Road User Detection and Analysis in Traffic Surveillance Videos

by

Jinling Li
B.Eng., Beijing University of Posts and Telecommunications, 2012

Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2014

APPROVAL

Name: Jinling Li

Degree: Master of Science

Title of Thesis: Road User Detection and Analysis in Traffic Surveillance Videos

Examining Committee: Dr. Ghassan Hamarneh
Chair

____

Dr. Greg Mori,
Associate Professor, Senior Supervisor

____

Dr. Ze-Nian Li,
Professor, Supervisor

____

Dr. Mark Drew,
Professor, SFU Examiner

Date Approved: August 14, 2014

Abstract

Road user data collection and behaviour analysis has been an active research topic in the last decade. Automated solutions can be achieved based on video analysis with computer vision techniques. In this thesis, we propose a method to estimate traffic objects' locations with state-of-the-art vision features and learning models. Our focus is put on the applications of cyclist's helmet recognition and 3D vehicle localization. With limited human labelling, we adopt a semi-supervised learning process: tri-training with views of shapes and motion flow for vehicle detection. Experiments are conducted in real-world traffic surveillance videos.

Acknowledgments

First and foremost, I want to give my sincere thanks to my supervisor Dr. Greg Mori for his patience, motivation, enthusiasm, and immense knowledge. His encouragement and guidance support me throughout my master study. He is such a wonderful person that his advice is always designed for maximizing our benefits. One simply could not wish for a better, nicer and wiser supervisor.

I would also like to thank my defense committee members: my supervisor Dr. Ze-Nian Li and my thesis examiner Dr. Mark Drew for their helpful suggestions and insightful comments. Both of them are also professors in Vision and Media Lab; I should have taken the chance to talk to them more and learn from their wisdom.

My heartfelt appreciation goes to all the professors I came across at SFU. Dr. Oliver Schulte and Dr. Ke Wang introduce me to the study of *Machine Learning* and *Data Mining* at the first semester when I came to SFU. Dr. Valentine Kabanets gives a great class on *Algorithms* which would help me in many ways in the future (that is if I am going to keep my career in computer science). And I have learned a lot from Dr. Mohamed Hefeeda's course on *Large-scale Multimedia and Cloud Computing*.

I am also deeply grateful for Dr. Tarek Sayed's group at UBC. Most of the projects I have done during my master study are collaborated with them. Thanks for providing related research topic and data and sharing the project with us. My special thank goes to Dr. Mohamed H Zaki for his support and hard work.

And I would like to thank all the members in Vision and Media Lab. All of them have been really nice and helpful. Thanks for all the advice I get from Tian Lan and Guang-Tong Zhou in my research. Thanks Mingjing Zhang for proof-reading my thesis. Thanks Arash Vahdat for always being the cheerful character in the lab. Hossein Hajimirsadeghi, Yasaman Sefidgar and I were the TAs for the course of *Data Structure* in the spring of 2013. It has been a wonderful experience working with them together. Thanks to our former master student, Jianqiao Li, for all the guidance she gave me when I first came to Vancouver. And I am grateful to all my collaborators: Hossein Hajimirsadeghi, Mengyao Zhai and Yuhao Liu for their great work.

Last but not least, I want to express my gratitude to my family. I would not be able to go this far if it were not for my parents. I will love them forever.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Traffic safety analysis requires the collection of data including spatio-temporal properties like the locations of objects, their moving directions, speed and appearance attributes like vehicle size, cyclist's helmet usage, and so on. Traditionally, these data are collected with human counting and measurement which is time consuming and not reliable. Systems automated to some degree have been proposed in recent years.

The application of computer vision techniques to the analysis of traffic videos has more advantages compared to other hardware based methods in terms of traffic data collection and road incident detection. Methods like infrared, radar, and microwave detectors require high installation and maintenance expenses while computer vision based detection can be implemented with existing surveillance videos. Example frames of the video data we use in our experiments are shown in Figure 1.1. In addition, the increasing reliance on traffic video analysis has resulted in a need for more and better traffic information. The state-of-the-art computer vision techniques can be used to provide traffic monitoring in wider respects allowing analysis of traffic flows, speed measurement, vehicle counting, vehicle classification, collision/conflict detection, etc [24].

1.1 Motivation

Traffic safety analysis involves the interpretation of related traffic safety data to support crash data analysis, site safety studies, and local high hazard location analysis. Traditionally, historical collision statistics are used as the data source, but they require a significant number of collisions to be recorded. An alternative approach is to analyze traffic conflicts from a broader perspective [39]. Traffic conflicts are more frequent than collisions, and they provide a way for analysts to immediately observe and evaluate unsafe driving manoeuvres. However, there is a high cost of training

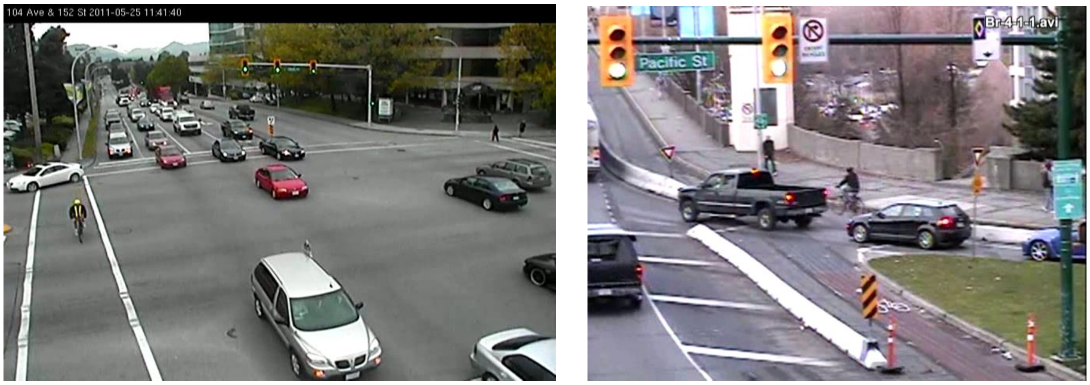


Figure 1.1: Example frames of the datasets we use in our experiments. The videos are taken with real-world surveillance cameras.

observers. And collecting the data is time-consuming. Therefore, a computer vision system to automatically extract conflict information from the video sequences would bring practical benefits to traffic safety analysis.

1.1.1 Cyclist's Helmet Usage Analysis

Cycling is becoming a staple of modern transportation systems. It is an active mode of transportation that enhances the quality of living and adds value in terms of livability and health benefits. This is accomplished by increasing the amount of physical activity, and by reduced congestion, energy conservation, as well as pollution reduction. However, endorsing cycling as a mainstream commuting mode faces several challenges that relate to mobility, efficient resources and more profoundly accepted levels of safety. Safety measures and law enforcement are continuously developed to address cycling safety concerns and to identify factors that may be causing safety deficiencies. Collision prevention programs, including the use of helmets, are being initiated. Such programs are of utter importance given the fact that half of fatal injuries among bicyclists are head injuries.

Safety perception is an important component in the helmet net benefit research. Goldacre and Spiegelhalter [13] suggest that the effect of unmeasured confounding variables makes the real health benefits of helmet wearing more difficult to justify. For instance, cyclists who choose to wear bicycle helmets tend to be more cautious, therefore less likely to be involved in crashes. Safety perception also was suggested as a motivator in reporting near misses. The results from conflicts analysis showed that cyclists always wearing helmets tend to comply with traffic regulations and were more likely to report traffic conflicts. But perception was downplayed by the findings in [49] which suggest that perceived risk does not appear to influence injury rates, nor do injury rates influence perceived risks of cycling. Walker's study [47] observed that drivers tend to allow larger clearance to cyclists without helmets. However, helmets as an injury prevention tool through risk compensation also change the cyclists' behaviour. Increased safety perception can lead cyclists to travel at higher speed which in turn can affect the actual safety as observed in [32, 52, 36]. Given the several facets of the helmet net benefit problem, Robinson [38] suggested a more thorough costs-benefits analysis would be essential for any proposed legislation. In the light of this ongoing debate, there is a census of the importance to collect more helmet usage data.

Collecting reliable data is often labour intensive and time consuming as it is usually collected by manual counts or measurements. In recent years, some degree of automation has been implemented for cyclist data collection [33]. However the information acquired by those methods is limited and it is not always feasible to extract important attributes like helmet usage. In this thesis we present automated computer vision tools to recognize whether the cyclist in the video is wearing helmet or not.

1.1.2 Vehicle Conflict Detection

The use of surrogate safety measures has been advocated as a complementary approach to address road safety measurement issues and to offer more in-depth analysis than relying on accidents statistics alone [20]. One of the most developed methods relies on traffic conflict analysis. Traffic Conflict Techniques (TCTs) involve observing and evaluating the frequency and severity of traffic conflicts at an intersection by a team of trained observers. The concept was first proposed by Perkins and Harris in 1967 [35]. A traffic conflict takes place when two or more road users approach each other in space and time to such an extent that a collision is imminent if their movements remain unchanged [1]. Traffic conflicts are more frequent than traffic collisions. A common theoretical framework ranks all traffic interactions by their severity in a hierarchy, with collisions at the top and undisturbed passages at the bottom [45].

TCTs were shown to produce similar accident frequency estimates compared with accident-based analysis [45]. Traffic conflicts are manually collected by a team of trained observers, either on site or offline through recorded videos. Despite the considerable effort that is put into the development of training methods and the validation of the observers judgment, such data collection is subject to intra- and inter-observer variability. This can compromise the reliability and repeatability of the traffic conflict data. In addition, the training and employment of human observers makes traffic conflict studies costly. The effort for extracting pedestrian and motorist data from videos was deemed immense. This type of data is not only difficult to collect, but also its usefulness is subject to the level of accuracy and precision of the collection process.

Due to the issues and limitations of manual data collection, a growing trend of the use of automated data collection systems has caught on in the field of transportation engineering. In particular, automated video analysis has attracted considerable interest, as video sensors are now widely available (traffic cameras are already installed on many roadways) and inexpensive [45].

1.2 Traffic Analysis in Computer Vision

Automated solutions for data collection can be achieved based on video analysis and computer vision techniques. Data recorded through video sensors is rich in details and captures valuable traffic information. Moreover, the costs associated with recording and storing the data are comparatively low. Certainly, computer vision techniques are not new to the transportation field. In recent years, computer vision has been used to track vehicles and cyclists to study traffic conflicts and their implications on traffic safety [39, 26, 40]. An important benefit of automatic tracking is its ability to capture the natural movement of the objects with high accuracy and consistency while minimizing the risk of disturbing the behaviour of observed road-users.

1.2.1 Cyclist Data Analysis

Traditionally, bicycle safety research relies on collision data analysis as well as factors that contributed to collision such as bicycle helmet usage, cyclist and vehicle driving characteristics among other factors. Other studies have examined safety consequences of proposed treatments and infrastructure designs. Lately safety analysis considered interactions between cyclists and other road-users as an alternative to collision data. Such paradigm shift paved the way to a computer vision automated cyclists-vehicles safety analysis based on conflicts detection.

Bicycles don't have a precise shape and their classification inherits many of the challenges associated with pedestrian classification. A blend of methods based on shape detection has been proposed by Cho et al. [9]. Somasundaram et al. [41] examined the performance of different types of texture-based and motion features for the discrimination between bicycles and pedestrians. They proposed using the distributions of typical velocity along geometric information to identify cyclists. There is a lack of research that addresses the automated identification of riders' helmets in computer vision environment. Chiu et al. [8] proposed a vision-based motorcycle monitoring system to track motorcycles. The system relies on helmet detection to identify partially occluded motorcycles.

1.2.2 Vehicle Data Analysis

Vision-based monitoring of road networks is a complex task. Monitoring intersections faces more problems than highways. These problems are related to the highly variable structure of the intersections, the presence of multiple flows of the vehicles with turning movements, the mixed traffic that ranges from pedestrians to trucks and vehicles that stop at traffic lights. Specific classification and

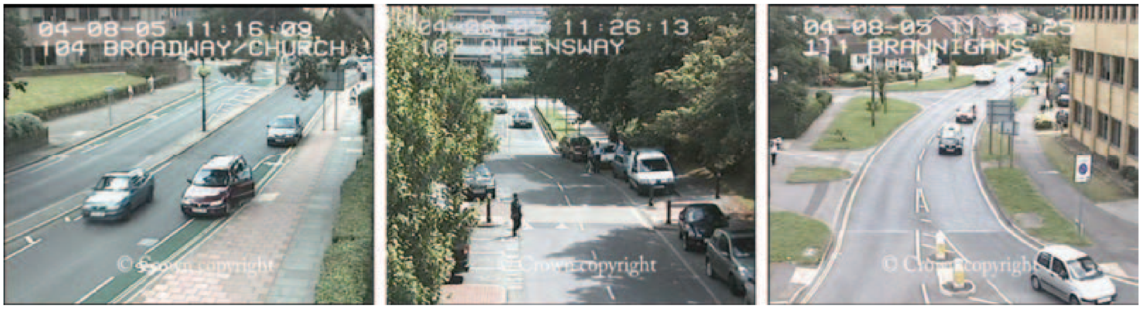


Figure 1.2: Example frames of the i-LIDs parked vehicle detection dataset. Images are taken from [17].

occlusion management techniques are required. Despite the potential benefits of automated traffic safety analysis based on video sensors, limited computer vision research is directly applied to road safety. Saunier and Sayed [39] proposed a system for traffic conflict detection in video data. The system is composed of a feature-based vehicle tracking algorithm adapted for intersections and a traffic conflict detection method based on the clustering of vehicle trajectories.

