and Vehicle Tracking

The multi-object tracking algorithm SORT introduced in the previous chapter is used to provide initial frame tracking and identification of targets. Then, scan this SORT output to find any illegally parked vehicles that may exist in the scene. All vehicles that were determined to be illegally parked will be recorded and stored ID and location information.

At the same time, the alarm will be generated on the first frame when they illegally parked and this frame will be defined as the initial frame. After the initial frame, the SORT tracking of the target will be replaced by the key point matching method proposed in this chapter. The bounding box of the target will be cropped first, and the key points of the cropped image will be described by SIFT features. The cropped target is shown in Figure 17.

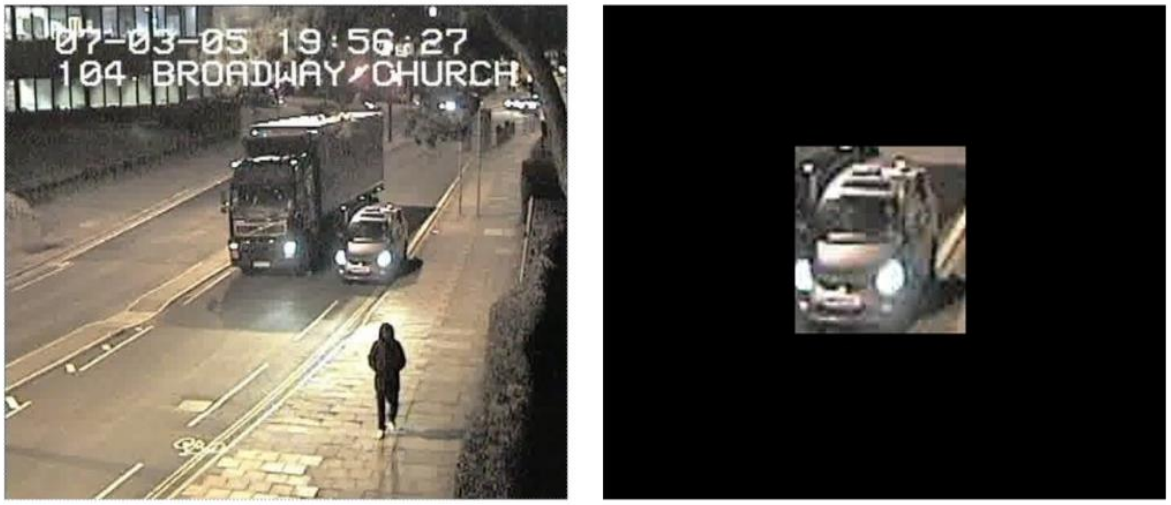

Figure 17 The illegally parked vehicle has been cropped. The cropped image generated by the bounding box of target vehicle. The cropped image usually contains the target, the shadow and the surrounding environment because the bounding box is rectangular.

The Figure 17(b) shows a processing area for feature extraction. In the local feature descriptor, the SIFT algorithm was selected for the description of key points. A review of the SIFT algorithm has been specifically introduced in the literature review. Figure 18 shows an example of the SIFT feature descriptor extraction result and the involved process.

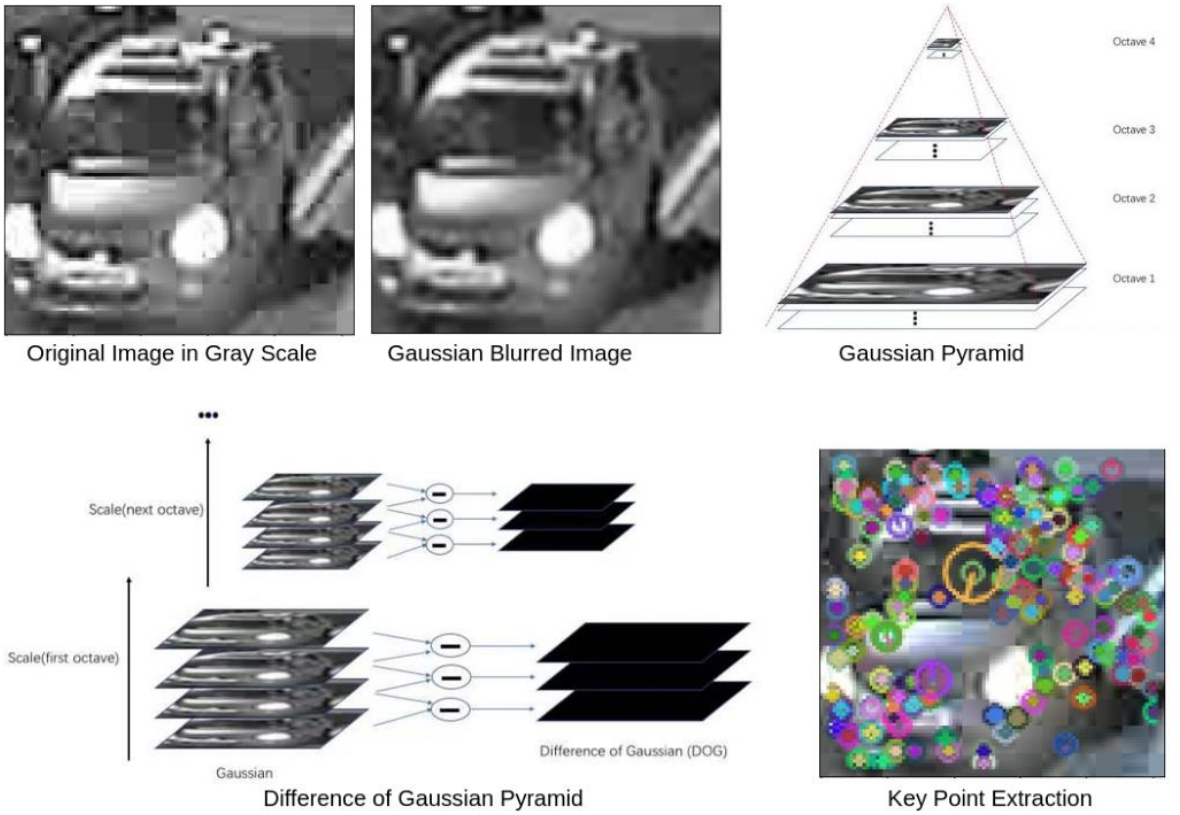

Figure 18 The steps involved in SIFT descriptor extraction. The generation of SIFT descriptor includes the generation of DoG (Difference of Gaussian) pyramid and histogram statistics on the gradient (see section 2.3.2 for detailed operation steps).

The result of applying the SIFT feature descriptor to the key point extraction is that the key points of the target vehicle such as corner points, edge points, bright spots in dark areas and dark points in bright areas are extracted and described by their position, scale and direction. Each key point will have a descriptor, and a set of vectors will describe the key point so that it does not change with various changes, such as changes in lighting, changes in perspective, and so on. These key points form a subset of key point descriptions about the template image (the template image is also called the initial frame). Then in the next

frame, the same feature point extraction and description will be used to build a new key point description subset for the observation image (current frame). Since the feature descriptor is a vector, the distance between the two feature descriptors can reflect the degree of similarity, that is, whether the two key points are the same.

Different matching methods can be used for different feature points. This study used the BF (Brute-Force Matcher) matcher to match the SIFT feature, because according to the test, the BF matcher has better accuracy and speed at the SIFT feature points than the Flannel (Fast Library for Approximate Nearest Neighbours) matcher. Noble's research [105] also proves this point. Brute-Force Matcher was used in this study to calculate the distance between a certain feature point descriptor of the initial frame and all feature point descriptors of the current frame, and then sort the obtained distances. The one with closest distance was taken as the successful matched point.

Although this method is simple, there were a lot of false matches, which required some techniques to filter out false matches. Therefore, the KNN (K-nearest neighbours) algorithm have been used to remove false matches. The specific method is to return the two nearest neighbour matches for the feature points (from the initial frame point set). If the ratio of the first match to the second match is large enough (the vector distance is far enough), it indicates that the first match is a correct match. Since only the tracking of stationary objects is considered, the tracking method based on feature point matching is very suitable for the situation discussed in this chapter. The tracking result can be judged by the number of successfully matched feature points. Figure 19 visualises the matching results.

(a)

(b)

Figure 19(a) A successful matching example (b) All successfully matched feature points are linked between two frames

Since the matched target is a stationary vehicle, the number of successfully matched feature points will not change over time under the ideal condition. However, due to changes in the

environment, the appearance of the vehicle may change. Therefore, in actual operation, the first matching of the vehicle will usually be the best result (successful matching peak) during the entire tracking of the target. Considering that the target may be partially blocked or interfered by environmental noise, the subsequent matching success value is greater than the selected threshold *G* to determine that the target is still being tracked. If the value is less than the threshold *G*, the target is in one of two situations:

1. The target vehicle has left the no parking zone
2. The target vehicle is blocked in a large area or even completely blocked

In both cases, the matching success rate is significantly lower than the peak value of successful matching. This will cause the algorithm to incorrectly classify the occluded situation as the vehicle has left the ROI and thus delete the tracker, or the vehicle does leave the ROI but the tracking is still running. Therefore, a timer has been added in the tracking process. Normally, the occlusion of the vehicle is short-lived, so once the number of successfully matched feature points is less than the threshold *G* and the duration is within the threshold *T*, the tracker will not be deleted. Conversely, once the tracker does not give enough successful matches for a long time (over *T*), it means that the target has left the no parking zone.

Through the statistics of matching results for different size targets, *G*=40 is an appropriate threshold to judge the matching result. The value cannot be smaller because the non-vehicle features in the bounding box will also give a successful matching. And the selection of the threshold *T* also needs to balance the occlusion event and the accuracy of tracking.

Too large a threshold for T will cause the tracking to not be released in time when the target leaves. Therefore, *T*=2 seconds (25fps) is an appropriate threshold.

4.5 Results and Discussion

The performance of the proposed method is evaluated from two aspects: a method based on feature point matching instead of SORT tracking for illegally parked vehicles; and the effectiveness of occluded vehicles. The method proposed in this chapter has been evaluated in different traffic scenarios from the i-LIDS dataset.

The video sequence set used in this chapter comes from the i-LIDS dataset. The set contains 39 illegally parking events and 21 videos of 2-10 minutes in length. As the same with the Chapter 3, the alarm is generated after the vehicle is stationary in the no parking zone for 60 seconds.

The overall test results of the proposed method are shown in Table 2. Among the 48 real events, the proposed method monitored 39 events and reported 1 false positive event. The precision rate of all alarms was 97.5%. At the same time, 7 real events were not found and tracked, and the overall recall was 84.8%.

Video Name	Total Event	True Positive	False Positive	False Negative
PVTRA101a01	2	1	0	0
PVTRA101a02	2	2	0	0
PVTRA101a03	2	1	1	0
PVTRA101a04	2	2	0	0
PVTRA101a05	1	1	0	0
PVTRA101a06	1	0	0	1
PVTRA101a07	1	1	0	0
PVTRA101a08	1	1	0	0
PVTRA101a09	3	1	0	2
PVTRA101a10	5	4	0	1
PVTRA101a11	2	2	0	0
PVTRA101a12	2	2	0	0
PVTRA101a13	2	1	0	1
PVTRA101a14	2	2	0	0
PVTRA101a15	2	2	0	0
PVTRA101a16	3	2	0	1
PVTRA101a17	4	4	0	0
PVTRA101a18	2	2	0	0
PVTRA101a19	3	2	0	1
PVTRA101a20	3	3	0	0
PVTRA101a21	3	3	0	0
Total	**48**	**39**	**1**	**7**

Table 2 Results obtained after testing the proposed method on i-LIDS dataset

According to observations, the proposed method is robust to partial occlusion (Figure 20(a), (b), (d)) and short-term large-area (Figure 20(c)) occlusion. As shown in Figure 20, the tracker can still keep working when the target is blocked by other objects.

In order to clearly show the progress of the proposed method compared to the previous chapter, the method in this chapter is represented by a white bounding box and the tracking generated by the SORT algorithm is represented by a coloured bounding box. When the word ‘alarm’ displayed above the bounding box, it indicates that the illegally parking tracker discussed in the Chapter 3 is also working. Figures 20(a) and (b) show a scene where

multiple vehicles gather and block each other. It can be clearly seen that when vehicles are close to each other, the tracking of illegally parked vehicles based on SORT cannot work. In contrast, the method proposed in this chapter still keeps track even if SORT tracking disappears or the bounding box drifts. The SIFT-based tracking represented by the white bounding box is not affected by partial occlusion. Figure 20(c) and (d) show the tracking results when the target is completely occluded. For short-time (several seconds) large-scale occlusion or complete occlusion of the scene in the figure, the proposed method has good robustness. The number of successfully matched feature points is obviously lower than the given threshold *G*, but the tracker will not be released within the pre-set time T, which ensures that the tracker is continuously positioned at the original position.

Figure 20 The tracking results under occlusion. (a)There are two tracking has been built. (b)The targets have been occluded by another vehicle while tracking. (c)-(d)The target has been nearly completely occluded by the red van but the tracking is very robust. The matched key-points drops to 5 during occlusion.

Although the proposed method is significantly improved over the method in the previous chapter, it still has shortcomings. The false negative event is caused by two reasons. As shown in Figure 21, the first reason for monitoring failure is because the target is far from the camera. The video sequence of i-LIDS was captured by low-quality analog camera with a resolution of 576*720. During the signal transformation from analog to digital, the edges and corners of small targets could become blurred, which causes the SIFT algorithm to be

unable to extract enough key points for feature matching. This situation also appeared in Figure 22. Although the SORT algorithm tracked one of the vans in the image, the SIFT based tracking still cannot be built because the number of feature points is too few for matching.

Figure 21 Even under relatively loose matching conditions, only 30 key points are extracted for feature matching. In this case, the target's matching success rate has been below the threshold G for a long time, causing the tracker to be deleted.

The second cause of false negative events is the failure of target detection. For example, in Figure 22 and 23, the YOLOv3 algorithm successfully identified the car, but the vans cannot be detected due to the occlusion and unclear object display.

Figure 22 Unlike typical vehicles, the van in the video is hard to be identified and classified by YOLOV3.

Figure 23 Through manual inspection, it was found that YOLOv3 detected one of the vans, but the detection only lasted 1 frame (the video sequence was running at 25 frames per second).

4.6 Conclusion

This chapter discusses the tracking technology based on feature point matching. The main focus is to determine whether the feature of the key point is strong enough for the matching

of illegally parked vehicles in different frames with real situations. The i-LIDS dataset can be used for this purpose, because this dataset provides a large number of real scenes used to track illegally parked vehicles in the no parking area. This method consists of two stages. First, YOLOv3 based target detection and SORT multi-object tracking methods are used to obtain the positions and target identities of all vehicles in the scene. These tracked targets will be recorded as soon as they become stationary in the no parking zone. Once they are identified as illegally parked targets, SIFT descriptors can be used to obtain their key features and store them in the database for further tracking. In the second stage, BF matching based tracking technology is used to process these feature data. By matching the feature points of the target in the current frame with the feature information stored in the database, an inter-frame connection can be established for each illegally parked target. The performance of the proposed method is evaluated from two aspects.

First of all, the target detection stage is evaluated along with SIFT feature description, the overall prediction rate of the system can be obtained to 97.5%. It benefits from the accuracy of the deep learning algorithm YOLOv3 in the object detection, and also benefits from the system made an object classifier to filter all none-vehicle objects, the system avoids as much as possible the false positive rate.

In the second stage of the evaluation, the tracking technology based on key point matching was analysed. It can be seen that compared to the method proposed in the previous chapter, the system has achieved a greater improvement. In a total of 48 real events, the overall recall rate is 84.8% compared to the 49% reported in the previous chapter. The main reason for the improvement of this recall rate is that SIFT features can establish the target's inter-

frame link more firmly than the data association method. On the other hand, the detection rate of 84.8% still cannot meet the monitoring requirements of illegally parked targets in real scenarios. The main reason which led to 7 failed monitoring in a total of 48 events was that the target was often occluded by other objects in the scene. In a real traffic environment, occlusion can happen in many ways. One is the short-term occlusion from fast moving objects, and the second is the long-term and large-area occlusion from other slow moving or stationary vehicles. Compared with the previous chapter, this system is well adapted to short-term occlusion regardless of whether it is completely occluded. In addition, tracking can also be maintained for a small part of long-term occlusion.

Therefore, it is concluded that in order to detect and track illegally parked vehicles in challenging traffic scenarios, the system needs to be continuously improved. This not only requires full adaptation to all occluded conditions, but also can track targets under drastically changing lighting conditions. In the next chapter, an improved feature point matching technology will be discussed, which can effectively track occluded and blurred targets, and Chapter 6 will explore whether the improved feature point matching method can track targets stably under changing lighting conditions.

Chapter 5 Tracking with Dense Feature Points

5.1 Introduction

The conclusion of the previous chapter is that although the matching of the SIFT feature descriptor can replace the multi-object tracking technology, its performance will be affected when the target is occluded for a long time or by a large vehicle. This situation usually occurs in a heavy traffic environment, and in the blurry scene, the feature point extraction of the SIFT algorithm will be more difficult. In the case that the feature points are collected sparsely, the only way can be used is loosing the matching conditions to obtain successful matching, which will reduce the performance of the system.