Estimating the moving direction and speed of the vehicle requires an accurate localization of the object. For automatic incident detection, moving road users must be detected and tracked frame-by-frame. Challenges in this context include occlusions, global illumination variations, multiple object tracking and shadow handling. In addition, traffic surveillance systems usually deal with low camera resolution footage and therefore can only offer a limited amount of visual details of road users. The scenes are usually more constrained than object recognition problems, with cameras mounted on poles stationary above roads [5]. In general, this surveillance task is not as well defined as image retrieval, and no benchmarking challenge has taken place so far. The i-LIDS data set [17] is provided by the U.K. Home Office and it aims at providing a benchmark for surveillance system with a focus on event detection. It covers the following scenarios: sterile-zone monitoring (intrusion detection), parked-vehicle detection, abandoned baggage detection, doorway surveillance, and multiple-camera person tracking. Example frames of the i-LIDS parked vehicle detection dataset is shown in Figure 1.2.

1.3 Contributions

The main contribution of this thesis is the application of computer vision techniques in traffic safety analysis. The goal of this research is (1) to improve the understanding of cyclists' behaviour in order to enhance riding conditions and provide a safe commuting environment (2) to help to locate the vehicles of each frame in world coordinates and use these as the first step of conflict detection.

The videos we use are taken from real-world traffic surveillance cameras. Given videos like this, we want to use computer vision techniques for the automated collection of data about helmet usage during cycling and the positions of the vehicles in the world. These are non-intrusive experiments and can be conducted at a distance. Example frames are shown in Figure 1.1. We tested our algorithms on traffic scenes of busy intersections and roundabouts where collisions and conflicts are likely to happen.

Difficulties in the tasks include (1) vehicles are going in different directions and vary from small cars to big trucks, (2) vehicle-vehicle and vehicle-human occlusions are common in the videos and the traffic is usually busy, and (3) the resolution of the video is relatively low; the area of the cyclist's head is small.

For the cyclist's helmet recognition system, we propose a method to cluster the training data and use selective instances to represent the video sequence of the cyclist's head instead of using all the frame-based descriptors to deal with the outliers in the learning process. For the vehicle localization problem, we use background subtraction on separate lanes to estimate the vehicle's 3D location in the image and its projection on the ground plane in the world based on the geometry layout of the intersection. With limited pre-labelled data, we applied tri-training with views of shape and motion features to learn vehicles' models.

The first application, cyclist's helmet recognition, is published in [28]. Further, the feature data of cyclist's helmet/non-helmet in the video sequences is also used in [15]. My roles in these publications are:

- In [28], I implemented the system with head localization, feature extraction, learning with modified SVM. And I measured the performance of our system with a series of experiments.
- In [15], I was responsible for the feature generation of the head area and I provided the baseline SVM learning results.

1.4 Overview of the Thesis

In this thesis, we propose methods for the application of cyclist's helmet recognition and 3D vehicle localization. The rest of the thesis is organized as follows.

Chapter 2 illustrates the related previous work in traffic object detection and traffic video analysis with computer vision techniques. Chapter 3 introduces the cyclist's helmet recognition system implementation and shows the experiment results with surveillance videos taken from two different locations. Chapter 4 presents our work on tri-training based 3D localization of vehicles in traffic scene and evaluates the performance of detection, tracking and model learning in our framework respectively. Chapter 5 concludes the thesis and provides possible future work.

Chapter 2

Previous Work

Automated traffic video analysis systems usually require image/video processing on each frame with computer vision techniques. Existing applications include vehicle counting, automatic number plate recognition (ANPR), and traffic scene understanding/incident detection [5]. There can be a large variety in incident detection: accident detection, illegal parking detection, and congestion detection to name a few.

A substantial body of previous research exists for traffic scene understanding/incident detection. Buch et al. provide a survey of this work [5]. Analysis of traffic scenes involves a preliminary step of obtaining trajectories of vehicles and other road users. Broadly speaking, this process typically involves an algorithm for detecting vehicles followed by a tracking step that links detections across time, and model learning for recognizing the target object with selected features. We provide a brief sampling of methods from this extensive literature.

2.1 Traffic Object Detection

Traffic object detection aims at the detections of road users to retrieve their positions and other information from the foreground image. It is usually the first step and serves as the basis for object tracking and recognition. There are mainly two different approaches for foreground estimation. One method is to construct a background model to compare with the current frame in order to obtain the motion blobs. This approach requires static cameras while stationary objects will be missing in the detection. Common implementations for this framework are frame differencing and background subtraction. Another approach is to acquire the foreground segmentation based on the objects' appearances or motion flow. Object detections with optical flow, contour template are examples of this method.

2.1.1 Frame Differencing

Calculating the frame difference of adjacent frames is a straightforward way of comparing the pixel values of several images in the time sequence. The moving objects are extracted by setting a threshold for the subtraction results. Frame differencing requires low time and space complexity and can be implemented in real-time. It is less sensitive to illumination changes with few frames in consideration. However, it is hard to get an accurate position of the object with the blur bordering. And it is not able to cope with noise or periodic movements in the background [18].

2.1.2 Background Subtraction

Background subtraction is the most common way of detecting moving objects from static cameras. An image's foreground is obtained by subtracting the background image from the original frame where the background image can be constructed/updated with models built from the video stream. Morphological operators are used to find foreground blobs of the appropriate size, which are determined to be vehicles/pedestrians.

Background construction is critical to foreground detection. An easy way is to use the median/average/mode value of a certain pixel along the video sequence. Another widely used method is to model the intensities of background with single Gaussian or Gaussian mixture models (GMM) [43]. With GMM, each pixel is modelled as a mixture of Gaussian and it is classified to belong with the background group if its distribution is relatively stable.

A substantial body of surveillance work utilizes the approach of background subtraction, addressing challenges in illumination changes, complex backgrounds, and object shadows [18].

2.1.3 Motion/Appearance Based Detection

Motion-based methods are also commonly used for vehicle detection, starting from the seminal work of Beymer et al. [2]. These methods track and group feature points based on motion coherence, and have proven robust across varieties of scenes and road users (e.g. [39]). Alternative bottom-up appearance-based methods include Ma and Grimson [30] that utilizes shape features.

Optical flow, as an example of the motion feature in object detection, is used to describe the motion of the object's surface when there is a relative movement between the camera and the scene. With the calculation of gradients, optical flow can retrieve the information of the object and the environment's structures and blob velocities. Compared to frame differencing and background subtraction methods, motion-based detection can be applied to moving cameras as well. However, the constraint function in building the motion structure is hard to achieve because of occlusion and noise [18].

2.2 Traffic Object Tracking

The goal of the tracking process is to maintain the identity of a moving object over the sequence of frames and measure the object's path. One way of object tracking is to estimate an object's next position by the study of its moving features. Another way of multi-object tracking relies on data association by matching instance labels with temporal observations.

A number of traffic analysis methods exist that use different trackers. Examples include Veeraraghavan et al. [46], who link moving blob-based vehicle detections via a Kalman filter. Kamijo et al. [23] use a spatio-temporal Markov random field approach for obtaining vehicle detections and trajectories.

2.2.1 Model, Region, Contour Based Tracking

Model-based tracking builds appearance models for the target object and associates the models between consecutive frames. The size and moving direction of the object can be obtained with this method but it requires a prior knowledge of the model parameters for every kind of moving target. Region-based tracking identifies connected regions of the image (blobs) that are associated with the object. Regions are often obtained through background subtraction and then are tracked over time. Contour-based tracking uses a representation of the contour of the moving object and updates the contour dynamically. Both region-based and contour-based trackings are sensitive to occlusions and shadows so they cannot be used to deal with congested traffic [5] .

2.2.2 Feature Based Tracking

Feature-based tracking abandons the idea of tracking the object as a whole; instead it tracks local features such as distinguishable points or lines on the object. Feature-based tracking algorithm has distinct advantages over other methods: it is robust to partial occlusions and it does not require any initialization. It can adapt successfully and rapidly to variable lighting conditions allowing real-time processing and tracking of multiple objects in dense traffic. However, feature-based tracking may have difficulty in delineating occluding objects with similar motion vectors. Kanade-Lucas-Tomasi (KLT) tracking [29] is one of the implementations of feature-based tracking methods. The features of one object can be grouped using spatial proximity and motion statistics.

KLT tracker and other feature-based tracking algorithms are proved to be efficient to represent videos. But sometimes it is hard to get good quality and sufficient quantity of trajectories. Dense trajectory [48] is proposed in the inspiration of dense sampling in image classification. Sampled dense points are tracked based on dense optical flow field. These trajectories are proved to be robust to irregular motions. Wang et al. [48] propose a novel descriptor based on motion boundary histograms. An visualization of dense trajectory is shown in Figure 2.1.

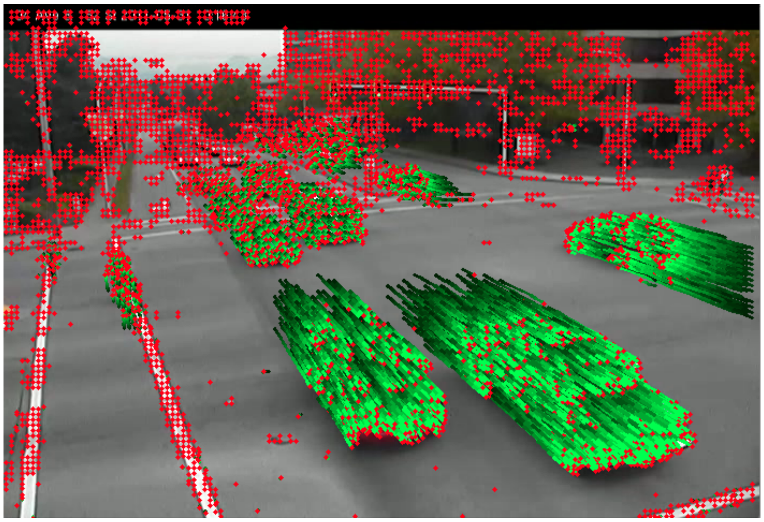


Figure 2.1: Dense trajectories extracted from one example frame in our dataset.

2.2.3 Association Based Tracking

Association based tracking methods usually start by acquiring tracklets with a low-level tracker and then stitch the tracklets by graph-based greedy algorithms including network flow, linear programming and feature matching approach [37]. Pirsiavash et al. [37] propose a way of retrieving the globally-optimal solution for object tracking with an estimation of the number of objects and their births and deaths without manual initializations. Tracking candidates are generated with Hidden Markov Model (HMM) given the detections. A maximum a posteriori (MAP) estimate of the tracks is retrieved with min-cost flow.

2.3 Model Learning

After the steps of detection and tracking, we obtain candidates of the possible objects. Learning method is used to build a general model for each class with descriptors of learning data set. The model can be used in the classification of unseen test data. For example, a classifier is learned in the task of cyclist's helmet detection to recognize whether it is helmet in the head area. And in the experiment of 3D localization of the vehicles, learning is used to retrieve the true positive vehicle detections from all the candidates.

2.3.1 Object Features

Traditional learning methods train models with features that represent the object class. Some of the features we use in our experiments are: Histogram of Oriented Gradients (HOG) [10] for shape, and Texton [31] and Local Binary Patterns (LBP) [34] for texture.

Object shapes usually generate strong changes in image intensities. Histograms of Oriented Gradients (HOG) [10] descriptor represents the appearance and shape of one local object based on image gradients computed around the target region. It first divides the image into small cells. Then the gradients of each pixel in the cell are computed and aggregated into a histogram to represent the cell. Finally the descriptor of the whole image is obtained as the concatenation of these per-cell histograms. Because of the use of local image gradients and normalization, HOG tends to be robust to illumination and geometric changes.

Texture is a measure of the intensity variation of a surface which quantifies properties such as smoothness and regularity. It is also less sensitive to illumination changes compared to colour. We represent texture via the texton histogram [31] and Local Binary Pattern [34] approaches.

The term of texton was proposed in 1981 [22] for describing human textural perception. In general, textons are defined as a set of blobs or prominent patterns sharing a common property all over the image. Subsequently a model of spatial filtering with orientation and scale-selective mechanisms based on texton theory became popular. Malik et al. [31] introduced a computational approach for textons that performs vector clustering on the outputs of image filters to find prototypes (textons). The images can be analyzed into texton channels by constructing a universal texton vocabulary and using K-means clustering.

Local Binary Pattern (LBP) [34] is another texture descriptor which measures and extracts texture information from gray scale images. First, it calculates the correlation between one pixel value and its surrounding pixel values. Then it forms the LBP code based on certain weight distribution rules. One simple calculation procedure is as follows. To a certain pixel $g_c = f(x_c, y_c)$, define a texture as the joint distribution of the gray levels of 8 pixels in the 3×3 window surrounding g_c, $T = t(g_c, g_0, ..., g_7)$. The binary value of the neighbour pixels in one window $g_0, ..., g_7$ are calculated using the value of $g_c = f(x_c, y_c)$ as a threshold.

$$T \approx t\left(s\left(g_0 - g_c\right), ..., s\left(g_7 - g_c\right)\right), s(x) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases} \tag{2.1}$$

Then group these eight binary values together we have a 8-digit binary number, encoding the texture for this window, which could also be represented as a decimal value, namely the LBP code value.

$$LBP(x_c, y_c) = \Sigma_{i=0}^{7} s\left(g_i - g_c\right) 2^i. \tag{2.2}$$

2.3.2 Object and Scene Models

Shape model mapping is a straightforward way of comparing the object with the target class. Buch et al. [4] propose a 3D wire model mapping method for the recognition of objects in i-LIDS vehicle dataset. They build 3D closed contour models for five categories of vehicles including bus/lorry, van, car/taxi, motorbike/bicycle and pedestrian respectively. However the orientation of vehicles has to be constant to match the models. The approach proposed by Song and Nevatia [42] uses vehicle shape models to find rectangles in the image which are likely to include a vehicle by maximizing a posterior probability (MAP). The size of the rectangle has to be calculated with the knowledge of the ground plane, vehicle type (sedan, SUV, and truck) and camera geometry. Jia and Zhang [21] present a front-view vehicle detection method with high-way traffic surveillance videos. They construct the model with edge information of the front-view vehicles and similar with [42], they also achieve detection and segmentation with MAP.

Vehicle as an object category has been well studied in object recognition. It has large variability of models and the view-point makes the appearance change dramatically. In situations other than traffic scenes, different methods have been applied to learn the viewpoint in order to have better performance on car detection. The series of work from Savarese and Fei-Fei [44] represents cars as a coherent ensemble of parts that are consistent under 3D viewpoint transformations by learning a model that captures the relative position of parts within each viewpoint. Gu and Ren [14] use a mixture of shape templates and a discriminative learning for joint viewpoint classification. Kuo and Nevatia [25] propose a method of grouping the examples with divide-and-conquer strategy. They use sub-categorization and cascade tree classifier instead of supervised learning. Carr et al. [6] achieve monocular object detection by employing geometric primitives as proxies for the true 3D shape of objects. It uses geometric primitives with occupancy maps to form spatially-changing-kernel models. Hoiem et al. [16] propose a strategy to model the scale and location variance of the objects in the image by modelling the relations between objects, surface orientations and camera viewpoints.

On the other hand, traffic scene layout has a critical effect on traffic video analysis. Geiger et al. [12] present a probabilistic model for traffic scene understanding from movable platforms. It infers scene topology, lane geometry and the location and orientation of objects by using monocular features (vehicle tracklet, vanishing point, scene labels) and stereo features (occupancy grid and scene flow).

2.3.3 Semi-supervised Learning

In the problem of vehicle detection, it is hard to label the vehicles in every frame. With a few human annotation beforehand, we decide to apply semi-supervised learning in the training program.

In semi-supervised learning, test data is supposed to help to improve the retrieved classifier.

In the assumption that the classifier's own high confidence predictions are correct, one can apply a self-learning by adding the most confident features and predicted labels to the training set and re-train the classifier and repeat. Self-training is the simplest semi-supervised learning method, however the early mistakes in the procedure can reinforce themselves in the following rounds [54]. Co-training [3] is proposed in order to improve the robustness of the 'add-in' data. Each instance is represented by two sets of feature descriptors and two classifiers are trained based on the two sets of features separately. Add one's most confident test results to the other classifier training set and repeat. Co-training is less sensitive to mistakes, but it requires natural-split (conditionally independent) features to describe the instance which are probably hard to get. Levin et al. [27] apply a co-training framework in a visual detection system and acquire higher detection rates. In order to avoid the feature split problem, a multiview-training is presented by training multiple classifiers of different types and classify unlabelled data with all classifiers. The data will be added into the training set by majority vote. Zhou and Li [53] prove that tri-training algorithm can be used in the application of web page classification.

In this thesis, we are going to use background subtraction to get initial traffic object detections. KLT tracking is applied to complement the detection candidates. Cyclist tracks are picked out with an analysis of the object's speed profile. Colour-histogram, HOG, texton and LBP features are extracted from the head region to train a classifier between helmet and non-helmet. In the task for 3D vehicle localization, we perform tri-training with a few pre-labelled vehicle examples to retrieve the true positive vehicle boxes from the candidates.

Chapter 3

Cyclist's Helmet Recognition

In this chapter, we introduce a method for cyclist's helmet recognition using computer vision features and machine learning classifiers. Cyclist-involved traffic safety analysis requires large amounts of cyclist's data collection. While traditional human counting and measurement is time-consuming and not practical, we use computer vision techniques to retrieve the information like helmet usage. Automated cyclist's helmet recognition system can be used for safety analysis of traffic videos and to provide related statistics for future law enforcement.

Figure 3.1 illustrates the helmet detection framework. First a tracker obtains the tracks of the moving objects in the video. During video recording, the three-dimensional real-world is captured on a two-dimensional image space. Camera calibration is the process of determining the homography matrix of a camera angle. A homography matrix is used to map the world coordinates to image plane coordinates [19]. Based on the analysis of the objects trajectories and speed profiles, only cyclist trajectories are identified and kept for further data analysis [51]. Second, we obtain the cyclists head location based on background subtraction. Third, image features are extracted from the head region. Four features are considered: colour-histogram, Histograms of Oriented Gradients (HOG) [10], texton [31] and Local Binary Pattern (LBP) [34]. Finally, we train a classifier using supervised learning to classify helmet and non-helmet cyclists. Since we are not able to correctly locate the head area in every frame, many features we extract are actually outliers in classification. We propose a method of using only top-ranked instances to rule out these outliers.

3.1 Tracking of the Moving Objects

Automated data collection of road user trajectories is carried out using a computer vision system described in [39]. The automated analysis relies on algorithms to differentiate between features of road users and features that are part of the environment based on the Kanade-Lucas-Tomasi (KLT)

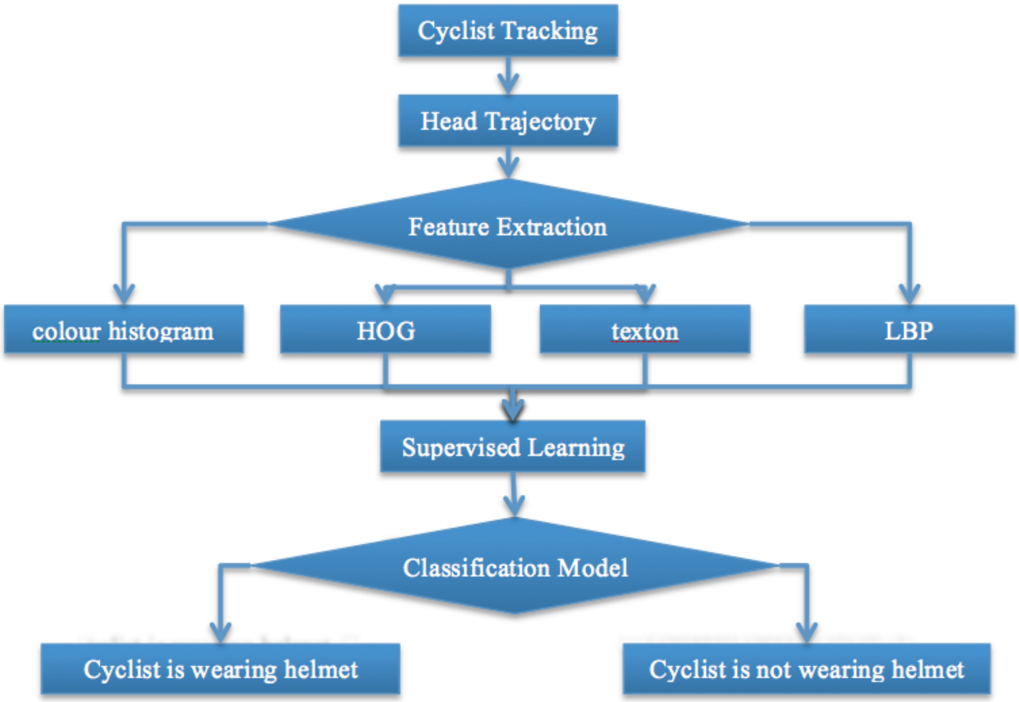


Figure 3.1: Methodology Framework for Cyclist's Helmet Recognition

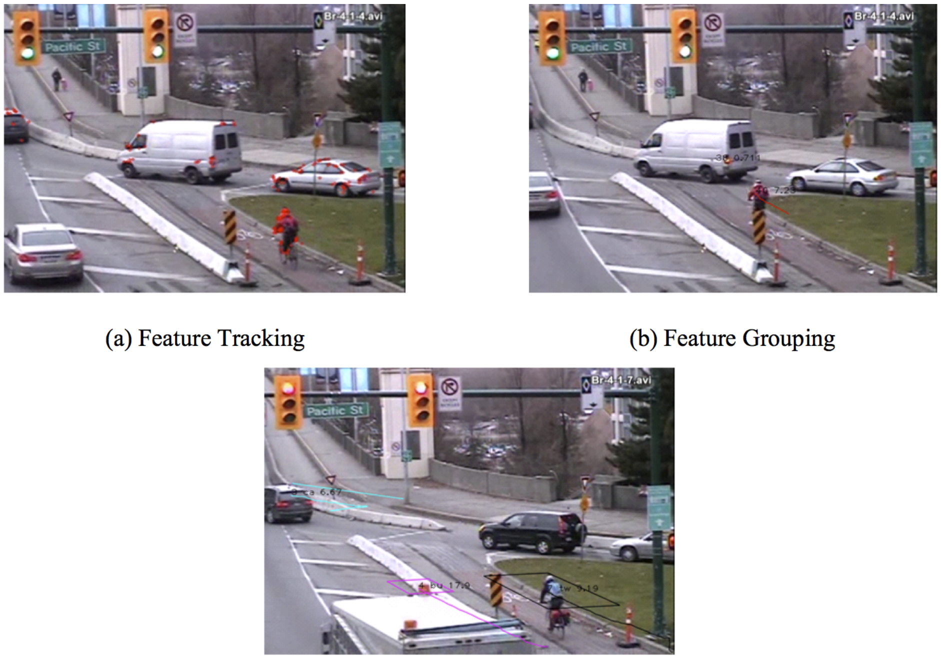
tracking algorithm. Figures 3.2.a demonstrates feature tracking for a cyclist. Since a road-user can have multiple features, a follow up to the tracking step is the grouping of features, i.e. deciding what set of features belongs to the same object. The grouping suggests the trajectory of a moving object. The main cues for grouping are spatial proximity and dynamical similarities. Figure 3.2.b provides an illustration for feature grouping.

The subsequent step in the analysis is the road-users classification. The road user trajectories have features that reveal the structures of the traffic scene and provide important clues of the characteristics of the objects. A regular movement pattern of a cyclist is described by its pedalling with the main attributes being cadence. The speed profile of a cyclist typically fluctuates periodically at each cadence. A cyclist speed profile will show periodic variations which are usually at lower repetitions than normal pedestrians walking frequencies. In contrast, vehicle movement patterns are primarily composed of linear segments. This oscillatory behaviour associated with pedestrians and cyclists provides a road-user classification basis. Other features like maximum speed, object size are used as complimentary queues to enhance the classification [51]. Figure 3.2.c provides road-user classification illustrations.

During video recording, the three-dimensional real-world is captured on a two-dimensional image space. The transformation between two coordinates is associated with the properties of the camera and its lens. Camera calibration is the process of determining the homography matrix of a camera angle, and is necessary for convert these tracks back into their positions in real-world. Each calibration process begins with the user annotating features in the camera image and in an aerial, orthographic image of the intersection. Details of the adopted mixed-feature camera calibration approach are presented in [19].