In order to develop a more powerful and accurate parking vehicle monitoring system that can be implemented in all-weather environments and traffic scenarios, this chapter introduces the feature point extraction method based on dense SIFT to help extract a more reliable set of key points for the target tracking. Then a background matching has been used to detect whether the target has left the no parking zone.

5.2 Chapter Organization

The rest of this chapter is arranged as follows: Section 5.3 introduces the feature point extraction method based on the dense SIFT. Section 5.4 shows the tracking under occlusion conditions and gives some examples. The release of tracking is discussed in Section 5.5. The

test results on the dataset are shown in Section 5.6. Finally, a summary of this method is given in Section 5.7.

5.3 Dense Key Point Extraction

As shown in the previous chapter, the entire system is divided into two main stages: detection and tracking. The first stage is to accurately detect illegally parked vehicles in the scene. Through a large number of tests, the target detection based on YOLOv3 and SORT algorithm is considered to be reliable. At the beginning of illegal parking, at least one frame of the detection result is given containing the identity and location of the target. The manually-divided no parking zone and target position and speed measurement all play an important role in this stage. More "no parking zone" are shown in Figure 24. Using the frame containing the detected illegally parked vehicle which generated in the first stage, the feature point matching can keep track in the second stage until the target leaves the no parking zone.

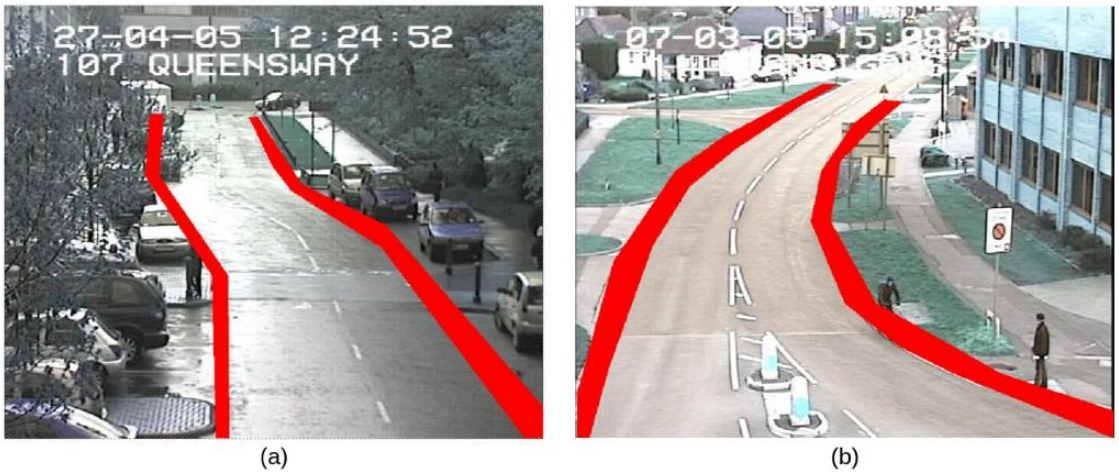

Figure 24 i-LIDS dataset for Illegally Parked Vehicle Detection. The no parking zone are marked in red.

However, as analysed in the previous chapter, the SIFT algorithm has defects in matching certain targets or under special conditions. In order to build feature descriptor subsets for targets with small areas and blurred targets, more feature points must be extracted. Even for targets whose features can be extracted normally, more key points mean that more accurate tracking can be established and can be adapted to more complex environments. However, this requirement is limited by the fact that the SIFT algorithm automatically selects the extreme points in the image pyramid when extracting feature points.

Therefore, the dense SIFT method has been introduced to extract key points. This technology is a dense version of SIFT, which can quickly calculate the key points of the sampling at a pre-set density. The main idea is shown in Figures 25.

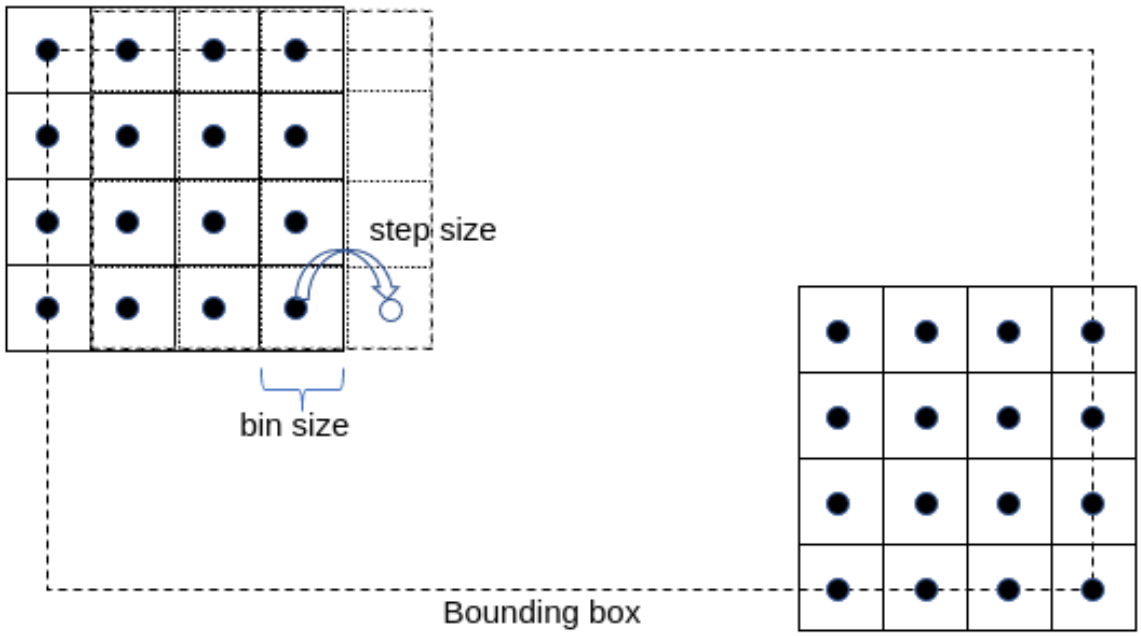

Figure 25 A patch has been used to slide on the image with a certain step size. This patch describes the sub-sampling area, and the patch size is 4bins*4bins

Dense SIFT effectively increases the number of feature points that can be used for matching. As shown in Figure 25, the 'step size' represent the density of extracted feature points. Feature points will be extracted according to the pre-set step size. When the step size is 1, all pixels will be extracted as feature points. The extracted feature points will be described in the same method as the original SIFT.

Through a fixed step size, dense SIFT can be reused for multiple images of the same size. However, for the application scenario in this thesis, dense SIFT needs to select different step sizes to adapt to different sizes of illegally parked vehicles. Figures 26 and 27 show a comparison of a series of target vehicles with key points extracted under the dense sift algorithm and the same target using SIFT.

Figure 26 Images shows the extraction scene of feature points. The tracked target is highlighted. (a) The target is in the night scene (b) The obscured fuzzy target (c) The fully displayed target is in the daytime scene

Figure 27 shows the key point extraction results of the above three targets. These three examples show that the dense SIFT method can achieve better key point extraction results than the SIFT algorithm in different sizes. Considering that a large number of key points will increase the calculation time during the matching process, the step size has been adaptively selected according to the size of the target to achieve a balance between the calculation speed and the matching result. During the operation of the system in this chapter, the size of the bounding box is regarded as the size of the illegally parked vehicle. By calculating the pixel size of the bounding box, the step size will be adaptively adjusted according to the size to maintain an appropriate number of key points.

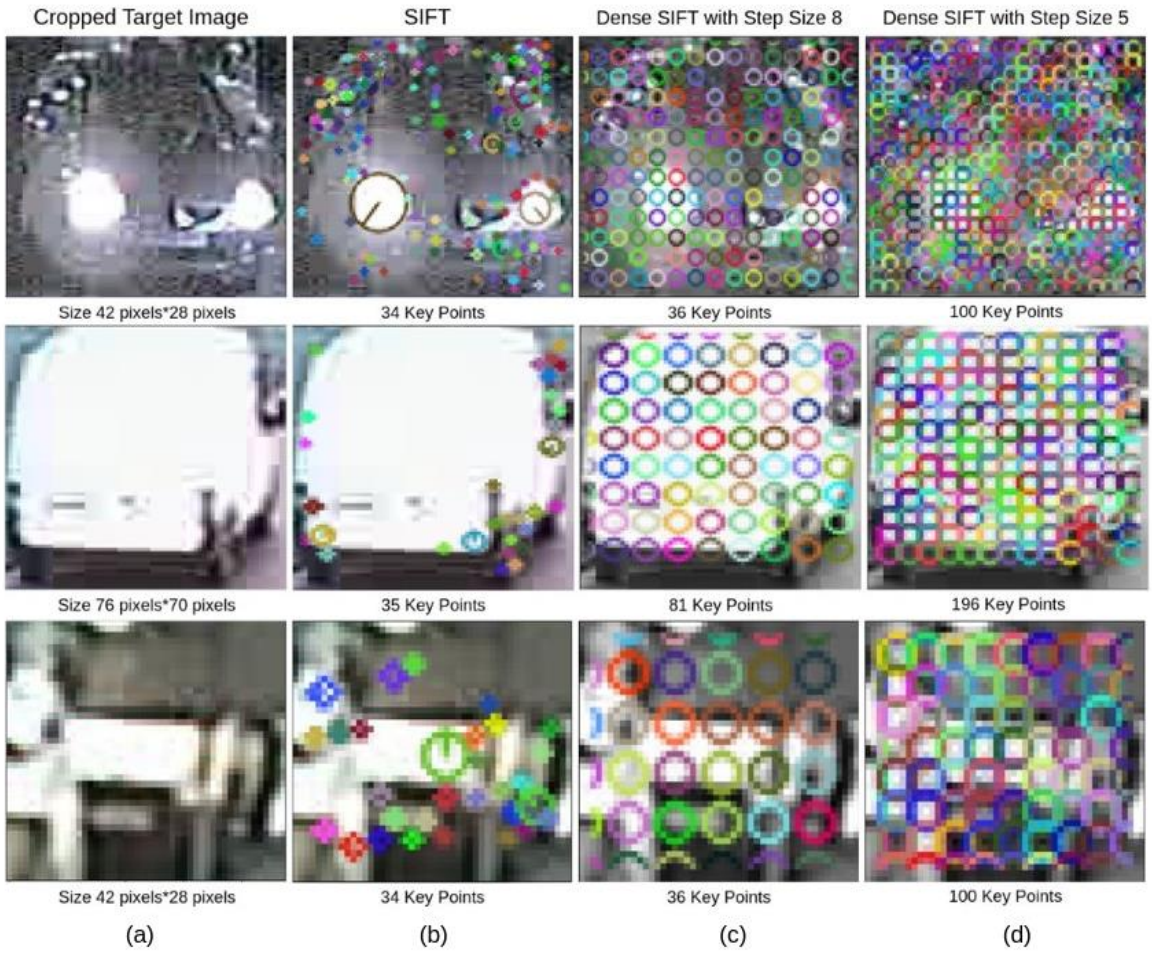

Figure 27 For the same target, there is a huge difference between the original SIFT and Dense SIFT when extracting feature points. (a) Pixel size of target (b) SIFT output (c) Dense SIFT output with step size 8 (d) Dense SIFT output with step size 5

5.4 Illegally Parked Vehicle Tracking Under Occlusion

The dense SIFT method changes the key point locations on the target and obtains more feature points for the matching to track. As the number of feature points available for matching increases, the performance should improve. Tracking occluded targets is the main challenge faced in the previous chapters. Many tracking methods, as in this thesis, need to

be robust against occluded objects to prove their performance. For example, Tian et al. [106] stated that occlusions are clearly a problem led to some low-correlation scores. The method proposed in this chapter attempts to overcome this problem. First of all, as discussion in the previous chapter, occlusion problems can be divided into the following categories, which can be analysed separately in this section:

1. Partial occlusion
2. Complete occlusion

Examples of category 1 have been shown in the images in Chapter 4 (Figure 20). The SIFT-based method in the previous chapter can be adapted to short-term or long-term partial occlusion, because the unoccluded part can provide feature points to match. The tracking can still work as long as these features can establish matching on the unoccluded part of vehicle. However, assuming that the light changes when the target is partially occluded or the target has few feature points due to blur, the matching feature points are likely to be less than the threshold G (defined in Chapter 4) and the tracking will fail. The method proposed in this chapter overcomes this problem. For illegally parked vehicles, the feature points are no longer autonomously selected by the SIFT algorithm, but intensively extracted by adaptively selected steps. Therefore, for the detected target, this method provides enough feature points when the target is partially occluded or blurred. Figure 28-29 shows the difference between Dense SIFT and SIFT when matching features of partially occluded targets.

Figure 28 Typical occlusion scene in the i-LIDS database, including some short-term partial occlusion.

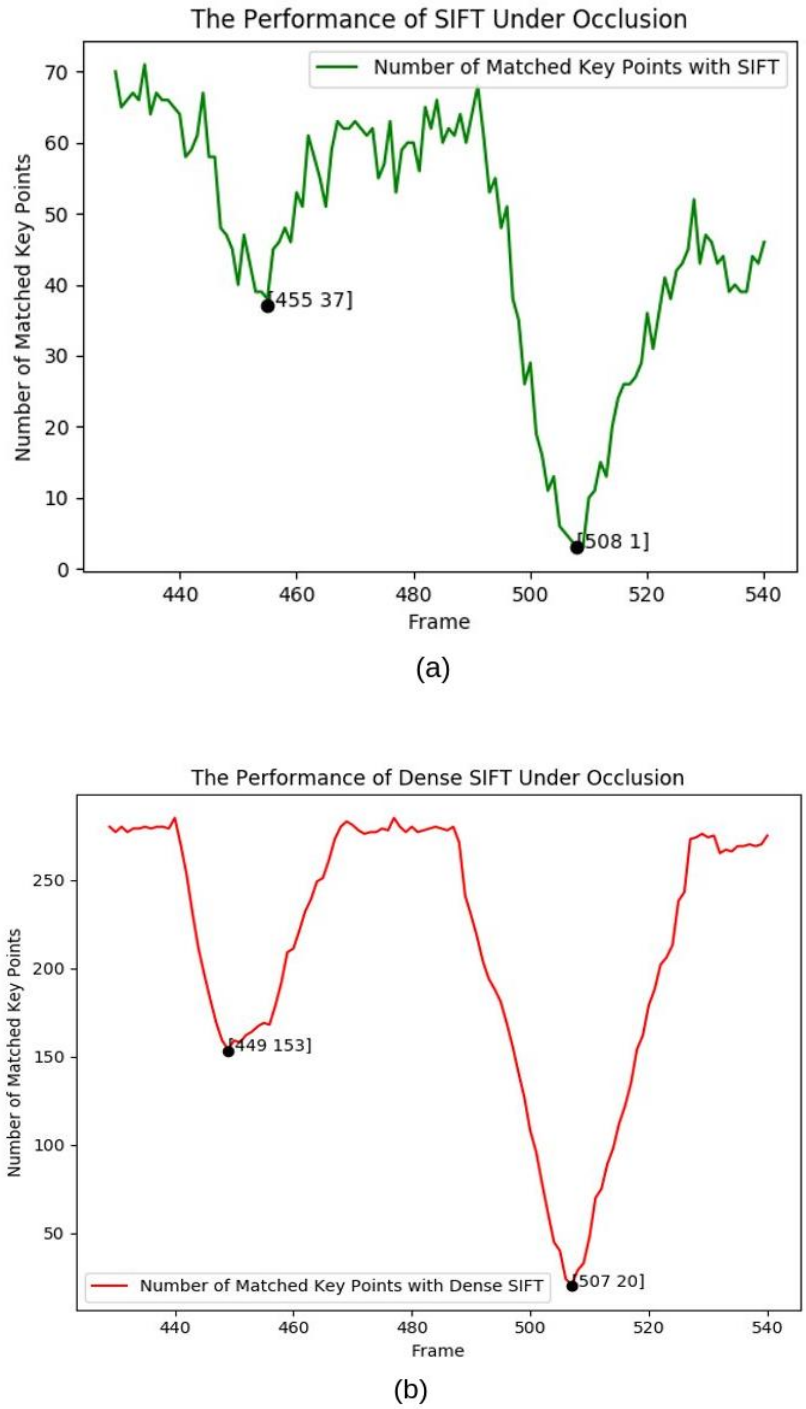

Figure 29 The continuous change in the number of matched key points from frame 429 to frame 540. (a) The key point is extracted from the SIFT algorithm in Chapter 4, and the number of matched key points reaches the minimum value of 1 at frame 507. (b) The key point extraction is taken from the method proposed in this chapter

Compared to the SIFT algorithm, the method in this chapter significantly increases the number of matching feature points, especially when occlusion occurs. For occlusions like the one shown in frame 455 (Figure 28), the feature points that dense SIFT can extract will not be reduced by the occlusion. More importantly, the occlusion that occurred at frame 508 almost completely covered the target vehicle. In this case, there is still a considerable number of matched points (change from 1 to 20) that can be obtained to ensure accurate tracking.