3.2 Localization of the Cyclist's Head

After getting cyclists' tracks, there is a need to focus on the head area. In order to obtain the trajectories of the head, we move our tracking points in every frame to the top of the cyclist. First, we use background subtraction to get a cyclist silhouette. Assume the original coordinate of the tracking point is $[x, y]$, then we will search from $(x - \Delta x)$ to $(x + \Delta x)$ to find the top-most point $[x', y']$, whose pixel value in the silhouette image is larger than a threshold. We move the tracking point to $[x', y']$ and draw a bounding box of size $(h_x \times h_y)$ with $[x', y']$ as top-middle point. Here we set Δx to be the roughly estimated width of the cyclist. We use camera calibration to calculate the size of the head area (h_x, h_y) as well as the width of the cyclist Δx. The process is shown in Figure 3.3.



(a) Feature Tracking

(b) Feature Grouping

(c) Road-user Classification

Figure 3.2: Demonstration of the Pre-processing in Cyclist's Helmet Recognition: Steps to Obtain Cyclists' Tracks.

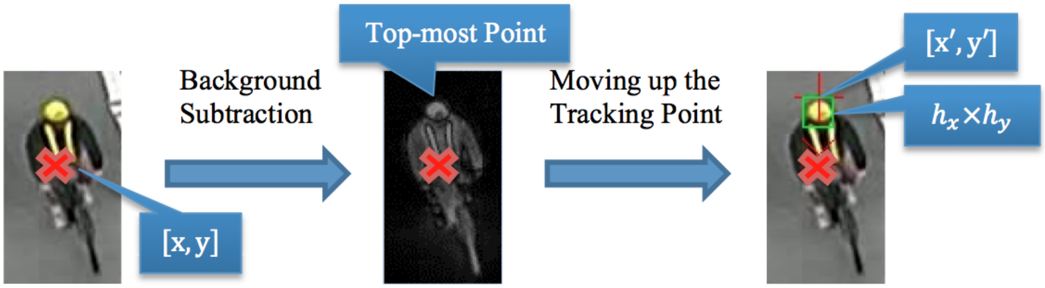


Figure 3.3: Move the tracking point to the top-most point of the cyclist based on background subtraction. Red x indicates the original tracking point $[x, y]$, red + indicates the location of the head $[x', y']$ and green square is the bounding box $(h_x \times h_y)$ where we extract feature representations.

3.3 Object Representation: Image Colour, Shape, Texture feature

Given the cyclist's head position, the next step is to extract features that can be used to determine if the cyclist is wearing a helmet. Objects can be represented by their shape and appearance. Selecting the right features plays a critical role in the performance of the classification.

We use features that represent colour, shape and texture: a colour histogram, Histogram of Oriented Gradients (HOG) [10] for shape, and texton [31] and Local Binary Patterns (LBP) [34] for texture.

- Colour: Colour is one of the features we use to distinguish between helmet and non-helmet. Most of the helmets are colourful while the hair is usually darker. We use a primitive colour representation by building an RGB colour histogram with 11 bins in each channel to represent the colour of the head region in each frame

- Shape: The round shape of the helmet is another distinctive character. Object shapes usually generate strong changes in image intensities. We use the Histograms of Oriented Gradients (HOG) [10] descriptor to represent the shape of the head area. In our experiments, the bounding box around the head is firstly resized to 40×40 pixels and then HOG descriptors are densely extracted on a regular grid at steps of 8 pixels.

- Texture: The texture of the helmet is also a distinguishing feature. Texture is a measure of the intensity variation of a surface which quantifies properties such as smoothness and regularity. It is also less sensitive to illumination changes compared to colour. We represent texture via the texton histogram [31] and Local Binary Pattern [34] approaches.

  The images can be analyzed into texton channels by constructing a universal texton vocabulary and using K-means clustering [31]. In our experiments, the texton vocabulary contains 128 entries, which are the responses of one filter. Local Binary Pattern (LBP) [34] is another texture descriptor which measures and extracts texture information from gray scale images. It calculates the correlation between one pixel value and its surrounding pixel values. In our experiments, the texture in LBP refers to 8 sampling points on the circle with radius 1 around the pixel. A uniform mapping of 59 patterns is used to generate a 59-dimension LBP histogram.

3.4 Supervised Learning: Training a Model for Classification

Once the features that discriminate one class from the other are selected, we use supervised learning to learn a classifier that can determine whether an individual cyclist is wearing a helmet or not. In our experiments, we use Support Vector Machines (SVMs) [7] as our classifier.

In this classification task, we have a training set and a test set for the helmet detection data. Each instance in the training set has two variables. One is the feature descriptor of the head area; the other is the class label (i.e. whether the cyclist is wearing a helmet or not). Based on these two variable values for the examples in the training set, SVM will produce a model to predict the class label of the test data given only the features descriptors of the test data.

3.4.1 Choosing Top-Score Instances in Learning

Since we are not able to accurately move the tracking point to the head area in every frame of the video because of the inaccuracy of background subtraction, sometimes the bounding boxes where we extract features from are still on the body of the cyclist, on the bike or on the street (see Figure 3.4).

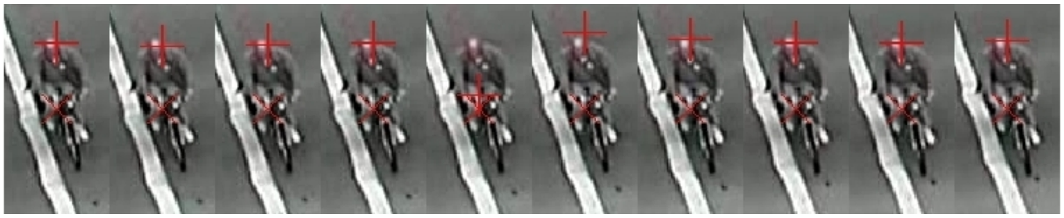


Figure 3.4: Ten consecutive frames from a video in the Surrey dataset. Red x indicates the original tracking point, red + indicates the location of the head. We are not able to move the tracking point to the top of the head in every frame.

The accuracy is higher when most of the bounding boxes are correctly located. To handle the situation of incorrectly located bounding boxes, we propose a method of using top-rank-score instances in each iteration of training. These top-rank-score instances are supposed to be the feature representatives extracted from bounding boxes around the head. But chances are that not all top-rank-score instances are from the head group. In order to further filter these instances, we first use mean-shift to cluster all the examples, with respect to their colour, HOG, texton and LBP feature descriptors, into several groups and then calculate the mode of the group number that those top-rank-score instances belong to. We use only the instances that belong to this group in the top-ranked set in training and test of the learning process (See Figure 3.5). Results and performance measurements of our method can be found in Section 3.5.

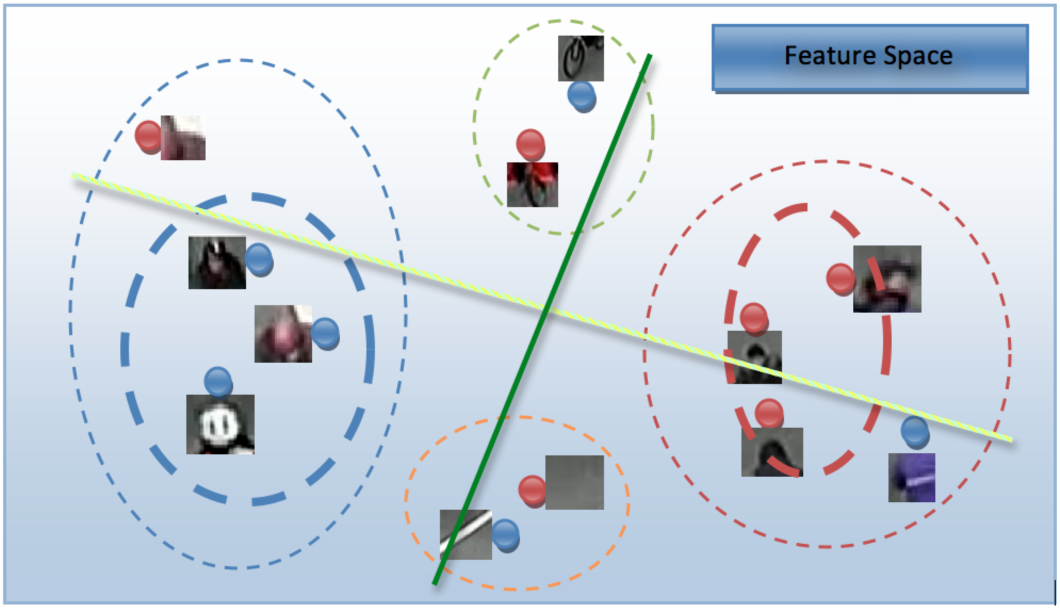


Figure 3.5: Choose top-rank-score instances that are in the correct clusters in learning process. The blue dots in the figure indicate the positive instances while the red dots indicate the negative instances. The thin circles are the results of cluster and the thick ones include the top-rank-score instances in the cluster. The yellow line is a possible linear classification model trained with all the examples in this feature space. The green line is the classifier learned with only examples inside of the thick circles. By ruling out outliers, we would be able to obtain more accurate and robust classifiers.

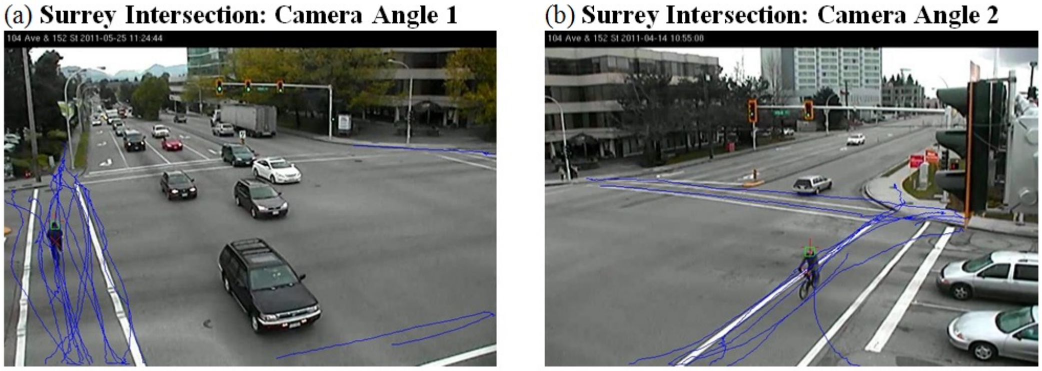
3.5 Datasets and Experiments

In the experiment of cyclist's helmet recognition, we use two groups of transportation videos from two different locations in Surrey and Vancouver in B.C. Canada.

3.5.1 Surrey Dataset

The data is collected from a busy 4-legged intersection. Two camera positions capture the different segments of the intersection as shown in Figure 3.6(a) and Figure 3.6(b). Out of two days of videos, our experiments datasets are selected during different time periods when the traffic is mixed with pedestrians, cyclists and vehicles with varying speeds. Data collection dates are consistent with typical traffic data collection standards (typical weekdays).

For the Surrey dataset, we select the videos with relatively good tracks of the cyclists' heads, which refer to the videos that the heads are correctly located in more than 50% of the frames. In


(a) Surrey Intersection: Camera Angle 1

(b) Surrey Intersection: Camera Angle 2

104 Ave & 152 St 2011-05-25 11:24:44
104 Ave & 152 St 2011-04-14 10:55:08

Figure 3.6: 4-Legged Intersection in Surrey, BC. Blue lines show the tracks of the cyclists. Red × indicates one example frame of the original track. Red + and the green box indicate the head's location.

order to see the result of classification, we choose 12 clips of videos where cyclists are wearing helmets and 12 clips of videos where they are not. We test the classification results with colour-histogram, HOG, texton and LBP features. Leave-one-out cross validation is used to get the average classification accuracy. The selecting top-rank-score and top-rank-score in the correct cluster instances in learning methods results are shown in Table 3.1.

Table 3.1: Leave-one-out average accuracy results of colour-histogram, HOG, texton and LBP features respectively in classification of the Surrey dataset.

	Colour	**HOG**	**texton**	**LBP**
Original Accuracy	0.74	0.79	0.74	0.65
Top Rank Score Selection	0.87	0.85	0.89	0.77
Top Score in the Cluster Selection	0.87	0.92	0.96	0.83

Even though our method can help to rule out some of the outliers in classification, the performance is still mostly related to the quality of the track. Videos with correct location of the head in almost every frame are usually correctly classified. On the other hand, the video that fails to be classified correctly in most of the experiments is the video with poorest head trajectories. In fact, in that video, the number of tracking points located on the head is smaller than the number of them located elsewhere, and even in the cluster that we choose in our method, these two numbers are also roughly the same. Some of the other commonly failed cases include a video where the cyclist is wearing a black helmet and a video where the cluster of images of the ground has the highest score in optimization. Some of the frames of these cases are shown in Figure 3.7. One thing to note is that due to the limitation of the accuracy of the original KLT tracker, we have situations like multiple tracks on the same cyclist or no track is detected on a certain cyclist. According to [28], with KLT tracker, we are able to obtain 90% of the tracks of the cyclists in the videos.