The bigger challenge is the second type of occlusion, that is, complete occlusion. Hassan set a timer in [104] to keep track of the completely obscured target within P seconds. When the target is occluded by another stationary object, the algorithm will delete the previous target and track the new stationary target. In the proposed system, the timer has been abandoned to deal with occlusion. This is because setting the waiting time also means that even if the target leaves the ROI, the tracker needs to keep tracking in place within the waiting time. Therefore, a step was needed to deal with the situation where sufficient key points cannot be extracted when the target is completely occluded. Considering that the large-area occlusion and the appearance of the target leaving the ROI are the same at the level of system detection, it means that the number of matching feature points is less than the threshold G (the threshold has been set to less than the minimum number of matches in the case of partial occlusion).

Therefore, a step was set up to verify whether the target is completely occluded. The intersection of the bounding boxes has been used to determine whether the vehicle is occluded. Assume that a scene with occlusion is shown in Figure 30. The rectangle

represents three cars. There are many ways to determine whether the rectangle (bounding box) intersects, for example, to determine whether any two sides of the rectangle intersect (referring to occlusion). However, this method has a drawback. When a rectangle is contained by another rectangle (when it is completely occluded), no edges are intersected but still meet the definition of intersection. For example, vehicles 1 and 2 are shown in the figure below.

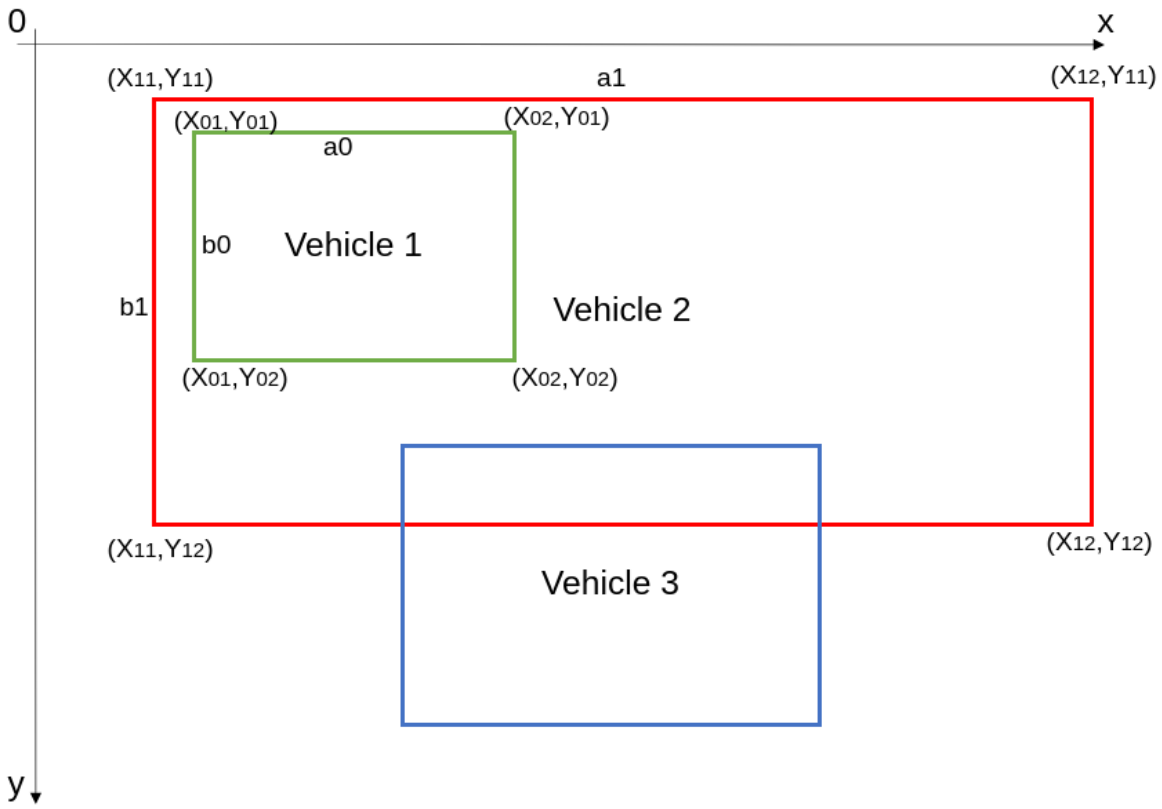

Figure 30 The bounding box intersection in different situations. The three rectangles respectively represent the vehicles in the scene. Two different occlusions are shown, namely the complete occlusion between vehicle 1 and 2 and the partial occlusion between vehicle 2 and 3. In general, if the distance between the centre points of the two targets is greater than the sum of their length and width respectively, which means the occlusion has occurred.

Another method has been used to detect intersections, which is to compare the distance between the centres of the two rectangles in the x-axis direction and half the sum of the lengths of the two rectangles. At the same time, comparing the distance between the centres of the two rectangles in the y-axis direction and half the sum of the widths of the two rectangles. If the distance of the centre on the x-axis and y-axis is less than half of the sum of their side lengths, the condition of intersection is met. Formulated as follows:

$$|x_{01}-x_{02}-x_{11}-x_{12}| < (x_{02}-x_{01}) + (x_{12}-x_{11}) \qquad Equation\ 5-1$$

And

$$|y_{01}-y_{02}-y_{11}-y_{12}| < (y_{02}-y_{01}) + (y_{12}-y_{11}) \qquad Equation\ 5-2$$

If the above conditions are met at the same time, the area of the intersection area can be calculated:

$$intersection = [min(x_{02}, x_{12}) - max(x_{01}, x_{11})] \times [min(y_{02}, y_{12}) - max(y_{01}, y_{11})]\ Equation\ 5-3$$

$$IOU = \frac{intersection}{target\ area} \qquad Equation\ 5-4$$

Since the bounding box coordinates of the illegally parked vehicles are fixed and clear, the Intersection over Union (IOU) of the intersection area can be calculated by using the

bounding boxes of other objects and the illegally parked vehicles for intersection detection (Equation 5-4). Considering that the irregular shape of the vehicle causes the bounding box to contain some non-vehicle pixels, the IOU was set to 80% by counting different size of occlusion.

If the IOU is less than 80%, it means that the target is partially occluded, but also means that there are still enough feature points to be extracted. On the contrary, it means that the target is completely occluded, and the target's tracker will be frozen in place until the occluder moves to expose the target. Figure 31 and 32 show the effect of occlusion detection. The blocked part is marked by blue.

Figure 31 Partial occlusion will not affect the collection of feature points. The intersection area will be marked as occluded and covered by a blue mask

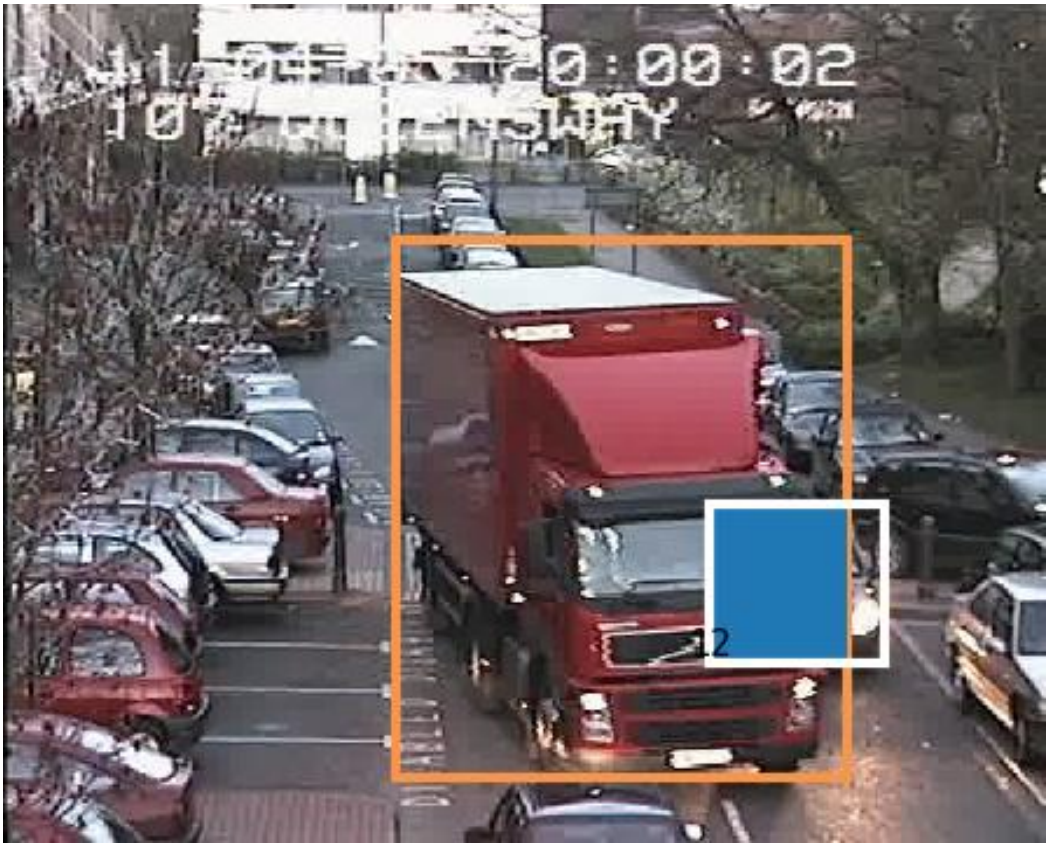

Figure 32 It is a common situation in tracking that illegally parked vehicles are almost completely occluded. The intersection has been detected and marked by blue.

5.5 Release the Tracker

If the matched feature point of the tracked target is less than the threshold *G*, the system will trigger the detection of occlusion. If an occlusion occurs, the tracker will not be deleted even if the matched feature points are too few. During the duration of the occlusion, the tracker will always remain in the original position of the target.

At the same time, considering that the SORT algorithm fails to track some targets, in some cases, the bounding box of the occluder cannot be given, as shown in the figure below.

Figure 33 Because the SORT algorithm fails to track the bus, it is impossible to detect the occlusion of the illegally parked vehicle. When the bus completely covers the car, the number of successfully matched feature points will approach 0, which will cause the tracker to be deleted

Therefore, a background matching was introduced to keep the system from deleting tracking when there were unrecognised occlusions in the scene.

If an image without any vehicle is defined as a background frame, it can be found that when the vehicle leaves the ROI, the no parking area and the background frame are almost the same (there may be shadows and different lighting). On the contrary, when there is an occlusion in the no parking area, the background frame and the area will not give any matched feature points.

Hence, a step of matching the target area with the background frame has been added to the system. This step will be triggered when the following two conditions are met at the same time: the number of matched feature points are less than the threshold G, and no occlusion is identified around the target. In this step, feature points will be extracted and matched for the background frame and the bounding box of the current frame. If the matching result is much higher than the matching result of the initial frame and the current frame, it means that the bounding box area of the current frame is more similar to the background frame (road surface). If it is not, which means that other objects are blocking the target. Figure 34 shows a typical case of distinguishing whether a vehicle has left or is blocked.

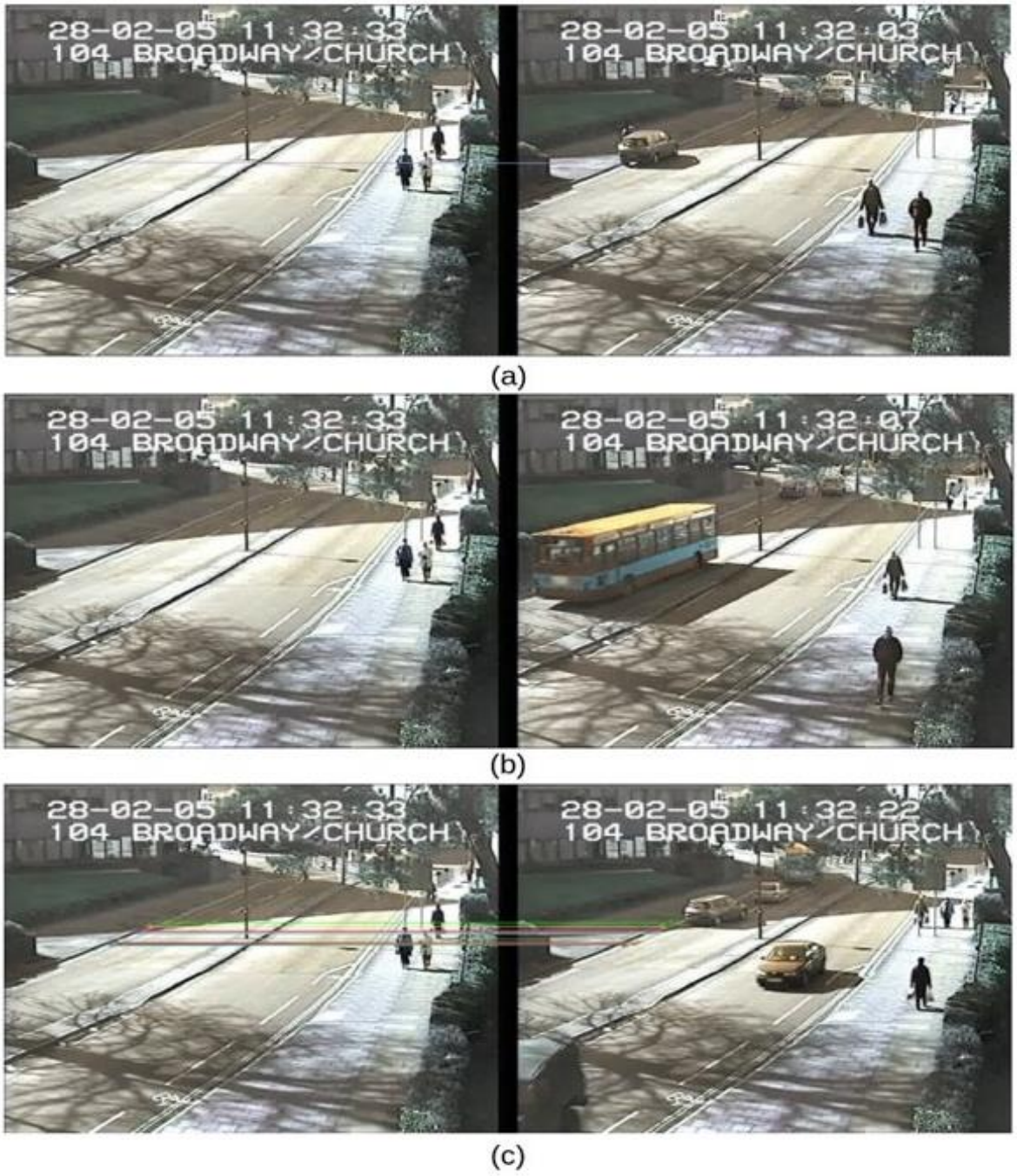

Figure 34 The matched feature points are represented by lines. All matches occur at the same position (target bounding box area) in different frames (a) Suppose that the bounding box area of the vehicle is matched with the same position in the background frame. Only one successful match occurs because the bounding box usually contains some non-vehicle parts (b) When the vehicle is completely occluded, it cannot match any similar feature points. (C) After the vehicle leaves the original position, the same road information can establish multiple successful matches

This step makes up for the inability to detect whether there is occlusion due to the defect of the SORT algorithm in target tracking, and also helps the monitoring system to achieve real-time tracker release. Compared with other methods that use a timer to deal with target occlusion, this system is more suitable for occluded target tracking. For example, for those occlusion events that are terminated by a timer, the proposed method can maintain long-term tracking without releasing the target because the target cannot be detected. The above steps have played an important role in the target tracking process. The figure below shows the operating structure of the system.

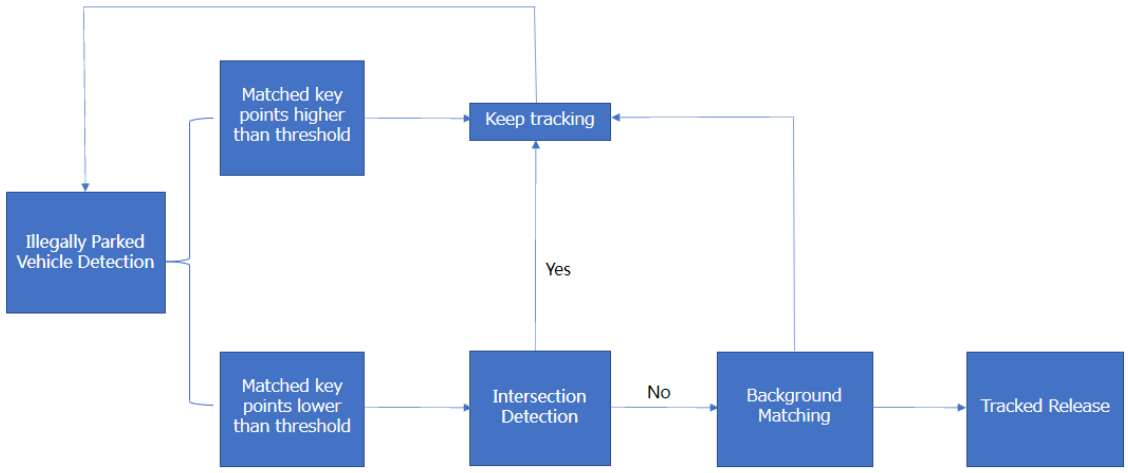

Figure 35 The framework proposed in this chapter for the tracking of illegally parked vehicles under occlusion. The tracking of the target will not be cancelled due to occlusion. Tracking will only be released when the target leaves the no parking zone, and this situation can be detected by the background matching function.