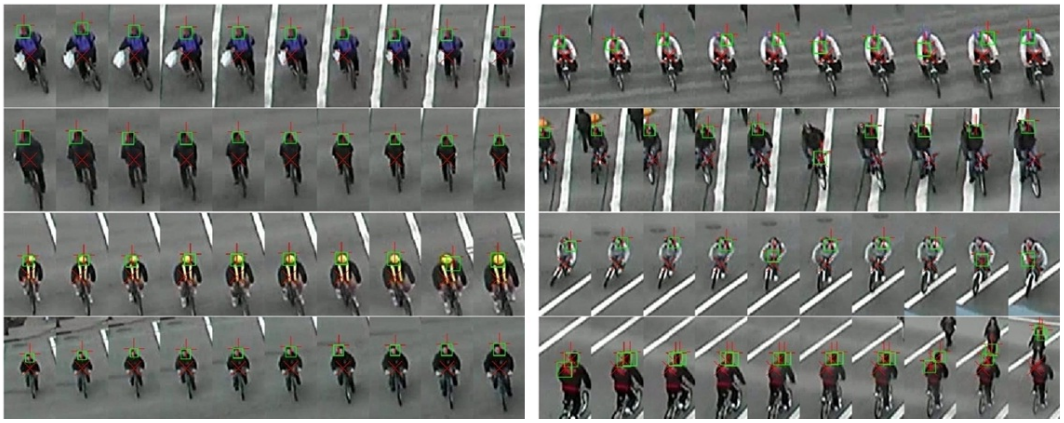


Figure 3.7: These samples are taken every five frames from videos in the Surrey dataset. The group of four videos on the left has good track of the head and can be classified correctly with all of the four features. Videos on the right are the most common incorrect classification cases. The frames in the second line on the right refer to the cyclist wearing black helmet case and the frames in the third line have the worst tracking among all 24 videos. The last video on the right has multiple tracks for one cyclist.

3.5.2 Vancouver Dataset

The videos in this dataset are of the Burrard Bridge Ramps at Pacific Street. The intersection analyzed in this study is selected because of perceived high rate of conflicts between vehicles and cyclists as well safety concerns related to merging vehicle conflicts. Right-turning vehicles in the ramp should yield for bicycles travelling southbound to the Burrard Street Bridge. Although right-turning vehicles should yield for the southbound vehicle through traffic, the current configuration limits the available sight distance leading to severe merging conflicts. Data collection dates are consistent with typical traffic data collection standards (see Figure 3.8).

Compared with the Surrey dataset, we have longer videos and more cyclists in the videos in the Vancouver dataset. The overall statistics can be found in Table 3.2. In total, we have 6 videos; each is one hour long. There are 34 tracks of cyclists in each video on average and 207 tracks of cyclists in total. Among the 207 objects, there are 22 negative cases where the cyclists are not wearing helmets. Since the camera in this dataset is farther away from the cyclists and there is usually a turn in cyclist's track (as seen in Figure 3.8), we have 55 bad track cases and 28 cases where

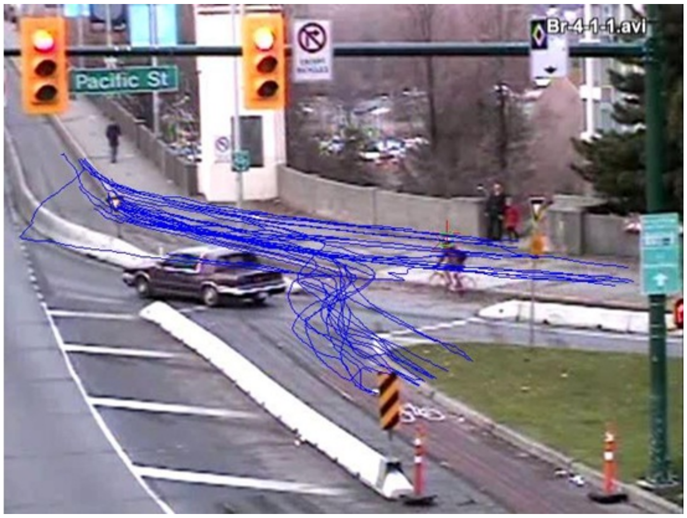


Figure 3.8: Burrard Bridge Northern Gate at Pacific Street. Blue lines show the tracks of the cyclists. Red × indicates one example frame of the original track. Red + and the green box indicate the track of head.

whether the cyclist is wearing helmet is hard to tell. Here bad track cases refer to the situations where the head is correctly located in less than 30% of the frames. We use the first 4 videos as training data and test on the two other videos. There are 110 total objects for test, where 99 are labelled as positive and 11 are labelled as negative. We did experiments on the whole dataset as well as selected parts of dataset with/without some of the above cases.

Table 3.2: Statistics of the Vancouver dataset

	Video1	**Video2**	**Video3**	**Video4**	**Video5**	**Video6**	**Total**
Number of Objects	37	17	13	30	39	71	207
Positive+Negative	35+2	16+1	11+2	24+6	33+6	66+5	185+22
Number of Objects With Bad Tracks	5	2	2	8	16	22	55
Number of Objects With Can't Tell Labels	4	1	3	5	6	9	28
	Training Videos				**Test Videos**		

In the process of detecting all the cyclists that are not wearing helmets from videos like this, we are facing two main difficulties.

First, there is unbalanced number of positive and negative instances. In the survey in [9], around

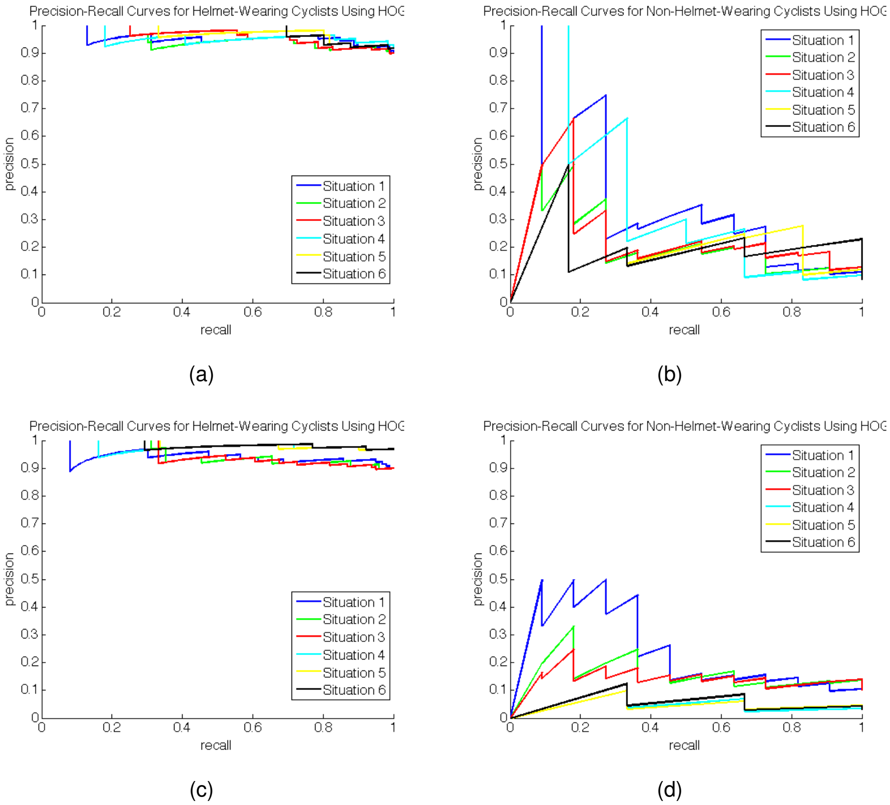
90% of the cyclists are wearing helmets in typical scenarios. And this number corresponds to our statistics in this dataset. In this case, the majority class (people wearing helmets) is represented by a large portion of all the examples while the minority class (people not wearing helmets) has only a small percentage of all the examples. The highly unbalanced class distribution usually results in poor performances from standard classification algorithms which generate classifiers that maximize the overall classification accuracy. The final classifier with these learning algorithms usually completely ignores the minority class or overfits the training examples even with rescaling to rebalance training examples.

Second, the camera settings in this dataset are farther away from the cyclist and the tracks are usually not straight. Compared to the Surrey dataset, the head area of the cyclist is much smaller. It is hard even for human eyes to distinguish between helmet and bare head and it is harder to label the cyclists when they are wearing hats or large backpacks. In addition, since there is a turn in the cycling path, the tracks are less stable and the location of the head is less accurate. And instead of only having the front and back looks of the helmet/non-helmet, we usually have side looks appearing in these videos, but the looks in these views are less distinguishable.

Since the data is highly unbalanced, using accuracy as a measure of the performance would not be a good idea. By classifying all data as the positive class we would still have around 90% accuracy but it also means we are not able to pick out any negative instance. Here we use precision-recall graphs as the measurement and use HOG feature representations since they have better performance in the Surrey dataset.

In the context of classification, the precision for a class is the number of true positives (i.e. the number of items correctly labelled as belonging to the positive class) divided by the total number of elements labelled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labelled as belonging to the class). Recall is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items that were not labelled as belonging to the positive class but should have been). To draw a precision-recall curve, we are going to need a group of true and false positives. With supervised learning, we have the model for classification and we can obtain the scores for each of the instance in test with this model. By setting a threshold for the score, we would be able to classify these instances as positive or negative cases. Compare these results with the ground truth labels we will have the true positive and false positive numbers for each of the threshold we set. With different thresholds and different true and false positive rates as a result, we are able to draw the precision-recall curves. Usually, precision and recall scores are not discussed in isolation. Instead, the value for one measure is evaluated for a fixed level at the other measure. For example, we want to see what the precision is when the recall level is 1.

Inverse precision and recall are simply the precision and recall of the inverse problem where positive and negative labels are exchanged. Since we are more interested in the cyclists who are not wearing helmets, it is essential that we also analyze the precision-recall curves for the negative case (inverse precision and recall). The Precision-Recall curves generated for both positive and negative cases are shown in Figure 3.9.



(a)

(b)

(c)

(d)

Figure 3.9: Precision-Recall Curves in Experiments on the Vancouver Dataset. For the first row of figures, we use the subset of better tracking examples in the learning process as comparison. Situation1: train and test with all the instances; Situation2: train and test with top20 score instances in each example sequence; Situation3: train and test with top score instances in the chosen cluster of each example sequence; Situation4: train and test with selected good track example sequences; Situation5: train and test with top20 score instances from the good track example sequences; Situation6: train and test with top score instances in the chosen cluster from the good track example sequences. In the second row of figures, we use selected correctly labelled example sequences in Situation 4, 5 and 6 as comparison.

From the curves in Figure 3.9, we can have the following conclusions. In general, testing with good track examples has better performance than testing all data for the positive case. To detect all the people who are not wearing helmets, we only need to check half of the original data based on the classification results. Since it is easier to tell that a cyclist truly is wearing helmet, reducing all the ”can't tell” label instances means reducing many potential negative cases and resulting in 82 positive and 2 negative cases in training and 92 positive and 3 negative cases in test. Short of negative examples, we are not able to train a good classifier to classify between helmet and non-helmet. It is even harder to pick out cyclists who are not wearing helmets from all the videos.

In this dataset, there are more videos with bad track of the cyclist's head. Similar with the result analysis for the Surrey dataset, true positive and true negative videos in classification are also usually the ones with high percentage of correctly located head. But there are more varieties of the false positive and false negative cases. Most of the cyclists of ”can't tell” labels are not correctly classified. In false positive cases, we have videos with cyclists wearing hats or big backpacks. And in false negative cases, most of the cyclists are farther away from the camera and have rather small head area. In addition, we do have false positive and false negative cases where the track is relatively good but the cluster chosen by our method consists of bounding boxes from areas other than the head. We believe this can be overcome by adjusting the mean shift cluster parameters or using better cluster methods. Some of the sample frames of these situations are shown in Figure 3.10.

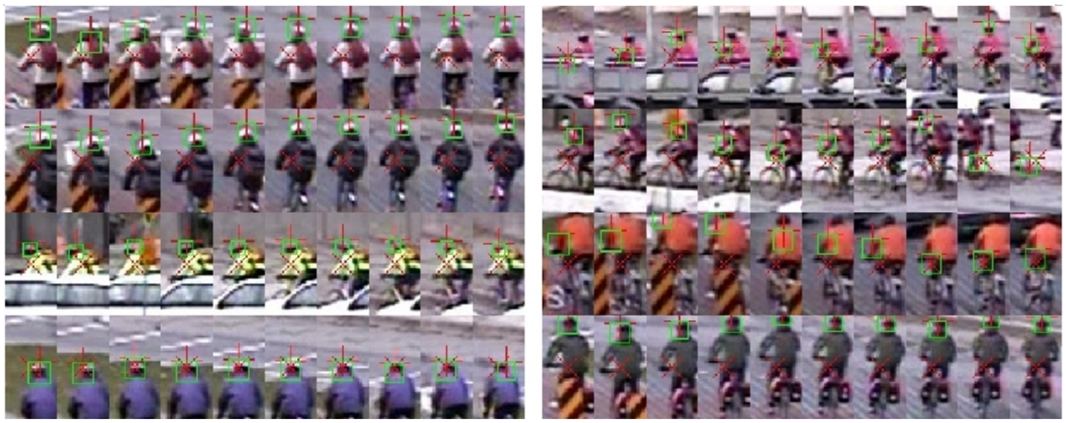


Figure 3.10: These samples are taken every five frames from the videos in the Vancouver dataset. The group of four videos on the left has good tracks of the heads. The first two rows of videos have similar distinguishable features as those in the Surrey dataset and they have high scores as true positives. But we only have side looks of the helmet in the third video and small helmet area in the fourth video. They are correctly classified in only some of the experiments. Videos on the right are the most common incorrectly classified cases. The first three rows of videos have poor tracks of the heads because of occlusions, tracker on the bike and tracker on the side respectively. The last row represents a good track but it is hard to tell whether the cyclist is wearing helmet or not even with human eyes (here we give it a positive label as wearing helmet).

Chapter 4

Tri-Training Based 3D Vehicle Localization

In this chapter we describe our work on vehicle localization using a 3D cuboid representation of the vehicle and using tri-training as the learning method. Conflict detection requires the information of the positions of the objects in every frame of the video. Point or axis-aligned rectangle representations are not able to identify the front and rear end positions of the objects. Here we build a 3D model for the vehicle based on the lane geometry to obtain its projection on the ground and its position in the real world coordinate.

With limited human labour and pre-labelled data, we use a semi-supervised learning method, tri-training, to learn the classifier that retrieve the vehicle boxes from the candidates generated with background subtraction detection and KLT tracking. In the tri-training process, each instance is described with three views that are independent with each other. A classifier is learned from each view's descriptors of the training set and a final decision on an instance from the test set is made by integrating the classification results from each view. By adding some of the most confidently classified data from test set into the training set, we hope to improve the classification performance by having more labelled examples. Figure 4.1 shows an intuitive understanding of the tri-training strategy.

4.1 Object Localization and Tracking

Traffic surveillance cameras are usually calibrated upon setting up. The calibration parameters provide a bridge between image and world coordinate systems. We construct the layout of the road based on the lane geometry in real world. Given this information, we are able to deal with some occlusion problems, calculate the moving directions of a certain detected object in the image,

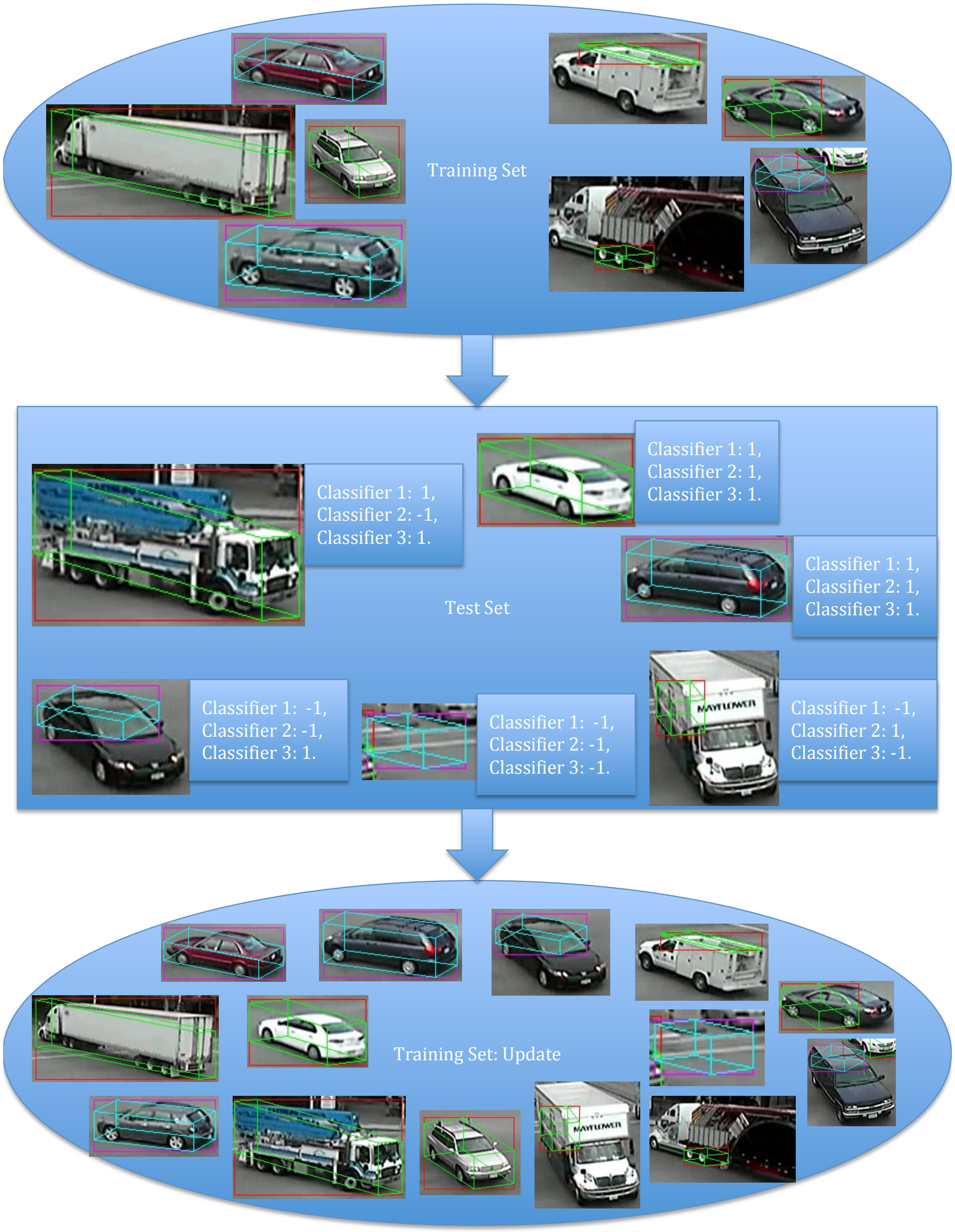


Figure 4.1: Overview of the Tri-training Learning Process.

and acquire a 3D location with the knowledge of the orientation of the vehicle. Compared to 2D axis-aligned rectangle representation, the 3D model helps us to find accurate interest points on the object and build up a better tracker to find missing vehicles from background subtraction detection.

4.1.1 2D Vehicle Localization

Same as most of the related work in traffic videos [4, 42, 21], we first use background subtraction to estimate the locations of the moving objects. The median values of each pixel over the video sequence are used to represent the background. Unbroken contours are detected in the background subtraction image via morphological operations. A rectangle is drawn around each contour as the 2D bounding box of the object.

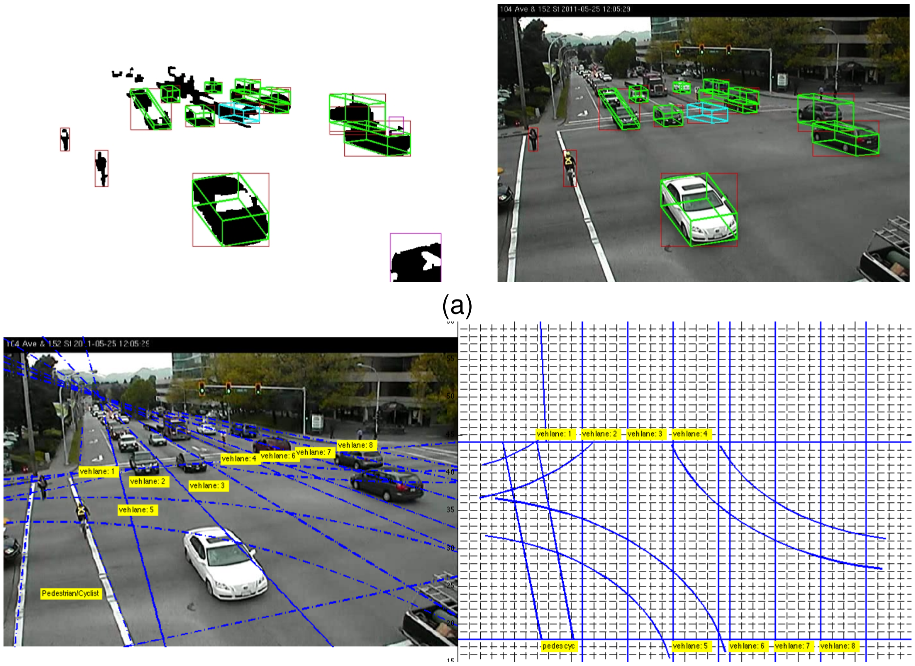
Inter-vehicle and vehicle-person occlusions are common problems in traffic scene analysis where the images or videos are taken from cameras placed on poles or tall buildings. Background subtraction with contour detection is not able to separate image blobs of multiple occluded objects. Here we do background subtraction on each lane separately and on the whole image to differentiate road users in adjacent lanes. Figure 4.2 illustrates the background subtraction result based on our method.

4.1.2 3D Vehicle Localization

Given the 2D vehicle detections, we lift them to a 3D representation. Figure 4.3 shows an example of generating the 3D box based on the 2D rectangle position. Assume the coordinate of the centre of the 2D rectangle in the frame is (μ_0, ν_0), we first map it to the world coordinate to get (x_0, y_0). Based on the moving direction of the vehicle, we are able to get the world coordinates of the four points that are one unit(meter) away from (x_0, y_0): (x_1, y_1) to (x_4, y_4) with the following equations. If the vehicle is turning, we assume the turning path is part of a circle and we calculate (x_1, y_1) to (x_4, y_4) based on the centre and radius of the circle.

$$\begin{aligned} &x_1 = x_0 - sin(\theta_0),\ y_1 = y_0 + cos(\theta_0);\ \ x_2 = x_0 + cos(\theta_0),\ y_2 = y_0 + sin(\theta_0); \\ &x_3 = x_0 + sin(\theta_0),\ y_3 = y_0 - cos(\theta_0);\ \ x_4 = x_0 - cos(\theta_0),\ y_4 = y_0 - sin(\theta_0); \\ &if\ straight : \theta_0 = 0;\ if\ turning : \theta_0 = atan(y_0 - centre_y, x_0 - centre_x) \end{aligned} \tag{4.1}$$

We then map each of the four points back to the image coordinates and obtain (μ_1, ν_1) to (μ_4, ν_4). By ignoring perspective effects, we are able to estimate the orientation of the vehicle in the image. We use $(\alpha,\ \beta)$ to represent the angle between the front and side edge of the vehicle and μ axis respectively. The width of the car in the image Δw and $(\alpha,\ \beta)$ can be calculated with the four coordinates assuming that the width of the vehicle is two meters in real world. In addition, we have the 2D to 3D bounding box transition equations as follows where Δl and Δh are the length and



(a)

(b)

Figure 4.2: (a) Result of background subtraction on a sample frame and visualizations of the 3D vehicle bounding boxes used in our paper. Green boxes indicate vehicles travelling straight while blue ones are referring to turning vehicles. (b) Lane geometry layout in the image and world coordinates. We calculate the foreground image blobs on each lane separately which helps us to deal with occlusions of adjacent vehicles like the ones on the top right corner in (a). But when the vehicles are far from the camera, it is still hard to distinguish one from another with background subtraction detection.

height of the vehicle in the image:

$$\begin{aligned} \Delta l \times cos\beta + \Delta w \times cos\alpha &= rect\_width \\ \Delta h + \Delta l \times sin\beta + \Delta w \times sin\alpha &= rect\_height \end{aligned} \tag{4.2}$$

When the road user is in the overlap area of several possible lanes, we create one 3D model for each lane layout. These 3D location candidates will be verified in the learning process.

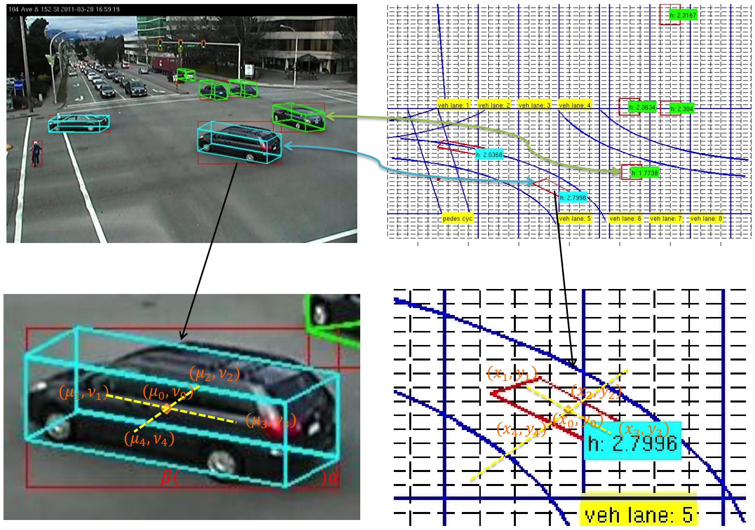


Figure 4.3: Calculate the 3D bounding box based on the 2D rectangle. By mapping the five coordinates shown on the two figures at the bottom back and forth in the image and world coordinates, we obtain the moving angles α and β as well as the eight coordinates of the 3D box in the frame.