5.6 Results and Discussion

The proposed technique has been tested on three different databases. In the scenario of the i-LIDS dataset, the alarm is triggered 60 seconds after the event occurs. In other dataset

scenarios, the alarm is generated immediately when the vehicle is illegally parked. Considering that the step size of key point extraction with each target are different, the threshold *G* for the lowest match is chosen to use 10% of the maximum number of matches instead of a fixed value. This adaptive threshold *G* helps the system to accurately determine the occlusion status of the target, and it can also delete invalid trackers in time to achieve accurate event monitoring. This system demonstrates excellent performance under the frequent occurrence of various occlusions and any camera angle. For scenes with blurred targets, the proposed method in this chapter had a better performance, because compared with the original SIFT, the proposed method can extract more feature points on the blurred target for matching. The following three datasets have different camera angles, image quality, and lighting conditions.

5.6.1 The I-LIDS Dataset

The proposed method was tested on the i-LIDS dataset. The same video sequence was also used by the method discussed in the Chapter 3. These video sequences have a total duration of more than 6 hours and contain three different scenes at different times of the day, which contains very complex environmental conditions. For example, image movement caused by camera shake, overexposure caused by thunder and lightning, and target blurring caused by low quality analogue camera. Table 3 shows the results obtained from the video sequence. Among 105 real illegally parked events, the proposed method successfully detected 100 events, while the method used in Chapter 3 only detected 51 events. All

alarms were generated within 10 seconds after the target has been marked as illegally parked vehicle. The method achieves a 95% recall, but there were 17 false alarms, accounting for 14% of the total number of alarms, this was due to the object detection error in the YOLOv3 algorithm. Some non-vehicle targets (such as tree and shrub) have been identified as vehicles and tracked by the SORT algorithm. These objects have been classified as illegal vehicles by the system because they located around the ROI. Therefore, a technology is needed that can screen out non-vehicle targets without consuming a lot of computing power. The specific method has been discussed in the next chapter.

In addition, four parked vehicle scenarios from the i-LIDS dataset were selected as the benchmark dataset for the AVSS (Advanced Video and Signal based Surveillance) conference. Table 4 shows the test results of this method on these four scenarios and compares it with other methods. The tracking images have been shown in Figure 36. The purpose of this test is to demonstrate the sensitivity of the proposed method to illegally parked events. Generally, faster response speed and timely tracking release can prove that the tracking system has better performance. The average error can be calculated by comparing the start and end time of the tracking with the ground truth, and the best tracking result is obtained compared with other results. Compared with other methods, the proposed method has the closest tracking result to ground truth

Video Name	Duration (hh:mm:ss)	Total Event	True Positive	False Positive	False Negative
PVTEA101a	01:21:29	26	25	0	1
PVTEA101b	00:24:40	8	8	0	0
PVTEA102a	00:51:56	18	18	0	0
PVTEA103a	01:03:54	16	15	0	1
PVTEA201a	00:18:50	4	3	0	1
PVTEA202b	01:05:23	17	17	17	0
PVTEA301a	00:15:29	3	2	0	1
PVTEA301b	00:28:07	7	6	0	1
PVTEA301c	00:27:09	6	6	0	0
Total	**06:16:57**	**105**	**100**	**17**	**5**

Table 3 Test results on the i-LIDS dataset. Compared with the test results in Chapter 3, F1 has increased from 0.64 to 0.90 but there are still 17 false positive events which are due to the misdetection of the YOLOV3 algorithm.

Method	Sequence	Ground Truth		Duration (sec)	Obtained Results		Duration (sec)	Error (sec)	Average Error (sec)
		Start Time	End Time		Start Time	End Time			
Proposed Method	Easy	02:48	03:15	00:27	02:47	03:18	00:31	4	4.25
	Medium	01:28	01:47	00:19	01:36	01:48	00:12	9	
	Hard	02:12	02:33	00:21	02:14	02:34	00:22	3	
	Night	03:25	03:40	00:15	03:25	03:41	00:16	1	
Bevilacqu et al. [72]	Easy	02:48	03:15	00:27	N/A	N/A	00:31	4	4.33
	Medium	01:28	01:47	00:27	N/A	N/A	00:24	5	
	Hard	02:12	02:33	00:21	N/A	N/A	00:25	4	

	Night*	03:25	03:40	00:15	N/A	N/A	N/A	-	
Boragno et al. [107]	Easy	02:48	03:15	00:27	02:48	03:19	00:31	4	5.25
	Medium	01:28	01:47	00:19	01:28	01:55	00:27	8	
	Hard	02:12	02:33	00:21	02:12	02:36	00:24	3	
	Night	03:25	03:40	00:15	03:27	03:46	00:19	6	
Lee et at. [71]	Easy	02:48	03:15	00:27	02:51	03:18	00:27	6	6.25
	Medium	01:28	01:47	00:19	01:33	01:52	00:19	10	
	Hard	02:12	02:33	00:21	02:16	02:34	00:18	5	
	Night	03:25	03:40	00:15	03:25	03:36	00:11	4	
Guler et al. [108]	Easy	02:48	03:15	00:27	02:46	03:18	00:32	5	6.75
	Medium	01:28	01:47	00:19	01:28	01:54	00:26	7	
	Hard	02:12	02:33	00:21	02:13	02:36	00:23	4	
	Night	03:25	03:40	00:15	03:28	03:48	00:20	11	
Venetiane r et al. [109]	Easy	02:48	03:15	00:27	02:52	03:16	00:24	5	9.33
	Medium	01:28	01:47	00:19	01:43	01:47	00:04	15	
	Hard	02:12	02:33	00:21	02:19	02:34	00:15	8	
	Night*	03:25	03:40	00:15	03:34	N/A	N/A	-	
Porikli [110]	Easy*	02:48	03:15	00:27	N/A	N/A	N/A	-	11
	Medium	01:28	01:47	00:19	01:39	01:47	00:08	11	
	Hard*	02:12	02:33	00:21	N/A	N/A	N/A	-	
	Night*	03:25	03:40	00:15	N/A	N/A	N/A	-	
Lee et at. [111]	Easy	02:48	03:15	00:27	02:52	03:19	00:27	8	12.33
	Medium	01:28	01:47	00:19	01:41	01:55	00:15	21	
	Hard	02:12	02:33	00:21	02:08	02:08	00:29	8	
	Night*	03:25	03:40	00:15	N/A	N/A	N/A	-	

Table 4 The results of the proposed method on the AVSS 2007 benchmark database. Compared with other methods that are also tested on the AVSS 2007 data set, the proposed method has the best performance, which is reflected in the smallest error time. The average error of 4.25 seconds is the best result currently available. Easy, Medium, Hard and Night refer to test videos in four different scenes and environments, and the difficulty of tracking increases sequentially. Figure 36 shows four video tracking examples. *: This item is not included in the average error calculation. N/A: The data is not available.

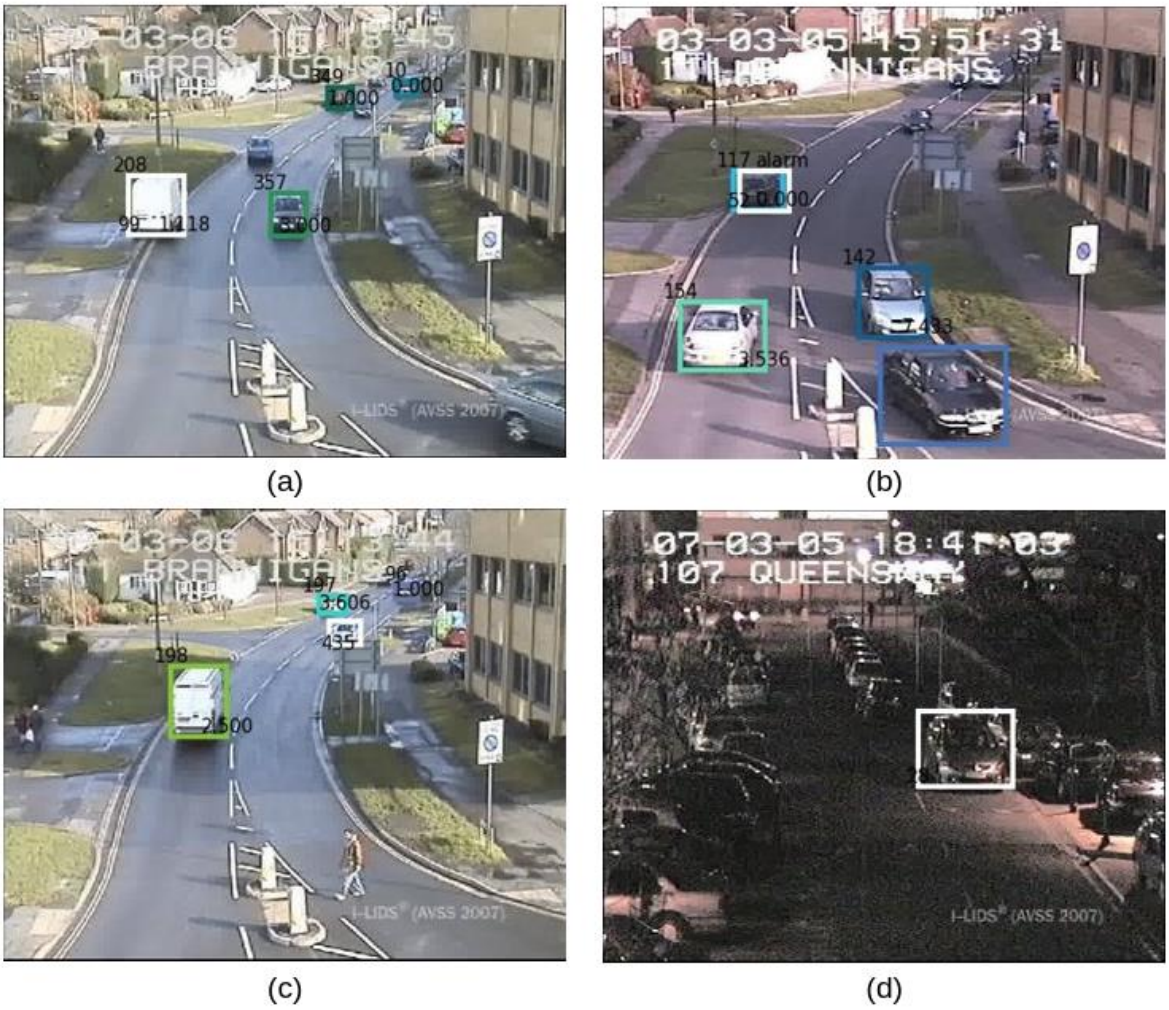

Figure 36 Shows the results of the proposed method on the AVSS2007 benchmark dataset. The dataset from i-LIDS contains shadow changes, blurred targets and night targets. (a) PV Easy (b) PV Medium (c) PV Hard (d) PV Night

5.6.2 Sherbrooke Video

Sherbrooke's video is used to study the problem of target tracking in the city which is provided by Jodoin et al. [112]. Although the video was not specifically shot for the detection of illegal parking, the illegal parking events in the video can still be used for testing. In the video, an illegal parking event occurred with 5 partial occlusions and 1 complete

occlusion. The proposed method successfully monitored the event, and no false alarm was generated. This video provides tests under different camera angles from i-LIDS. After testing, it is found that the same thresholds and parameters as i-LIDS can be used in this test. There was only one true positive alarm generated in testing and no false alarm generated (100% recall and 100% precision). Figure 37 shows the tracked scene and the tracking results under occlusion. The tracked vehicle is highlighted by a white bounding box.

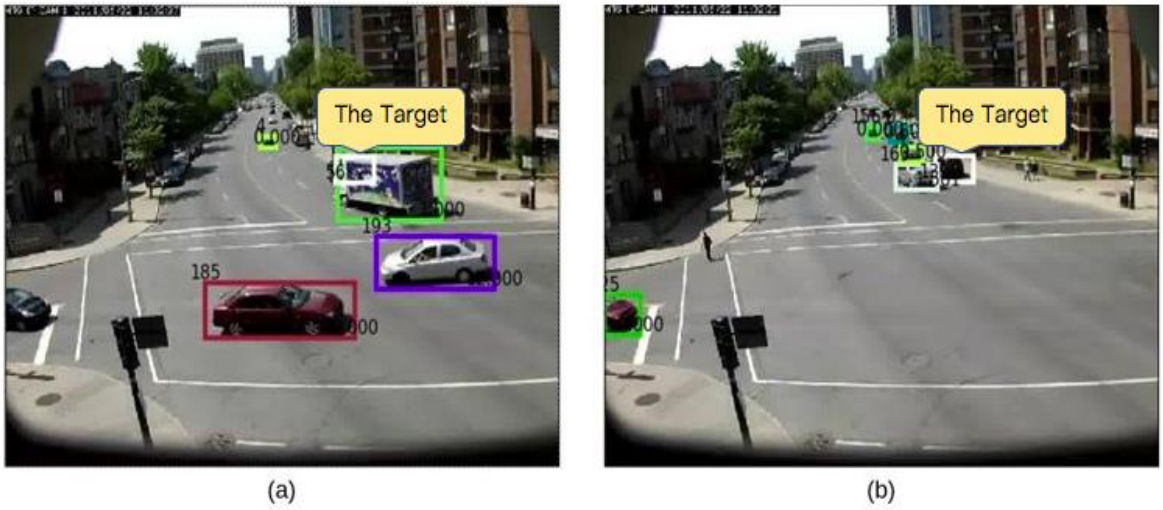

Figure 37 The image shows the tracking scene of illegally parked vehicles from Sherbrooke video. The video contains 6 occlusion events and the target is very small and blurry.

5.6.3 Sussex Daytime Video Dataset

This method was also tested on the Sussex daytime video. This video was produced by Industrial Informatics and Signal Processing Research Group from the University of Sussex to track illegally parked vehicles. The total length of the video is 33 minutes and focuses on a busy highway with a roadside parking area. As a test, the parking area is defined as the

illegal parking area in this test. In the video, there are a total of 6 illegally parking events generated and accompanied by thousands of occlusions. During the testing on this video, all illegally parked vehicles have been tracked and both recall and precision reached 100%. Figure 38 shows an image from Sussex daytime video, where two tracked events are highlighted by a white bounding box.

Figure 38 The images from the Sussex daytime video. All events are monitored and no false events are generated.

5.7 Conclusion

This chapter introduces a powerful method for tracking illegally parked vehicles, which solves one of the biggest challenges faced by this application, which is occlusion. Firstly, an extraction method based on adaptive selection of key points has been used to establish the feature description of the target, in which dense extraction was used to overcome the difficulties caused by blurred targets or occluded targets. Once the object has a large

number of feature points extracted, the brute-force matching method was still used to track the target. When the matching result of the target is significantly lower than the normal level, the method based on intersection detection will be used to determine whether the target has left the no parking zone or is blocked by another object. Research shows that the proposed method has a 95% recall under various occlusion conditions.

The proposed method was tested on three datasets. A total of more than 7 hours of video recording traffic scenes in different locations and different environments were used to evaluate the performance of the method. Among 116 real events in all video sequences, the proposed method gave 111 alarms, and the obtained recall rate exceeded 95.6%, compared to 49% in the Chapter 3. At the same time, compared with the methods proposed in the previous chapters, this method produced 17 false alarms with a precision rate of 87%, compared with the 97% in Chapter 4 and 94% in Chapter 3. Since the method in this chapter does not include the recognition function of the detected target, tracking of non-vehicle targets in the no parking zone cannot be deleted. Another important performance that can be obtained through experimentation is that as long as the tracking starts, it will not be lost until it leaves the no parking zone. Compared with other methods, the proposed method does not generate a missed track. For example, the method of [104] produced 17 missed tracks on the same i-LIDS dataset.

The methods discussed in this chapter and the previous two chapters are based on the target detection and classification results of the YOLOv3 algorithm. Therefore, in order to reduce false alarms caused by classification errors, new methods need to be added to improve the methods in this chapter. The next chapter highlights a method of screening

non-vehicle objects. Background subtraction technology will be used and designed to reduce computational cost to improve image processing speed and tracking effect under lighting conditions.