4.1.3 Extended Detections by KLT Tracking

Since we are not able to correctly locate the position of the vehicle in every frame of the video because of occlusions, shadows, incorrect background calculation etc., tracking can help to retrieve the bounding boxes that are not obtained with background subtraction detection. With a 3D localization of the vehicles, the features used for tracking can be more precisely acquired. For example, in the left image of Figure 4.4, because of the viewing angle of the camera, the truck is occluded by the car besides it. Features extracted from the 3D box will be more accurate than those extracted from the 2D rectangle.

After getting the corner features of three angles, we use the Kanade-Lucas-Tomasi (KLT) algorithm [29] to track the moving objects. As seen in Figure 4.4, the corner features extracted from the vehicle cover descriptors of the license plate, windshield, lights and others that are representative of the object. The tracking points are mostly valid when there is a clear view of the vehicle.

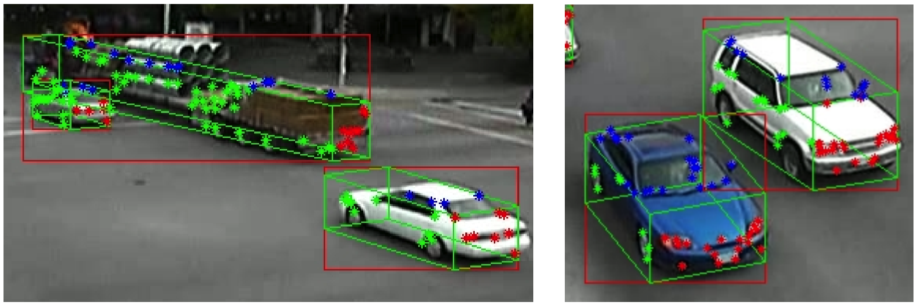


Figure 4.4: Corner Features in Tracking. Points with colour red, green, blue represent features from front/back, side and top angles of the vehicle respectively. Extracting features from the 3D model helps to rule out features extracted from points that are not on the object vehicle but are inside of the 2D rectangle box.

Each background subtraction-based detection is extended temporally using this approach. This results in a set of vehicle detections that has relatively high precision and recall, obtaining the majority of vehicles present in each frame (quantitative results appear in Section 4.3).

4.2 Tri-training and Model Learning

A learning model is used to obtain the *True Positive* boxes from the candidates. In traditional supervised learning, numerous labelled data is required to learn a valid model. However we are trying to achieve vehicle detection automatically to reduce human resources while labelling usually requires a lot of human work. With few labelled examples and large amounts of unlabelled samples, semi-supervised learning is chosen in the training process.

Given a bounding box of the moving object, one first notices its components; A vehicle consists of lights, wheels, windshields, license plates, mirrors and other features. People use bag-of-words model to simulate this process. On the other hand, bag-of-words model lacks the information of the relationship between these components. For example, the license plate and lights should be in the front while wheels should be on the side of the vehicle. An overall shape can help to make sure the components are in the correct location. In addition, some of the false positives share similar shape descriptors with the vehicles, a motion flow feature can be used as a third view to exclude them.

The false positives we have include bounding boxes on the background, boxes of part of the vehicles and boxes with incorrect orientations, which can be ruled out with motion feature, template shape feature and shape feature on BoW model respectively. These three complementary views provide three descriptors for the object and naturally lead us to a tri-training process.

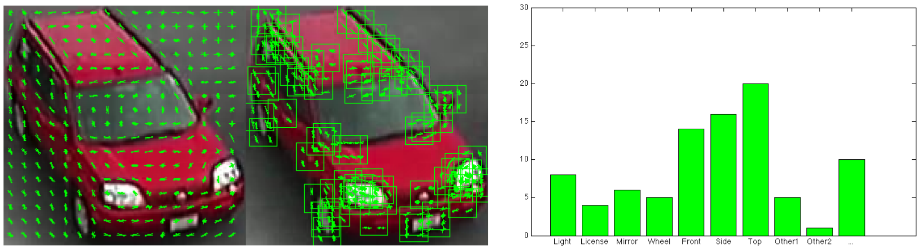


Figure 4.5: Visualization of HOG and Bag-of-Words features. On the left are visualizations of template HOG feature extracted from the 2D rectangle around the vehicle and the local features extracted from interest points. On the right is an intuitive calculation of the bag-of-words model.

4.2.1 Latent SVM in Learning

We implement tri-training with a margin-based classifier: Support Vector Machine (SVM) and we use the margin as a confidence measure for predicting unlabelled examples. We use regularized support vector classification C-SVC to obtain the decision scores.

Since the vehicles in the scene are moving in different directions and there is a big difference between the shape of big trucks and small vehicles, the extracted features can actually form various classification models. The size and orientation of the vehicle can be used to divide the instances and fill in different latent variable values. In this sense, we build a latent SVM model [11] in the learning phase:

$$\begin{aligned} &\min_{w} \frac{1}{2}\|w\|^2 + C_1 \sum_{i=1}^{N_1} \xi_1 + C_2 \sum_{i=1}^{N_2} \xi_2 \\ where\ \ &\xi_1 = \max(0, 1 - y_i w^T \phi(x_i, h_i)) \\ &\xi_2 = \sum_{h \in H(x_i)} \max(0, 1 - y_i w^T \phi(x_i, h)) \end{aligned} \tag{4.3}$$

ξ_1 and N_1 are the positive deduction and positive instance number while ξ_2 and N_2 are referring to the negative case. $\phi(x_i, h_i)$ is acquired with inference $h_i = \arg\max_{h \in H(x_i)} w^T \phi(x_i, h)$. Since the number of instances in each latent category h can be highly unbalanced, here we scale the value of ξ_i by adjusting the parameter C_1 to make sure different h is treated equally and the number of times of adding them into the update of optimization is the same.

4.2.2 Tri-training Process

Tri-training is a semi-supervised learning process that is modified on top of co-training. In the same spirit of adding some of the confidently classified data from the test set into the training set, we hope to improve the learning model by having more labelled data. Each instance is described with three views that are conditionally independent with each other. Classifiers C_1 C_2 and C_3 are learned from three views' descriptors of the training set and a final decision on an instance from the test set is made by integrating the classification results from each view. A pseudo-code of the tri-training process is listed in Algorithm 1.

Algorithm 1: Tri-training Model Learning Process

```
Data: $L_0$: Pre-Labelled examples, $U_0$: Unlabelled examples
$L$: Current Training Examples, $U$: Current Unlabelled Examples;
Initialization $L = L_0, U = U_0$;
for Pass Number = 1:20 (run for 20 rounds at most) do
    Classifier $C_1$ $C_2$ and $C_3$ are trained with three different views of features on $L$;
    Apply $C_1$ $C_2$ and $C_3$ on $U$, obtain decision scores $D_1$ $D_2$ and $D_3$;
    Calculate decision scores on $U$, $D = D_1 + D_2 + D_3$;
    SetP: The set of examples that are classified as positive by at least two of the three
    classifiers;
    SetN: The set of examples that are classified as negative by at least two of the three
    classifiers;
    if SetP is not empty then
        NewP = the examples in SetP whose $D$ are confidently large;
        L = L+NewP;
        U = U-NewP;
    end
    if SetN is not empty then
        NewN = the examples in SetN whose $D$ are confidently small;
        L = L+NewN;
        U = U-NewN;
    end
    if SetP and SetN are both empty then
        Break;
    end
end
```

When classifier C_1 C_2 and C_3 are trained with latent SVM model, we initialized the instances with 4 latent variable values. We can also use the model with separate classifiers. For example, C_1 is composed of C_{11} and C_{12} where C_{11} is learned and applied to vehicles travelling straight only and C_{12} is for turning vehicles only. Using the separate classifiers requires initial knowledge of the lane information while the latent SVM strategy can be applied to other cases when vehicles have different moving orientations and sizes.

Evaluations of the background subtraction detection, extended detection with KLT tracking and our learning method for vehicle localization in traffic surveillance videos are shown in the following section.

4.3 Datasets and Experiments

In the experiment of 3D vehicle localization, we choose four video clips taken from the same intersection in Surrey (details can be found in Section 3.5.1). The videos are 2 minutes and 30 seconds long with 30 frames per second and include approximately 250 different vehicles, 4 cyclists, and 6 pedestrians. The size of each frame is 704 by 480 pixels and the average size for a passenger car is 60 by 40 pixels. The vehicles in the videos include various sizes of trucks, vans, small cars, etc. On average, there are 10 moving objects in one frame and 4 in the intersection area.

Unlike the work from Geiger et al. [12] that infers the layout and topology of the scene by learning, we manually identify the lane geometry. We have 9 pre-defined lanes in which there are 3 vehicle turning lanes and one lane for cyclists and pedestrians. The turning lanes are defined as parts of circles. Vehicles in Video 1 and Video 3 are mostly travelling straight, while vehicles in Video 2 are mostly turning. And there is a large variety of vehicle types in Video 3. Vehicles in Video 4 cover all the travelling lanes.

We manually label 50 different frames (including around 40 different vehicles) from Video 1 and Video 2 as the set of pre-labelled examples. Among these examples, there are 3 trucks, 37 normal-sized cars; 15 turning vehicles on lane 5 and 25 vehicles on the straight lanes. Negative examples come from the same 50 frames where the bounding boxes are obtained in background subtraction and tracking but are not the same as the manually labelled vehicles.

4.3.1 Evaluation on BGS and Tracking

For background subtraction, we use the median value of the frame pixels in each video as the background. After obtaining the 3D model of an object based on the methods described in Section 4.1, we extracted corner features and track 150 frames before and after the target frame. For vehicles that are coming towards the camera, we only track them backwards of the video sequence.

The precision and recall evaluations after these two steps can be found in the first two lines in Table 4.1 and Table 4.2. A major goal of our method is to accurately locate the vehicles in the real world. Here we compare the output 3D bounding boxes of each step to the ground-truth 3D bounding boxes by their locations in the world coordinate (i.e. the objects' projections on the ground plane). The ground-truth bounding box is manually labelled every 5 frames in the video. A detection of a vehicle bounding box is considered as a *True Positive* if the ratio of the intersection area to the union area is larger than 0.4. Tracking helps to obtain around 92% of all the vehicles in the intersection area. Our method can help to separate vehicles in separate lanes, but since the vehicles in most of the frames are adjacent to each other, most of the mis-detected vehicles are due to partial occlusions.

Method	Video1	Video2	Video3	Video4
BGS	0.93	0.85	0.87	0.87
KLT Tracking	0.96	0.88	0.92	0.90
Base Model with Feat1 and Feat2	0.85	0.70	0.73	0.69
Base Model with Feat1 and Feat3	0.85	0.75	0.77	0.68
Base Model with Feat2 and Feat3	0.84	0.76	0.70	0.72
Base Model with Feat1, Feat2 and Feat3	0.92	0.80	0.80	0.77
Co-training Model with Feat1 and Feat2	0.86	0.70	0.74	0.72
Co-training Model with Feat1 and Feat3	0.90	0.82	0.81	0.81
Co-training Model with Feat2 and Feat3	0.90	0.80	0.78	0.78
Tri-training Model	0.95	0.84	0.82	0.81
Final Output	0.96	0.87	0.86	0.84

Table 4.1: Recall Evaluations in 3D Vehicle Localization. Feat1, Feat2 and Feat3 are referring to template HOG, HOG on bag-of-words model and motion feature on bag-of-words model respectively. Base models are learned with manually pre-labelled examples (50 frames of objects).

4.3.2 Evaluation on Co-training and Tri-training

Since cars usually have strong edges in their appearance, we use Histogram of Oriented Gradients (HOG) [10] as the feature. We extract HOG with block size 2×2, cell size 8×8 and 9 orientation bins. For the bag-of-words model, we regularly sample the points on front, side and top angles of the 3D box and cluster them into 100 visual words to construct our code book.

For each iteration of the co-training and tri-training process, we use latent SVM [50] to learn the classifiers and calculate the decision scores. We perform a non-maximum suppression after thresholding the scores for the final vehicle detection. If two vehicles overlap in the world coordinate, the one with higher score will be selected. We also track the positive output objects of the learned classifier on ± 5 frames with KLT tracking to fix some of the mis-classified vehicles in the end.