Chapter 6 The Illegally Parked Vehicles Tracking with Actively Selected Feature Points and Performance Evaluation

6.1 Introduction

In Chapter 2, Gaussian Mixture Model (GMM) technology is introduced as an object detection method in traditional target tracking methods, and then methods such as SVM are used to classify the detected objects. Nowadays, deep learning-based object detection technology can complete the detection and classification of objects in one step, which has led to less and less relevant studies using GMM and other background subtraction technologies for target tracking. However, compared with the bounding box generated by the deep learning algorithm, the target generated by the background subtraction is more accurate. Although the shape and size of each target are completely different, the bounding box will always contain many non-vehicle parts while marking the target area. Therefore, when dense key points are extracted in the bounding box area, these key points not only describe the features of the vehicle but also the features of the non-vehicle, as shown in Chapter 5.

If the extraction of key points can be focused on the vehicle, it can firstly reduce the time required for feature point matching while ensuring accuracy, and secondly, it can help the system better adapt to tracking under conditions of severe lighting changes. This chapter introduces a novel actively selected feature points (ASFP) method that can accurately extract key points from the tracked object. Its main purpose is to use fewer key points to

accurately track the object and thereby reduce computational costs and adapt to lighting conditions. The key points will still be densely extracted but the position will be restricted to the target, instead of densely collected key points in the bounding box. The experimental results show that this method not only can better adapt to the environment of light changes than the previous method, but also has a significant increase in speed.

6.2 Chapter Organisation

This chapter is arranged in the following way. The ASFP method is analysed and discussed in Section 6.3. This section also highlights the differences between different background subtraction programs. Section 6.4 discusses the application of the ASFP method under changing lighting conditions, and gives examples and comparisons with other methods under the same lighting conditions. Section 6.5 describes how the proposed method response to the false positive events. Section 6.6 shows the test results of the proposed method on the dataset and gives a comparison between the method finally proposed in this thesis and other methods. A summary is given in Section 6.7.

6.3 Key-point Extraction with Background Model

This thesis is dedicated to proposing a method that can quickly and effectively monitor illegally parked vehicles. The previous chapters have proposed various methods to achieve this function step by step. However, as the extraction density of key points changes from

sparse to dense, the computational cost of the system also increases, and additional background matching also exacerbates this consumption. Therefore, in order to reduce computational cost and achieve faster processing speed, the selection of key points needs to be more precise, that is, to delete useless key points while ensuring a sufficient number. Both semantic segmentation and background subtraction technologies can help to achieve this goal.

Considering the huge difference in execution time and memory usage between semantic segmentation technology and background subtraction technology, background subtraction was chosen to be introduced into this system. Compared with the semantic segmentation technology that requires a multi-layer convolutional neural network and a large amount of pre-training, the background subtractor provided by OpenCV can achieve fast and lightweight background subtraction and realise the required functions in this system.

6.3.1 The Comparison of Background Subtractors

There are currently four easy-to-use background subtractors provided by OpenCV, namely mixture of gaussians segmentation algorithm (MOG) [100], MOG2 [101], geometric multigid (GMG) [99] and k-nearest neighbours background segmentation algorithm (KNN) [98]. These four background subtractors can all realise part of the function of semantic segmentation technology, that is, the extraction of foreground objects (moving objects), and each has its own advantages. Figure 39 shows the test results of different background

subtractors on the same scene. The foreground areas have all undergone the same morphological processing.

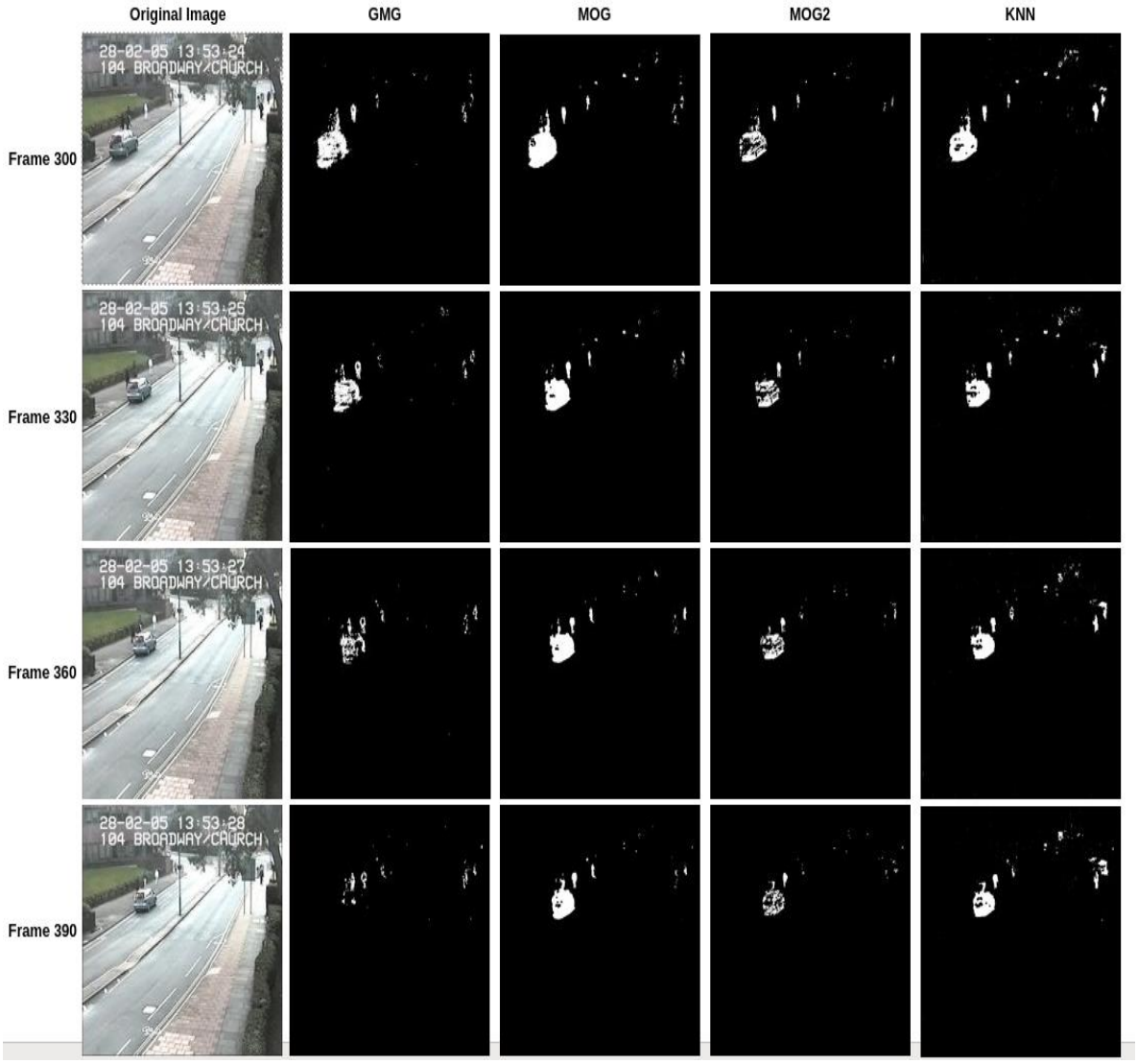

Figure 39 Moving targets are extracted by different background subtraction algorithms. The image shows a representative picture ranging from 300 to 390 frames. The binaries image uses white and black to represent the moving foreground object and the static background respectively. Different subtractors have different foreground object segmentation effects.

Considering the purpose of the above background subtraction methods are detecting moving objects, not all subtractors are suitable for this system. When the target is stationary,

the algorithm will change the background model within a few frames to turn the target from the foreground to the background. Under normal circumstances, the illegal vehicle tracking algorithm in this thesis needs a short time (usually within a few frames) to wait for the bounding box to stop and trigger an alarm. Specifically, the tracker started to work after the target is stationary, but the process from moving to static usually taking few frames. Hence, when the vehicle is finally parked, the background subtractor is already set this vehicle to the background. In other words, there is a time gap between tracker and foreground. This problem can be solved by changing the update rate to keep the parked vehicle belonging to foreground more time.

Marcomini et al. compared the precision rate and processing time of MOG, MOG2 and GMG in the context of vehicle segmentation in [113]. According to the comparison, it can be found that GMG has the worst segmentation effect and MOG2 has the fastest calculation speed. The research of Adinanta [114] and Trnovszký [115] proved that KNN has the best performance among the four subtractors.

The test results shown in Figure 39 also prove the conclusions of the above research. The MOG2 and KNN subtractors can reduce the update rate to slow down the time when the target turns from the foreground to the background when it is stationary. In addition, the MOG algorithm shadow detection is not perfect, even if it helps to restore the target contour to the original shape.

In general, KNN has better accuracy and shadow detection capabilities with an excellent computing speed. Considering that the purpose of this chapter is to locate densely

extracted key points in the target area, the existence of shadows will interfere with the selection of key points. Therefore, KNN has been used and its output will help the system to accurately select key points.

6.3.2 Precise Key Point Selection

When the specific position and range of the target are obtained by background subtraction, the positions of key points can be actively selected instead of being uniformly distributed in the bounding box. The specific effect is shown in Figure 40. The system will detect the positions of all key points, and use the difference between the foreground target and the background on the binarized image to determine the key point category. Once it is found that the key point is located in a non-vehicle area, such as the road surface or shadow in the bounding box, the key point will be deleted. Through a test on all datasets which used in this chapter, the number of key points after deletion is 25%-40% less than before.

Figure 40 The key point extraction case comes from the i-LIDS dataset (a) shows the target being tracked (b) Key points extraction method used in Chapter 5 (c) The proposed method deletes all non-vehicle key points

When the object turns from a moving state to a static state, the background subtraction cannot maintain this foreground object for a long time, even if reducing the update rate of the background model. Relevant research shows that some methods can be used to maintain a stationary target in the foreground. For example, [104] proposed a method called SHI (Segmentation History Image) to retain stationary pixels from the segmented stationary objects, so as to realise the detection of illegally parked vehicles based on background segmentation.

However, considering that the method proposed in this chapter does not require the detection and tracking of illegally parked vehicles to be based on the background subtractor method, only the first frame after the target is stationary will be used to test for invalid key point deletion. This solves the problem that the feature points cannot be extracted after the foreground target becomes the background.

In addition, the proposed method also has good applicability for partial occlusion. Through the intersection detection described in Chapter 5, the system can calculate the ratio of the intersection area (IOU). If the ratio is between 0-80%, which indicates that the target is partially occluded, and the system will delete the key points in the intersection area as shown in Figure 41.

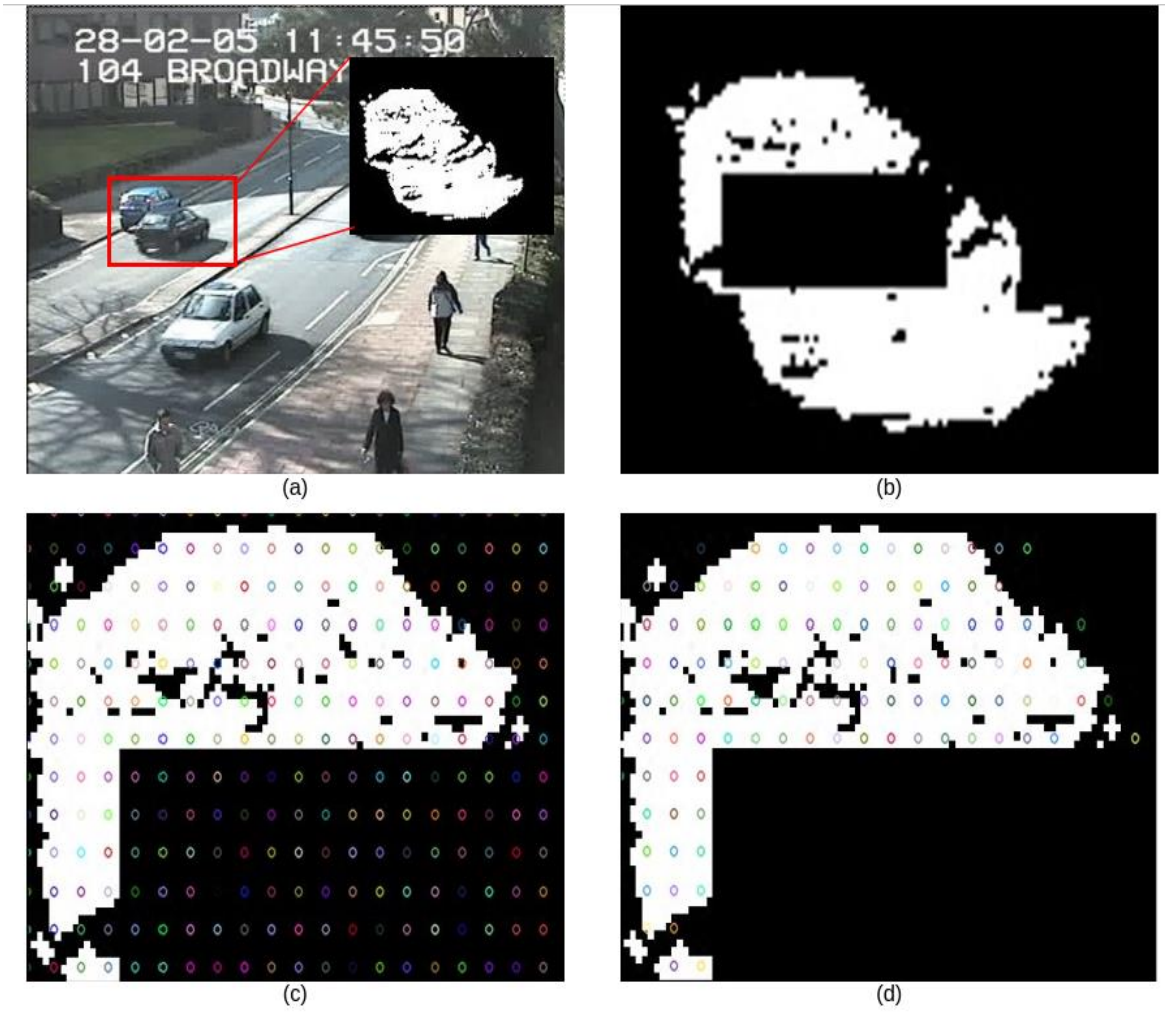

Figure 41 (a) The image comes from the i-LIDS dataset showing a partially occluded scene (b) On the image after the background is subtracted, the occluded area can be deleted by using the intersection detection of the bounding box. (c) Extract key points for the entire bounding box area (d) The key points of the blocked area and non-vehicle area are removed

For the case where the matched feature points are less than the threshold G, namely the target is completely occluded or leaves the no parking zone, the solution in Chapter 5 is to calculate the IOU of the intersection area. If the IOU exceeded the threshold (80%), the target will be judged to completely occluded and the tracker will remain in place. Otherwise, it means the target has left the no parking zone.

However, when tracking targets in actual scenes, the target detection module based on YOLOV3 and SORT algorithms (the method proposed in Chapter 3) do not always provide complete tracking results for all vehicles. For example, when two vehicles were close to each other, the system may lose tracking. According to the statistics given by the SORT algorithm in [17], 30% of the targets have not been tracked for at least 20% of its life span, and there are 1764 frags (number of fragmentations where a track is interrupted by miss detection) in the MOT benchmark sequences. The same problem occurs in this thesis. Therefore, a background matching method has been proposed in Chapter 5 to solve it when the intersection detection fails and the matched feature points are less than the threshold.

The ensuing problem is that the calculation cost of the system increases and the processing speed is reduced because the feature point matching is performed one more time. The processing speed for the i-LIDS dataset is only 7 frames per second. And the background frame needs to be manually selected to meet the condition that no vehicle is illegally parked in the screen. For different time periods of the same scene, such as day and night, the background frames also need to be selected separately. These conflicts with the goal of this thesis to realise the automatic detection of the system.

In order to reduce the computational pressure of the system, the method proposed in this chapter makes full use of the existing data for analysis and gives a simpler detection method. As shown in Figure 42, when the illegally parked vehicle is completely blocked, the number of matched feature points will be less than the threshold G, but a new foreground object (occluder) appears in the target area. When extracting the key points of the target area, the proposed method will extract the key points from the occluder. Since in the binaries image, the occluder and the illegally parked vehicle belongs to the foreground target (shown as the white area in Figure 42b), these key points will not be removed.

Figure 42 A typical completely occluded event from the i-LIDS dataset (b) Extract key points from the original target area and remove all key points belonging to the background

In contrast, as shown in Figure 43, when the target leaves the no parking zone, the original tracking area appears as a background area in the background subtraction screen, and all key points extracted in this area will be removed.

Figure 43 (a) The illegally parked vehicle leaves the no parking zone (b) The key points cannot be extracted because there is no foreground object in the original tracking area (red bounding box)

In summary, the proposed method is based on background subtraction to modify the method in Chapter 5, which improves the calculation speed of the system (from 7fps to 12fps) and reduces the consumption of feature point matching.

6.4 Tracking Under Lighting Changes

Tracking under changing light conditions is another challenge faced by this ASFP method. For systems such as the monitoring of illegally parked vehicles, it is necessary to adapt to tracking under drastic changes in light. Changes in light will not only lead to the appearance of shadows, but also affect the appearance of the target, especially for long-term events that may last for several hours. This thesis uses the SIFT feature descriptor-based matching

method and utilises the illumination invariance of the SIFT descriptor, which performs well in the case of light changes. In addition, the removal of non-vehicle key points through the background subtraction method also significantly enhances the system's robustness to illumination changes. This is because, compared to the key points on the road or the background, the key points from the vehicle are more likely to reflect the feature of corner points, edges and extreme points. These key points have relatively better stability when the light changes. For example, matching the feature points of the vehicle and the road under the same lighting conditions, it can be found that the matching result of the vehicle has experienced a change from 130 to 90, while the matching result of the road has dropped from 36 to 3.

Figure 44 shows the proposed method tested on a video with obvious light changes, and the results are shown in Figure 45. Figure 45(a) shows the number of matched feature points and the number only drops when occlusion occurs. The dramatic change of light can be obtained by observing the fluctuation of the grey value. Specifically, in order to reflect the change of light, the average grey value of the vehicle area has been calculated and counted, and the change of the average grey value is consistent with the change of lighting conditions. Figure 45(c) shows the proportion of matched feature points to all extracted feature points. Through this ratio, it can be found that the tracking performance is very robust to lighting.

Figure 44 The image shows the change in lighting conditions. The tracked target is marked by a red bounding box. This video sequence shows dramatic changes in lighting within a minute

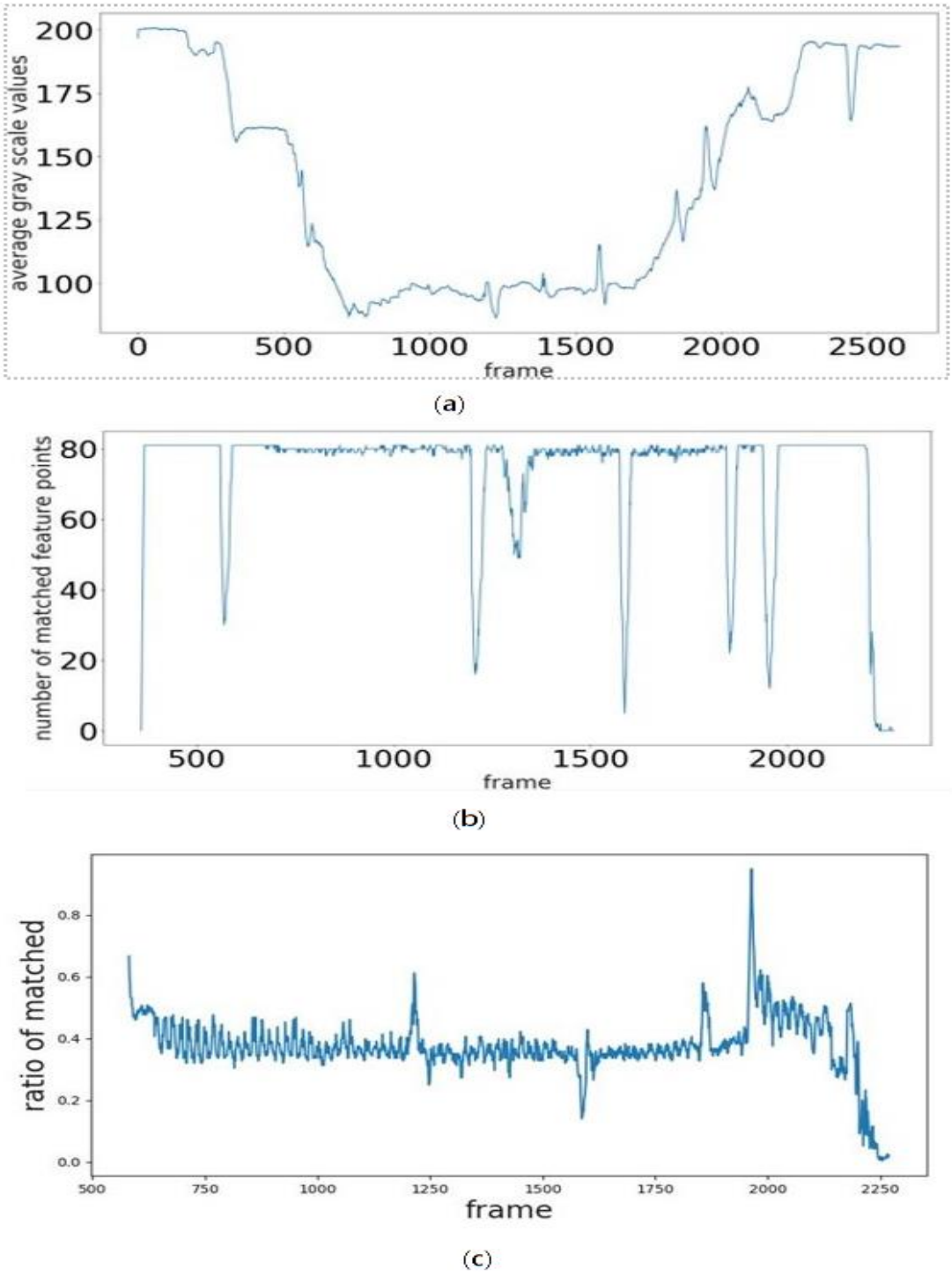

Figure 45 The matching result with lighting changes. (a) The lighting changes of the scene can be shown by the changes of the average grey value. It can be seen that compared to the highest value of 200, the grey value drops to 40 when the environment becomes dark. (b) In the scene of light changes but the ASFP method maintains a stable tracking. The value of matching only decreased due to occlusion. (c) The proportion of matched feature points to the total feature points can also be used to judge the stability and robustness of the tracking.

The ASFP method also tested on the video sequence as shown in Figure 46. It can be clearly seen through the image that the change of light causes the appearance and disappearance of shadows. At the same time, the target experienced multiple complete occlusions during the tracking period, which made tracking more difficult.

Figure 47 shows the test results of the proposed method on the video sequence. The first thing to be sure is that the proposed method keeps track under light changes and occlusions. At the same time, compared with the SIFT-based and the dense SIFT-based method proposed in previous chapter, the method proposed in this chapter has better tracking results.

Table 5 shows the comparison between the proposed ASFP method and the other two tracking methods based on edge [104] and colour [116]. The running average variations in matching of three methods has huge difference. It can be seen from Table 5 that the matching results in this chapter have achieved better performance than the other two technologies, especially when the light changes drastically.

Figure 46 The image shows the dramatic lighting changes from 900 to 1700 frames. The tracked target is marked by a red bounding box

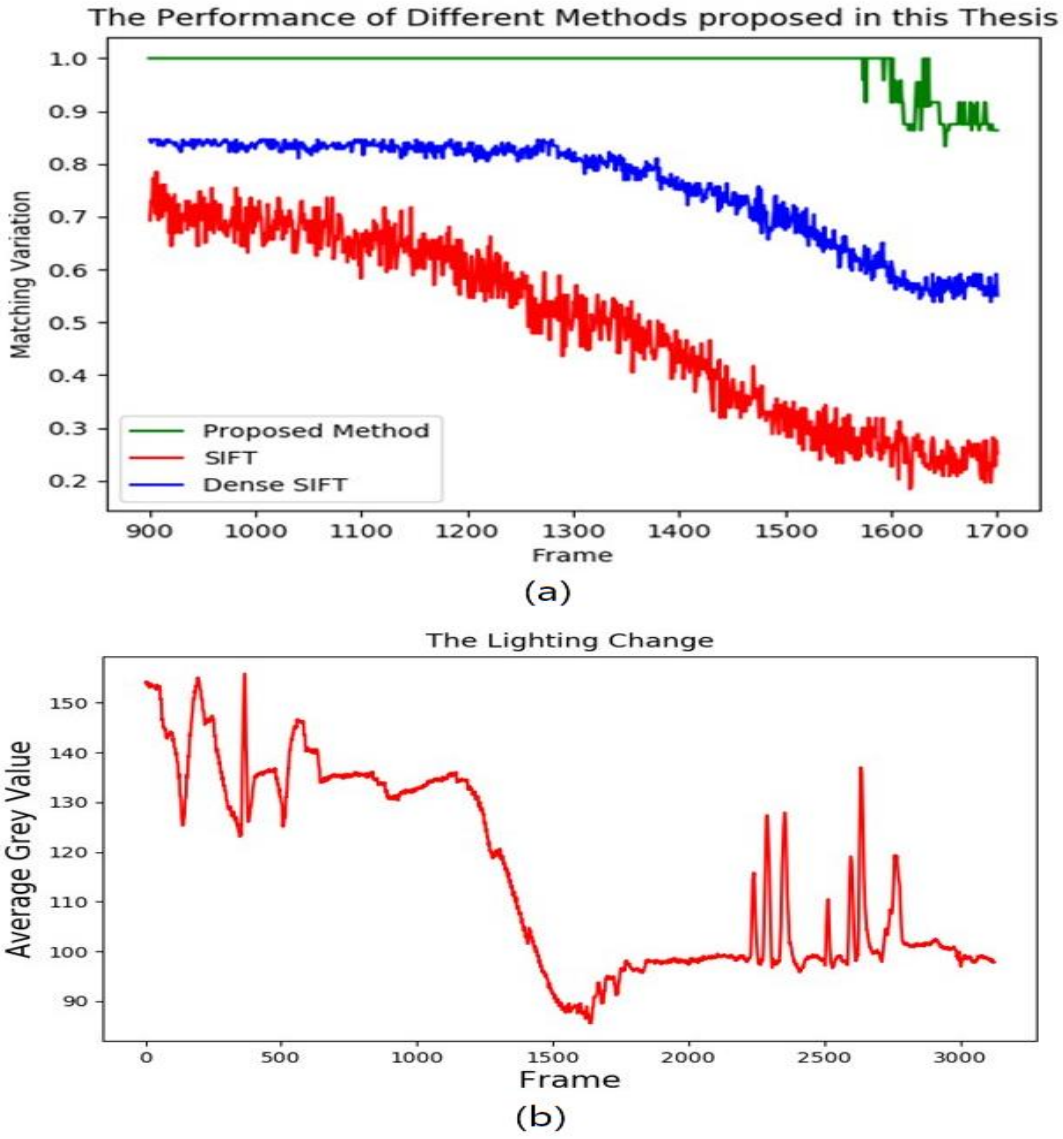

Figure 47 (a) This line chart records the matching results of the three methods proposed in this thesis from 900 frames to 1700 frames. Among them, the three methods have undergone illumination changing and occlusion together. (b) This line chart indicates the change of lighting by showing the average grey value of the overall image

Frame Number	Proposed method	Edge energy tracking [104]	I-MCHSR tracking [116]
900	100%	88.54%	95.68%
1200	100%	83.43%	76.06%
1400	100%	78.51%	71.89%
1700	86.30%	67.54%	60.09%

Table 5 Three types of tracking results of illegally parked vehicles based on different features. The data of Edge energy tracking and I-MCHSR tracking are obtained by [104] and [116], respectively.

6.5 False Positive Events Removal

According to the dense SIFT based tracking results shown in Table 3, a total of 17 false positive alarms were generated in a total of 6 hours of video sequence. YOLOv3 algorithm as the detection module provides all vehicle targets to tracking system. Therefore, once the false positive alarm occurs, it can be concluded that the YOLOv3 algorithm gives an incorrect detection result. As shown in Figure 48, the YOLOv3 algorithm incorrectly detects non-vehicle objects.

Figure 48 False positive events caused by failed detections. This is a test result that appeared in the previous chapters and contained a false positive event. This error is due to the YOLOV3 algorithm mistakenly recognizing a street light and vehicle as a truck.

Since YOLOv3 was used to classify objects, in some cases, non-vehicle objects will be classified as cars, which leads to the algorithm to extract key points in this area. In the previous chapters, the deletion of the tracker requires feature matching between the background frame and the current frame to determine whether the target has left the no parking zone. For the misidentified non-vehicle target, the algorithm will recognise it as still in place, even if the matched key point is lower than the threshold G. For example, Figure 48 shows a typical false positive event from the test results in Chapter 5.

In this chapter, the proposed ASFP method solves this problem and uses foreground targets to detect this type of event. Through the observation and analysis of all false positive events, it can be found that the number of matched feature points of these false positive events will be very small (very close to 0). This is because the feature points extracted by this system will be filtered by foreground objects. All feature points belonging to the background will not be used for matching (see section 6.3.2). Therefore, when the detection results of these non-vehicle objects which belonging to the background appear in the screen, the system will delete these false positive detections based on the number of feature points.

6.6 Results and Discussion

The proposed technology was tested on two different datasets and divided into two scenarios: illegally parked vehicles on crowded roads and tracking targets that lasted from daytime to dusk accompanied by occlusion and illumination changes. In the former scenario, the alarm event is generated after the target is stationary in the ROI for more than 60 seconds. The results have been compared with other advanced methods from the literature, and it was found that this method obtained state of the art results. In addition, through testing on the latter dataset, the proposed method showed good robustness to light changes and occlusion conditions. The running speed and computing cost have been optimised and improved. Through testing, it was found that the proposed method can run on PC with an Intel Xeon(R) CPU E5-1620 V3 running at 3.50 GHz, NVIDIA Quadro K4200 and with 15.6 GB RAM at a speed of 12 frames per second. This speed is calculated after testing

all videos. Compared with the original YOLOV3 algorithm (78fps) and SORT algorithm (260fps), the speed of proposed system has been significantly reduced. This is firstly because the tracking module based on feature point matching reduces the computing speed, and secondly because the platform running the system cannot provide sufficient computing power. This speed can be higher for more powerful computer hardware, such as NVIDIA 3080 GPU card and Intel I9 CPU. Secondly, the lack of code optimization is also the reason for the unsatisfactory speed. Since the program involves the combination of multiple modules and parallel operations, a better optimization can speed up the running speed of the program.

For example, [125] provides a good idea to run the SIFT algorithm on the GPU to improve the speed of the program. The research of Blaine et al. found that the SIFT algorithm run and optimized by GPU can reduce the computational power consumption by 87%. According to the test of the YOLOV3 algorithm, it can be found that the YOLOV3 algorithm that runs alone on this PC platform can reach a speed of 18-20fps. Considering that the largest computational power consumption of this program occurs in the target detection section, if other section can be optimized, the running speed of the entire program can eventually be very close to 20fps.

In addition, the proposed method is always accompanied by defects caused by the use of the YOLOv3 algorithm. In detail, because there is no regional sampling, YOLOV3 has a good performance on global information, but it performs poorly on small objects. Therefore, once very small vehicles and extremely dense targets cannot be detected, tracking failures will be unavoidable. In the multi-object tracking stage, the SORT algorithm does help the

system establish the target tracking and digital ID, but as introduced in section 3.7, the purely coordinate-based target tracking algorithm has certain limitations. Relying only on the target coordinates to correlate the same target in the previous and subsequent frames is less effective in some situations. Therefore, the proposed method uses feature point matching to avoid this problem.

6.6.1 I-LIDS dataset

The method was first tested on two sets of parked vehicle datasets. Each set of video sequences contains different scenes and different times, covering the increasingly worse tracking environment. The first set of video sequences contains a series of short events, and the video length exceeds 6 hours. The second set contains multiple long-term events (5-15 minutes each). In these two sets, the alarm is generated after the vehicle is stationary in the parking violation area for 60 seconds. Table 6 shows the result obtained from the first set. Among 105 real events, the proposed method successfully detected 100 of them, and these alarms were generated within 10 seconds, which fully met the official requirements of i-LIDS. The overall recall rate exceeds 95%. At the same time, no false positive alarms were generated compared with the method in the previous chapter, there were 17 false alarms. This is because all non-vehicle targets have been filtered. At the same time, the overall frame rate of the system reached 12fps. The proposed method is also compared with other technologies, as shown in Table 7. It can be seen that the system can detect illegally parked vehicles more accurately than other methods.

Video Name	Duration (hh:mm:ss)	Total Event	True Positive	False Positive	False Negative
PVTEA101a	01:21:29	26	25	0	1
PVTEA101b	00:24:40	8	8	0	0
PVTEA102a	00:51:56	18	18	0	0
PVTEA103a	01:03:54	16	15	0	1
PVTEA201a	00:18:50	4	3	0	1
PVTEA202b	01:05:23	17	17	0	0
PVTEA301a	00:15:29	3	2	0	1
PVTEA301b	00:28:07	7	6	0	1
PVTEA301c	00:27:09	6	6	0	0
Total	**06:16:57**	**105**	**100**	**0**	**5**

Table 6 The proposed method is tested on video sequence from i-LIDS dataset. (also used in Chapter 3 and 5)

Algorithm	Total Events	True Positive	False Positive	False Negative	Precision	Recall	F1
Ours-Chapter 3	105	51	3	54	0.82	0.49	0.61
Ours-Chapter 5	105	100	17	5	0.85	0.95	0.90
ASFP method	105	100	0	5	1.00	0.95	0.97
Hassan[104]	105	99	18	6	0.84	0.94	0.88
Jo[78]	105	93	N/A	N/A	0.89	0.89	0.89

Fan[117]	105	89	8	N/A	0.91	0.84	0.87
Albiol [81]	105	N/A	N/A	N/A	0.98	0.96	0.97

Table 7 The proposed method is compared with others. N/A: Data is not provided from the author of the technology

The proposed method tested the detection and tracking results of long-term events in the second set from the i-LIDS dataset. Compared to the short-term events of the first set (average of 2-3 minutes per event), long-term events require more powerful performance to keep tracking. Especially some blurred targets can only be detected by the YOLOv3 algorithm for a few frames. In fact, the proposed method is robust to poor target detection results. During this period, complex light changes and frequent occlusion have stricter requirements for tracking. Through the test, all events from the second set have been successfully detected and tracked, and no false alarms have been generated. The overall results of the proposed method are shown in Table 8.

Video Name	**Duration (hh:mm:ss)**	**Total Event**	**True Positive**	**False Positive**	**False Negative**
PVTEN102d	00:19:43	1	1	0	0
PVTEN201a	00:29:56	2	2	0	0
PVTEN201b	00:30:00	1	1	0	0
PVTEN201c	00:20:05	2	2	0	0
PVTEN201d	00:19:54	2	2	0	0
Total	**01:59:38**	**8**	**8**	**0**	**0**

Table 8 Results from long-term events testing. The proposed method passed the test very well and did not produce any errors or failures

The Figure 49-56 show the tracking results of the proposed method on a series of videos from the i-LIDS dataset.

Figure 49 Tracking from night events. Both targets can be accurately detected and tracked, including an blurred target.

Figure 50 Tracking under completely occlusion. Even if the target is almost completely occluded, the tracking remains stable.

Figure 51 Tracking in low light environment at night. It can be seen that the lighting conditions for the scene are very poor but still cannot interfere with the tracking of the target.

Figure 52 Tracking under partial occlusion. The unobstructed part of the target can provide enough feature points for matching.

Figure 53 Tracking on rainy days, including a blurred target and refraction of light. The environmental interference in this scenario is more complicated, which puts higher requirements on the robustness of tracking.

Figure 54 Tracking in rainy days, including a half-displayed target and a blurred target. One of the targets is almost completely occluded, and the system can only detect one matching feature point.

Figure 55 Tracking of two targets which close to each other. In this case, the YOLOV3 algorithm can easily fail to detect because they are too close to each other.

Figure 56 Tracking under light changes. The target is far away from the lens and blurry, so 409 matched feature points can provide very robust tracking.

6.6.2 Day to Dusk Video

The above testing shows the tracking performance of various events under a short-term light changing. In order to test the tracking effect of the proposed method on the target under the completely changed background environment, a video dedicated to this purpose was made. As shown in Figure 57, the target area was recorded from 10 am to 6 pm. During this period, the target experienced a change of light from day to dusk, accompanied by the occlusion of the people passing by. Through testing, it can be found that the proposed method does not occur target loss during the event and keeps tracking stable all the time.

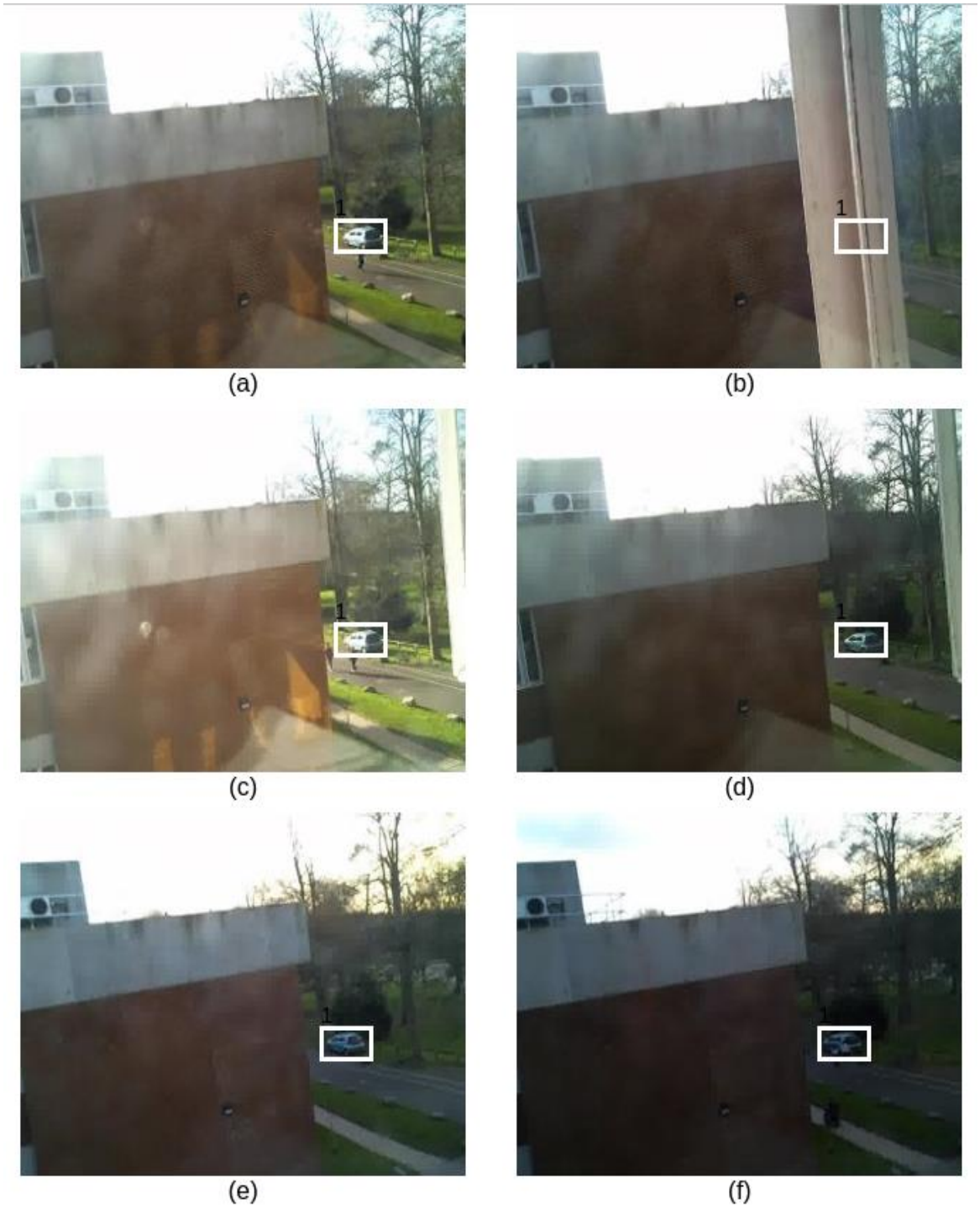

Figure 57 the image comes from a fixed position from 10am to 6pm. (a) target vehicle parked at 10am(b)target blocked by window, this situation can be seen as occlusion event (c) the strong light (d)-(e) weak light (f) target vehicle parked at 6pm.

6.7 Conclusion

Based on the method proposed in Chapter 5, an ASFP method proposed in this chapter that can accurately select the location. The focus is to improve the robustness of the tracker to light changes and occlusion conditions, and improve the processing speed of the system. Changes in the tracking environment usually bring great challenges to the system, for example, when the tracked object goes through different time periods of the day, its appearance changes will cause the tracking to fail. The ASFP method overcomes these challenges. The traditional but efficient background subtraction algorithm has been used to extract the illegal target from the background, by using the position of the pixels in the foreground area to delete the key points that were not related to the target, and the dense SIFT method ensured sufficient key points for matching. The system was tested on videos from different datasets, and the state of the art results were achieved while maintaining the computing speed of 12fps (7fps in Chapter 5).

Chapter 7 Conclusions and Future Work

7.1 Conclusions

This thesis is dedicated to the research and development of a reliable and efficient parking vehicle monitoring system, which can be used for the detection and tracking of illegally parked vehicles. In the past, many studies have proposed a variety of methods for the monitoring of illegally parked vehicles. Due to technological progress in the past decade, especially the proposal of target detection and multi-object tracking algorithms represented by deep learning networks, more powerful algorithms can now be used to achieve various image processing functions more accurately. Considering the importance of parking vehicle monitoring, this thesis proposes a method that uses a combination of deep learning detection and key point matching.

This research focuses on how to keep tracking of illegally parked vehicles in different environments and scenarios. This thesis that introduces this research is divided into four progressive steps to complete the desired results. For the specific implementation process, please see the past chapters.

The first step of the project is to establish an illegally parked vehicle detection module, which is introduced in Chapter 3. A vehicle detection framework based on the deep learning algorithm YOLOv3 and the SORT algorithm is established. All detected vehicles are classified as illegal vehicles and others through their position and speed judgment. If a stationary vehicle is found in the no parking zone, an alarm will be generated and the target will be marked by a bounding box. However, even if the target is identified as an illegally parked

vehicle, the corresponding tracking cannot be maintained during the event. This is because the MOT based tracking cannot continuously track the target. This chapter focuses on the fact that the surveillance system has great defects in the tracking of targets, which is caused by the inability of the SORT algorithm to continuously track the target. However, it can be observed that the target detection module based on YOLOv3 algorithm can well realise the recognition of the target, and the target state judgment module based on position and speed can classify the recognised vehicle very well. Although the proposed method achieves a precision rate of 94%, more than 50% of the targets are not tracked (49% recall), indicating that the method cannot adapt to the work in the real scene. In subsequent chapters, the method of key-point matching will provide a more powerful solution.

Chapter 4 and Chapter 5 discuss the tracking method based on key point matching which can be used in traffic scenes. Both chapters use SIFT feature descriptors as the basis for establishing tracking matching. Taking into account that occlusion will cause the global features of the detection area to change, in comparison, the SIFT feature is a local feature, which makes it robust to occluded targets. But on the other hand, the number of key points determines the tracking effect of the occluded target. Especially for occluded targets, only a few key points can be successfully matched. Taking these conditions into account, the feature points need to be collected intensively to ensure that the match can be established under extreme conditions. In addition, the completely occluded target also needs to be distinguished from the target leaving the no parking zone.

Chapter 4 mainly uses sparse methods to extract SIFT feature descriptors. The method from Chapter 3 provides the first frame of the detection result and state judgment of the illegally

parked vehicle and the identity ID. Since the extraction of feature points needs to separate the corner points, edges, and extreme points of the target, when the target is occluded, a large number of key points will be lost, and the matching will fail. Therefore, the occluded target will not be tracked very stably. For the matching of feature points, the well-known BF matching method is selected, and the output matching results will be used as the basis for tracking. When the number of successfully matched feature points exceeds the threshold, it indicates that the target is still in place. The method was tested on 21 video sequences and achieved an overall recall rate of 84.8% in a total of 48 real events. It can be seen from the results that the main challenge faced by this method is the inability to extract enough feature points for matching due to the blurred target. Secondly, a large area of occlusion will lead to a too low matching success rate. Although this method uses a timer to retain the tracker in the case of a short-term matching failure, the release of the tracker when the target leaves the no parking zone is affected and cannot even be deleted. Considering that these are the most common problems in illegal vehicle monitoring, the next chapter proposes a more powerful key point extraction method, which can track the object when the target is completely occluded and will not interfere with the release of the tracker.

In order to adapt to complex road scenes, the key point extraction method based on dense SIFT is used in Chapter 5. The proposed extraction method can not only accurately extract effective key points when the target is partially occluded, but also has a good performance for fuzzy targets and small targets. Then the method of intersection detection is used to determine whether the target has left the no parking zone. Experimental results show that

the proposed method can obtain an overall recall rate of 95.6% even under frequent occlusions. In more than 7 hours of video, the method tested the tracking performance in different scenes and different camera angles. By comparing the results with the previous chapters, this method significantly improves the performance of the monitoring system.

It can be seen from Chapter 4 and Chapter 5 that although the SIFT feature descriptor has illumination invariance, a certain degree of illumination robustness can be guaranteed. However, considering the complex scene where the event occurs, this method needs to be improved for the scene where the light changes. As we all know, background subtraction as a traditional motion detection method has a good performance on moving targets extraction. Since the extraction of the foreground target is based on the pixel-scale differential detection, compared to the bounding box generated by the YOLO algorithm, the foreground target output by the background subtraction method has a contour close to the real target. By reducing the update frequency of the background model, this method can also continuously output the detection result when the moving target becomes stationary.

The main motivation of Chapter 6 is to track illegally parked vehicles steadily in scenes with changing lighting, especially the system needs to be suitable for tracking such as from day to night. And considering the effect of actual use, the computational consumption of the system needs to be reduced. The idea is to select key points more accurately while keeping enough key points for matching. Using the foreground target area detected by background subtraction, all feature points unrelated to the tracked vehicle will be removed. The proposed ASFP method has been evaluated in different real scenarios. Experimental results show that the ASFP method achieves the best tracking results under illumination changes

with a relatively small number of feature points (compared to the number in Chapter 5), but the speed is improved.

The comparison of the methods proposed in Chapter 3,5, and ASFP on the same dataset is shown in the following table and contains the test results of other algorithms on the dataset.

Algorithm	Total Events	True Positive	False Positive	False Negative	P	R	F1
Chapter 3	105	51	3	54	0.82	0.49	0.61
Chapter 5	105	100	17	5	0.85	0.95	0.90
ASFP method	105	100	0	5	1.00	0.95	0.97
Hassan [104]	105	99	18	6	0.84	0.94	0.88
Jo [78]	105	93	N/A	N/A	0.89	0.89	0.89
Fan [117]	105	89	8	N/A	0.91	0.84	0.87
Albiol [81]	105	N/A	N/A	N/A	0.98	0.96	0.97

Table 9 Tracking performance and comparison with I-LIDS dataset. These results show comparisons between all versions of the proposed method and other methods. P: Precision rate. R: Recall rate. F1: A measure that combines precision and recall.

7.2 Future Work

The method proposed in this thesis can be further improved to try to obtain broader application. The method based on deep learning detection and feature point matching can provide a recall rate more than 95%. Nevertheless, in some cases, a target that is too small may cause the detection of the object to fail. Therefore, additional target detection modules can be deployed to deal with the missed illegal targets. For example, the target detected by the background subtraction technology and semantic segmentation can also help the algorithm to make up for the failed target. Such methods may require a more powerful processing platform, because the output results of such methods need to be processed in the shortest time. In addition, the SIFT feature descriptor is used in this thesis because of the robustness of the feature to light. At the same time, feature descriptors such as SURF [26], DAISY [118] and ORB [31] can also achieve the similar functions of SIFT. In addition, based on the deep learning, many states of art descriptor can also provide more robust matching, such as the SuperPoint [119] and ContextDesc [120]. The high-speed, dense extraction and other characteristics of these descriptors may be very useful for this topic.

The work of this thesis only focuses on the monitoring of illegally parked vehicles. In fact, using the YOLOv3 algorithm can detect more than 80 kinds of objects, this research can be applied to more scenes to track different objects of interest. For example, in the sensitive area intrusion detection, abandoned baggage detection and other fields, this method can provide a good monitoring effect. For these scenarios that require continuous monitoring

for 24 hours, mistakes in manual monitoring can lead to very serious results. This research is dedicated to realising automated monitoring in fixed scenarios, which can make up for mistakes in manual monitoring.

References

1. Green, Mary W. The appropriate and effective use of security technologies in US schools: A guide for schools and law enforcement agencies. US Department of Justice, Office of Justice Programs, National Institute of Justice, 1999.

2. Avigilon Corporation. Enhancing human attention span with HD Analytics,2014.Web. 28 Jul.2021. https://www.avigilon.com/news/white-papers/enhancing-human-attention-span-with-self-learning-video-analytics

3. AXIS Video Motion Detection, 2016.Web. 28 Jul.2021. https://www.axis.com/products/axis-video-motion-detection.

4. Huang, Shih-Chia. "An advanced motion detection algorithm with video quality analysis for video surveillance systems." IEEE transactions on circuits and systems for video technology 21.1.(2010): 1-14.

5. Truong, Tung Xuan, and Jong-Myon Kim. "Fire flame detection in video sequences using multi-stage pattern recognition techniques." Engineering Applications of Artificial Intelligence25.7.(2012): 1365-1372.

6. IDC Releases China Video Surveillance Equipment Tracker Report AI & 5G Bring Video Surveillance To A New Era. 2019.Web. 28 Jul.2021 https://www.telecomtv.com/content/video-technology/idc-releases-china-video-surveillance-equipment-tracker-report-ai-5g-bring-video-surveillance-to-a-new-era-36395/

7. "CCTV: Too Many Cameras Useless, Warns Surveillance Watchdog Tony Porter". BBC News, 2021, https://www.bbc.co.uk/news/uk-30978995.

8. Wilson, Cecilia, et al. "Speed cameras for the prevention of road traffic injuries and deaths." Cochrane database of systematic reviews 11 (2010).

9. Stipaničev, Darko, et al. "Advanced automatic wildfire surveillance and monitoring network." VI International Conference on Forest Fire Research. 2010.

10. Boyle, Louise. "Global fires are up 13% from 2019's record-breaking numbers". The Independent. 8 September 2020.Web. 28 Jul.2021

11. AT&T,AT&T Digital Life - Home Security & Automation Systems. 2016.Web. 28 Jul.2021. https://www.att.com/digital-life/.

12. Aljoufie, Mohammed. "Analysis of Illegal Parking Behavior in Jeddah." Current Urban Studies 4.04 (2016): 393.

13. "Stop Sign Cameras Are Costing Some Park Visitors Plenty". NBC Los Angeles, 2017.Web. 28 Jul.2021

14. Kidwell, David, and Abraham Epton. Emanuel's Speed Cameras Issue $2.4 Million In Bad Tickets. 2015. Web. 24 July 2021. https://www.chicagotribune.com/investigations/ct-speed-camera-bad-tickets-met-20151117-story.html.

15. Elsom, Dan. "The Parking Rules You Should Know - From Yellow Lines To Cycle Tracks". The Sun, 2018.Web. 24 Jul.2021.

16. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

17. Bewley, Alex, et al. "Simple online and realtime tracking." 2016 IEEE international conference on image processing (ICIP). IEEE, 2016.

18. Branch, Home Office Scientific Development. "Imagery library for intelligent detection systems (i-lids)." 2006 IET conference on crime and security. IET, 2006.

19. Vezzani, Roberto, and Rita Cucchiara. "Video surveillance online repository (visor): an integrated framework." Multimedia Tools and Applications 50.2 (2010): 359-380.

20. Lowe, David G. "Object recognition from local scale-invariant features." Proceedings of the seventh IEEE international conference on computer vision. Vol. 2. Ieee, 1999.

21. Zou, Zhengxia, et al. "Object detection in 20 years: A survey." arXiv preprint arXiv:1905.05055 (2019).

22. Ma, Jiayi, et al. "Image matching from handcrafted to deep features: A survey." International Journal of Computer Vision129.1 (2021): 23-79.

23. Lindeberg, Tony. "Feature detection with automatic scale selection." International journal of computer vision 30.2 (1998): 79-116.

24. Piccinini, Paolo, Andrea Prati, and Rita Cucchiara. "Real-time object detection and localization with SIFT-based clustering." Image and Vision Computing 30.8 (2012): 573-587.

25. Hashmi, Mohammad Farukh, Aaditya R. Hambarde, and Avinash G. Keskar. "Copy move forgery detection using DWT and SIFT features." 2013 13th International conference on intellient systems design and applications. IEEE, 2013.

26. Bay, Herbert, et al. "Speeded-up robust features (SURF)." Computer vision and image understanding 110.3 (2008): 346-359.

27. Pranata, Yoga Dwi, et al. "Deep learning and SURF for automated classification and detection of calcaneus fractures in CT images." Computer methods and programs in biomedicine 171 (2019): 27-37.

28. Zhao, Jin, Sichao Zhu, and Xinming Huang. "Real-time traffic sign detection using SURF features on FPGA." 2013 IEEE high performance extreme computing conference (HPEC). IEEE, 2013.

29. Mistry, Darshana, and Asim Banerjee. "Comparison of feature detection and matching approaches: SIFT and SURF." GRD Journals-Global Research and Development Journal for Engineering 2.4 (2017): 7-13.

30. Calonder, Michael, et al. "Brief: Binary robust independent elementary features." European conference on computer vision. Springer, Berlin, Heidelberg, 2010.

31. Rublee, Ethan, et al. "ORB: An efficient alternative to SIFT or SURF." 2011 International conference on computer vision. Ieee, 2011.

32. Yang, Tsun-Yi, et al. "Deepcd: Learning deep complementary descriptors for patch representations." Proceedings of the IEEE International Conference on Computer Vision. 2017.

33. Tian, Yurun, Bin Fan, and Fuchao Wu. "L2-net: Deep learning of discriminative patch descriptor in euclidean space." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

34. Mishchuk, Anastasiya, et al. "Working hard to know your neighbor's margins: Local descriptor learning loss." arXiv preprint arXiv:1705.10872 (2017).

35. Dai, Zhuang, et al. "A comparison of cnn-based and hand-crafted keypoint descriptors." 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.

36. Mistry, Darshana, and Asim Banerjee. "Comparison of feature detection and matching approaches: SIFT and SURF." GRD Journals-Global Research and Development Journal for Engineering 2.4 (2017): 7-13.

37. Cannons, Kevin. "A review of visual tracking." Dept. Comput. Sci. Eng., York Univ., Toronto, Canada, Tech. Rep. CSE-2008-07 242 (2008).

38. Lucas, Bruce D., and Takeo Kanade. "An iterative image registration technique with an application to stereo vision." 1981.

39. Denman, Simon, Vinod Chandran, and Sridha Sridharan. "An adaptive optical flow technique for person tracking systems." Pattern recognition letters 28.10 (2007): 1232-1239.

40. Kalman R E, Bucy R S. New results in linear filtering and prediction theory[J]. 1961.

41. Patel, Hitesh A., and Darshak G. Thakore. "Moving object tracking using kalman filter." International Journal of Computer Science and Mobile Computing 2.4 (2013): 326-332.

42. Yang, Changjiang, Ramani Duraiswami, and Larry Davis. "Fast multiple object tracking via a hierarchical particle filter." Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. Vol. 1. IEEE, 2005.

43. Rui, Yong, and Yunqiang Chen. "Better proposal distributions: Object tracking using unscented particle filter." Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. Vol. 2. IEEE, 2001.

44. Cho, Jung Uk, et al. "A real-time object tracking system using a particle filter." 2006 IEEE/RSJ international conference on intelligent robots and systems. IEEE, 2006.

45. Hu, Jwu-Sheng, Chung-Wei Juan, and Jyun-Ji Wang. "A spatial-color mean-shift object tracking algorithm with scale and orientation estimation." Pattern Recognition Letters 29.16 (2008): 2165-2173.

46. Zhou, Huiyu, Yuan Yuan, and Chunmei Shi. "Object tracking using SIFT features and mean shift." Computer vision and image understanding 113.3 (2009): 345-352.

47. Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. "Real-time tracking of non-rigid objects using mean shift." Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662). Vol. 2. IEEE, 2000.

48. Ross, David A., et al. "Incremental learning for robust visual tracking." International journal of computer vision 77.1 (2008): 125-141.

49. Wang, Dong, Huchuan Lu, and Ming-Hsuan Yang. "Online object tracking with sparse prototypes." IEEE transactions on image processing 22.1 (2012): 314-325.

50. Kwon, Junseok, and Kyoung Mu Lee. "Tracking by sampling trackers." 2011 International Conference on Computer Vision. IEEE, 2011.

51. Kwon, Junseok, and Kyoung Mu Lee. "Visual tracking decomposition." 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010.

52. Collins, Robert T., Yanxi Liu, and Marius Leordeanu. "Online selection of discriminative tracking features." IEEE transactions on pattern analysis and machine intelligence27.10 (2005): 1631-1643.

53. Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." Proceedings of the fifth annual workshop on Computational learning theory. 1992.

54. Osuna, Edgar, Robert Freund, and Federico Girosit. "Training support vector machines: an application to face detection." Proceedings of IEEE computer society conference on computer vision and pattern recognition. IEEE, 1997.

55. Avidan, Shai. "Support vector tracking." IEEE transactions on pattern analysis and machine intelligence 26.8 (2004): 1064-1072.

56. Tian, Min, Weiwei Zhang, and Fuqiang Liu. "On-line ensemble SVM for robust object tracking." Asian conference on computer vision. Springer, Berlin, Heidelberg, 2007.

57. Yin, Yingjie, et al. "Online state-based structured SVM combined with incremental PCA for robust visual tracking." IEEE transactions on cybernetics 45.9 (2015): 1988-2000.

58. Sharma, Vijay K., and Kamala Kanta Mahapatra. "Visual object tracking based on sequential learning of SVM parameter." Digital Signal Processing 79 (2018): 102-115.

59. Bai, Yancheng, and Ming Tang. "Robust tracking via weakly supervised ranking SVM." 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012.

60. Ning, Jifeng, et al. "Object tracking via dual linear structured SVM and explicit feature map." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

61. Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009.

62. Wang, Lijun, et al. "Visual tracking with fully convolutional networks." Proceedings of the IEEE international conference on computer vision. 2015.

63. Ma, Chao, et al. "Hierarchical convolutional features for visual tracking." Proceedings of the IEEE international conference on computer vision. 2015.

64. Nam, Hyeonseob, and Bohyung Han. "Learning multi-domain convolutional neural networks for visual tracking." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

65. Cui, Zhen, et al. "Recurrently target-attending tracking." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

66. Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

67. He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE transactions on pattern analysis and machine intelligence 37.9 (2015): 1904-1916.

68. Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.

69. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015): 91-99.

70. Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.

71. Lee, Jong Taek, et al. "Real-time detection of illegally parked vehicles using 1-D transformation." 2007 IEEE Conference on Advanced Video and Signal Based Surveillance. IEEE, 2007.

72. Bevilacqua, Alessandro, and Stefano Vaccari. "Real time detection of stopped vehicles in traffic scenes." 2007 IEEE Conference on Advanced Video and Signal Based Surveillance. IEEE, 2007.

73. Akhawaji, Rami, Mohamed Sedky, and Abdel-Hamid Soliman. "Illegal parking detection using Gaussian mixture model and kalman filter." 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA). IEEE, 2017.

74. Chunyang, Mu, Ma Xing, and Zhang Panpan. "Smart detection of vehicle in illegal parking area by fusing of multi-features." 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies. IEEE, 2015.

75. Sarker, Md Mostafa Kamal, Cai Weihua, and Moon Kyou Song. "Detection and recognition of illegally parked vehicles based on an adaptive gaussian mixture model and a seed fill algorithm." Journal of information and communication convergence engineering 13.3 (2015): 197-204.

76. Hassan, Waqas, et al. "Real-time occlusion tolerant detection of illegally parked vehicles." International Journal of Control, Automation and Systems 10.5 (2012): 972-981.

77. Mutsuddy, Aapan, et al. "Illegally parked vehicle detection based on Haar-cascade classifier." International Conference on Intelligent Computing. Springer, Cham, 2019.

78. Jo, Kang-Hyun. "Cumulative dual foreground differences for illegally parked vehicles detection." IEEE Transactions on Industrial Informatics 13.5 (2017): 2464-2473.

79. Zhao, Xiping, Xiaodong Cheng, and Xiaofei Li. "Illegal Vehicle Parking Detection Based on Online Learning." Int. J. Web Appl. 5.3 (2013): 128-135.

80. Maddalena, Lucia, and Alfredo Petrosino. "Self organizing and fuzzy modelling for parked vehicles detection." International conference on advanced concepts for intelligent vision systems. Springer, Berlin, Heidelberg, 2009.

81. Albiol, Antonio, et al. "Detection of parked vehicles using spatiotemporal maps." IEEE Transactions on Intelligent Transportation Systems 12.4 (2011): 1277-1291.

82. Xie, Xuemei, et al. "Real-time illegal parking detection system based on deep learning." Proceedings of the 2017 International Conference on Deep Learning Technologies. 2017.

83. Tang, Huanrong, et al. "SSD real-time illegal parking detection based on contextual information transmission." CMC-COMPUTERS MATERIALS & CONTINUA 62.1 (2020): 293-307.

84. Ng, Chin-Kit, et al. "Outdoor Illegal Parking Detection System Using Convolutional Neural Network on Raspberry Pi." International Journal of Engineering & Technology 7.3.7 (2018): 17-20.

85. Chen, Weiling, and Chai Kiat Yeo. "Unauthorized parking detection using deep networks at real time." 2019 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE, 2019.

86. Shotton, Jamie, Matthew Johnson, and Roberto Cipolla. "Semantic texton forests for image categorization and segmentation." 2008 IEEE conference on computer vision and pattern recognition. IEEE, 2008.

87. Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

88. Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

89. Zhang, Shanghang, et al. "Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras." Proceedings of the IEEE international conference on computer vision. 2017.

90. Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40.4 (2017): 834-848.

91. Liu, Fangfang, and Ming Fang. "Semantic segmentation of underwater images based on improved Deeplab." Journal of Marine Science and Engineering 8.3 (2020): 188.

92. Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." IEEE transactions on pattern analysis and machine intelligence 39.12 (2017): 2481-2495.

93. Lin, Guosheng, et al. "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

94. Zhao, Hengshuang, et al. "Pyramid scene parsing network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

95. Audebert, Nicolas, Bertrand Le Saux, and Sébastien Lefèvre. "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images." Remote Sensing 9.4 (2017): 368.

96. Poudel, Rudra PK, Stephan Liwicki, and Roberto Cipolla. "Fast-scnn: Fast semantic segmentation network." arXiv preprint arXiv:1902.04502 (2019).

97. Poudel, Rudra PK, et al. "Contextnet: Exploring context and detail for semantic segmentation in real-time." arXiv preprint arXiv:1805.04554 (2018).

98. Zivkovic, Zoran, and Ferdinand Van Der Heijden. "Efficient adaptive density estimation per image pixel for the task of background subtraction." Pattern recognition letters 27.7 (2006): 773-780.

99. Godbehere, Andrew B., Akihiro Matsukawa, and Ken Goldberg. "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation." 2012 American Control Conference (ACC). IEEE, 2012.

100. KaewTraKulPong, Pakorn, and Richard Bowden. "An improved adaptive background mixture model for real-time tracking with shadow detection." Video-based surveillance systems. Springer, Boston, MA, 2002. 135-144.

101. Zivkovic, Zoran, and Ferdinand Van Der Heijden. "Efficient adaptive density estimation per image pixel for the task of background subtraction." Pattern recognition letters 27.7 (2006): 773-780.

102. Wojke, Nicolai, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association metric." 2017 IEEE international conference on image processing (ICIP). IEEE, 2017.

103. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

104. Hassan, Waqas. "VIDEO ANALYTICS FOR SECURITY SYSTEMS". University Of Sussex, 2012.

105. Noble, Frazer K. "Comparison of OpenCV's feature detectors and feature matchers." 2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP). IEEE, 2016.

106. Tian, YingLi, et al. "Robust detection of abandoned and removed objects in complex surveillance videos." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 41.5 (2010): 565-576.

107. Boragno, Simone, et al. "A DSP-based system for the detection of vehicles parked in prohibited areas." 2007 IEEE Conference on Advanced Video and Signal Based Surveillance. IEEE, 2007.

108. Guler, Sadiye, Jason A. Silverstein, and Ian H. Pushee. "Stationary objects in multiple object tracking." 2007 IEEE Conference on Advanced Video and Signal Based Surveillance. IEEE, 2007.

109. Venetianer, Péter L., et al. "Stationary target detection using the objectvideo surveillance system." 2007 IEEE Conference on Advanced Video and Signal Based Surveillance. IEEE, 2007.

110. Porikli, Fatih, Yuri Ivanov, and Tetsuji Haga. "Robust abandoned object detection using dual foregrounds." EURASIP Journal on Advances in Signal Processing 2008 (2007): 1-11.

111. Lee, Jong Taek, et al. "Real-time illegal parking detection in outdoor environments using 1-D transformation." IEEE Transactions on Circuits and Systems for Video Technology19.7 (2009): 1014-1024.

112. Jodoin, Jean-Philippe, Guillaume-Alexandre Bilodeau, and Nicolas Saunier. "Urban tracker: Multiple object tracking in urban mixed traffic." IEEE Winter Conference on Applications of Computer Vision. IEEE, 2014.

113. Marcomini, L. A., and André Luiz Cunha. "A comparison between background modelling methods for vehicle segmentation in highway traffic videos." arXiv preprint arXiv:1810.02835 (2018).

114. Adinanta, Hendra, Edi Kurniawan, and Jalu A. Prakosa. "Physical Distancing Monitoring with Background Subtraction Methods." 2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET). IEEE, 2020.

115. Trnovszký, Tibor, Peter Sýkora, and Róbert Hudec. "Comparison of background subtraction methods on near infra-red spectrum video sequences." Procedia engineering192 (2017): 887-892.

116. Piccardi, Massimo, and Eric Dahai Cheng. "Multi-frame moving object track matching based on an incremental major color spectrum histogram matching algorithm." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops. IEEE, 2005.

117. Fan, Quanfu, Sharath Pankanti, and Lisa Brown. "Long-term object tracking for parked vehicle detection." 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2014.

118. Tola, Engin, Vincent Lepetit, and Pascal Fua. "Daisy: An efficient dense descriptor applied to wide-baseline stereo." IEEE transactions on pattern analysis and machine intelligence 32.5 (2009): 815-830.

119. DeTone, Daniel, Tomasz Malisiewicz, and Andrew Rabinovich. "Superpoint: Self-supervised interest point detection and description." Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018.

120. Luo, Zixin, et al. "Contextdesc: Local descriptor augmentation with cross-modality context." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

121. Mikolajczyk, Krystian, and Cordelia Schmid. "Indexing based on scale invariant interest points." Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. Vol. 1. IEEE, 2001.

122. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

123. H. W. Kuhn, “The Hungarian method for the assignment problem,” Naval Research Logistics Quarterly, vol. 2, pp. 83–97, 1955.

124. Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European conference on computer vision. Springer, Cham, 2014.

125. Rister, Blaine, et al. "A fast and efficient SIFT detector using the mobile GPU." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.