For an one-minute video clip, it usually takes around one hour for the vehicle detection generating steps including background subtraction and KLT tracking, and another one-hour for the tri-training process using MATLAB on a machine with two 3.0GHz proessors.

We draw the precision-recall curves for test results of multiview learning classifier and base classifier for the four videos (shown in Figure 4.6). As seen from the figures, tri-training procedure helps to improve the learning performance in our experiments. The false positive instances in the generated candidates are mainly in three cases: (1) bounding boxes of the background; (2) bounding boxes of part of the vehicle and (3) bounding boxes with incorrect orientation of the vehicle. While template HOG features help to rule out the first two cases, motion and HOG features on BoW model are distinguishable in the case 1 and case 3. Three features compensate with each other and provide more reliable test results for the iterations of the multiview learning process.

Method	Video1	Video2	Video3	Video4
BGS	0.81	0.71	0.63	0.68
KLT Tracking	0.68	0.62	0.56	0.60
Base Model with Feat1 and Feat2	0.83	0.82	0.67	0.78
Base Model with Feat1 and Feat3	0.87	0.84	0.71	0.81
Base Model with Feat2 and Feat3	0.85	0.85	0.68	0.77
Base Model with Feat1, Feat2 and Feat3	0.91	0.86	0.76	0.77
Co-training Model with Feat1 and Feat2	0.85	0.83	0.68	0.80
Co-training Model with Feat1 and Feat3	0.88	0.83	0.74	0.84
Co-training Model with Feat2 and Feat3	0.87	0.91	0.72	0.81
Tri-training Model	0.92	0.89	0.84	0.86
Final Output	0.92	0.88	0.85	0.86

Table 4.2: Precision Evaluations in 3D Vehicle Localization. Feat1, Feat2 and Feat3 are referring to template HOG, HOG on bag-of-words model and motion feature on bag-of-words model respectively. Base models are learned with manually pre-labelled examples (50 frames of objects).

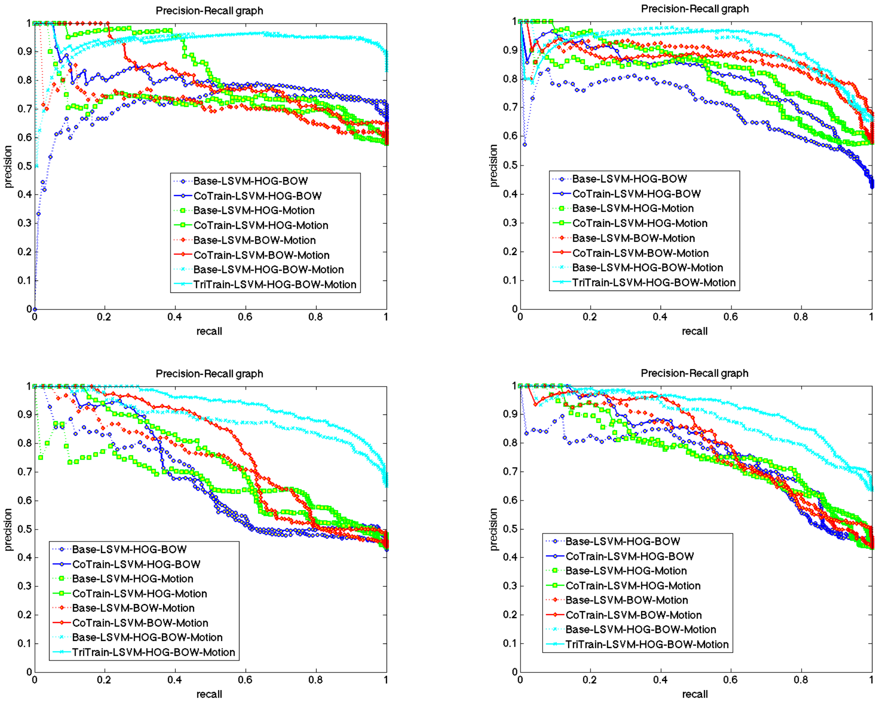


Figure 4.6: Precision-Recall Curves for Co-training and Tri-Training Classification Results. The figures (from left to right, from top to down) are referring to Video 1 to Video 4 respectively. In the legend, HOG is template HOG feature; BOW is local HOG features on bag-of-words model and Motion is Histogram of Optical Flow(HOF) feature as described in [48].

The evaluations (precision and recall numbers) of our learning method and the final frame-based 3D vehicle detection are also listed in Table 4.1 and Table 4.2. A visualization of the localization results of background subtraction, KLT tracking, Co-training, and Co-training with KLT fixation is shown in Figure 4.7. Most of the background subtraction detections are correct except for the adjacent vehicles in the same lane and vehicles across different lanes. KLT tracking helps to extend the number of vehicle candidates while it also produces *False Positives*. Multiview learning extracts *True Positives* from all the candidates and a final tracklet makes minor refinements to improve the accuracy of localization.

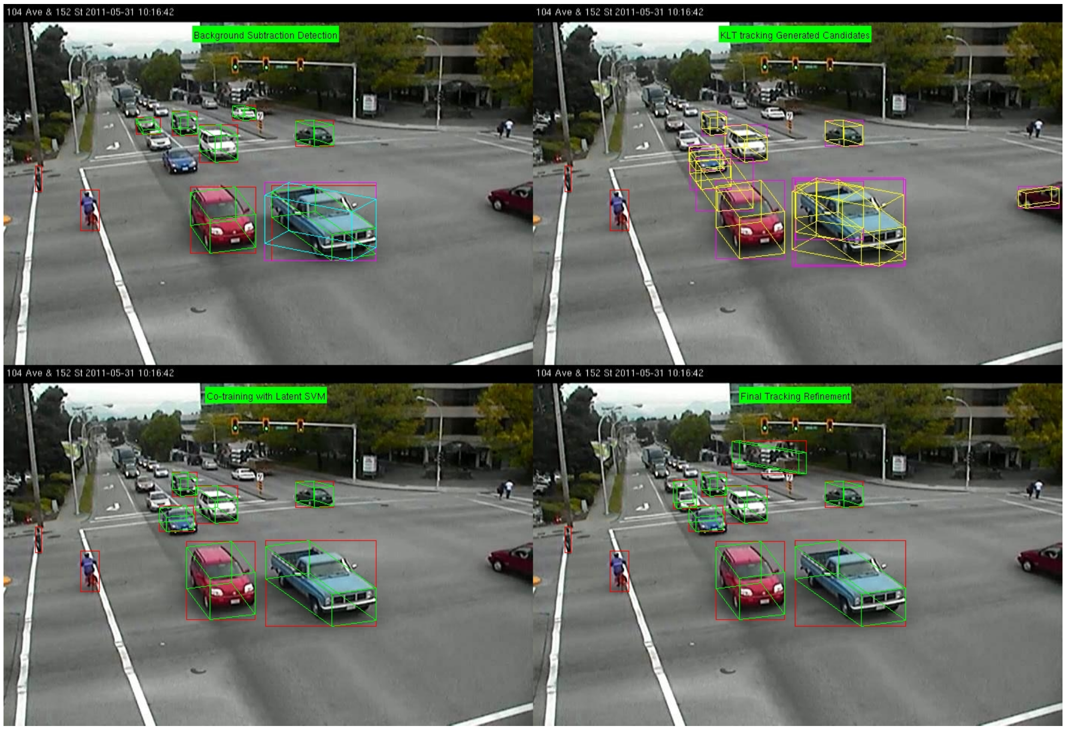

104 Ave & 152 St 2011-05-31 10:16:42
Background Subtraction Detection
104 Ave & 152 St 2011-05-31 10:16:42
KLT tracking Generated Candidates
104 Ave & 152 St 2011-05-31 10:16:42
Co-training with Latent SVM
104 Ave & 152 St 2011-05-31 10:16:42
Final Tracking Refinement

Figure 4.7: 3D localization results after background subtraction (top left), KLT tracking (top right), co-training (down left) and final refinement (down right).

In order to compare the classifier after tri-training with the base classifier, we modify the number of pre-labelled vehicle examples from 5 to 40 in step of 5. F_1 measurement $F = \frac{2*Precision*Recall}{Precision+Recall}$ results can be found in Figure 4.8.

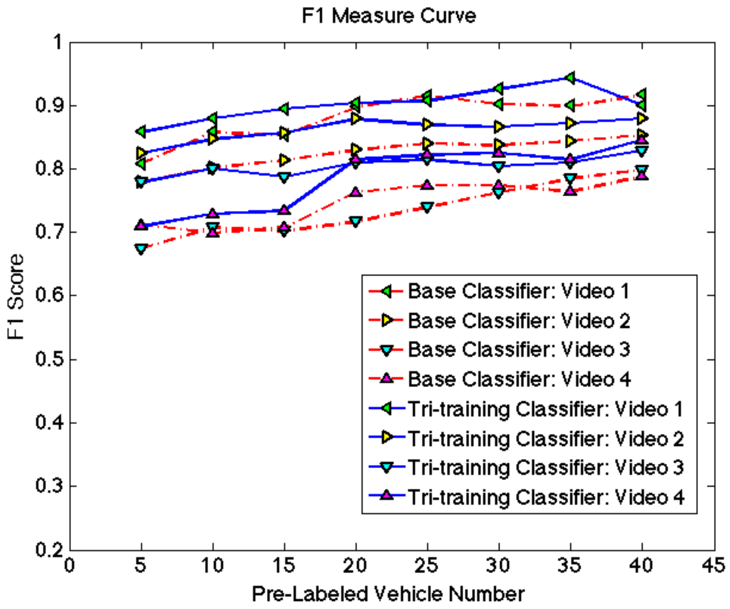


Figure 4.8: F_1 Measure Curves for Different Pre-labelled Example Numbers. Base classifier is trained with pre-labelled instances only and tri-training classifiers are trained with tri-training procedure.

There is not much difference between F1 measurements of the base classifier and tri-training classifier for the test in Video 1. This is because the vehicles in Video 1 are mostly normal sized cars and it is one of the two videos where most of the instances are pre-labelled by human. On the contrary, the performance of multiview learning classifier is much better than base classifier (especially when the training example numbers are little) for Video 3 where there are more varieties of vehicle types. Multiview training helps to recognize vehicles that are abnormal than the manually labelled objects by adding them into the training set within the learning process. An average F-measure improvement using tri-training on pure test set (Video 3 and Video 4) is 0.07. As we can see from the figure, tri-training model can be applied to fewer pre-labelled example cases while obtaining similar accuracy.

Chapter 5

Conclusions and Future Work

This thesis includes our work in two applications: cyclist's helmet recognition and 3D vehicle localization.

The first part of our work shows that it is possible to use computer vision and machine learning algorithms in the application of detecting cyclists who are not wearing helmets. We developed a system that utilizes automated road user detection based on feature tracking. Cyclist's tracks are collected, and head regions are automatically extracted. A set of features ranging from colour to shape to texture is extracted from this region. Supervised learning is used to classify cyclists as wearing helmets or not. Results on two datasets show the method is promising: good classification results can be obtained when tracker performance is reliable, and the amount of manual labour needed to find all helmetless cyclists can be significantly reduced by using our system.

One limitation of this method is that high accuracy of helmet/non-helmet recognition requires good cyclist tracking and correct localization of the cyclist's head. And the distance between the cyclist and the surveillance camera should be within certain range in order to obtain a clear view of the cyclist's head in the frame. The main directions for future work include better tracking of the cyclist's head to help in model learning and the exploration of additional features for helmet wearing classification.

In the second application, we proposed a method of obtaining accurate 3D locations of the moving objects in heavy traffic scenes based on lane geometry information. Compared to a 2D rectangle, a 3D box provides more reliable feature points. Another contribution of this work is that we use tri-training on an overall shape of the vehicle, local features on bag-of-words model, and dense trajectory features three views. While the bag-of-words view shows what components the object has, an overall shape makes sure the components are at the correct place. In addition, the motion feature complements the shape pattern descriptor with an optical flow descriptor. Based on an accurate localization of the vehicle, future applications include car collision detection, conflict detection, vehicle classification, vehicle tracking etc.

Limitations of this method include its requirement of pre-defined lane geometry and homography matrix. The homography matrix is applied to the coordinate transformations between world and image coordinates. The transformation is usually not perfect which results in some deviations of the calculated positions. We use the pre-defined lane layout to generate background subtraction detection and calculate the orientation of the vehicle in each lane. It requires extra human work to define the lanes and the partition of the lanes sometimes brings error to the algorithm. Vehicles not going in the correct moving direction of that lane, vehicles changing lanes, vehicles swinging wide around the corner, etc. will be mis-detected with our strategy.

Bibliography

[1] F Amundsen and C Hyden. The swedish traffic conflict technique. In *Proceedings of First Workshop on Traffic Conflicts, Institute of Transport Economics, Oslo*, pages 1–5, 1977. 3

[2] David Beymer, Philip McLauchlan, Benjamin Coifman, and Jitendra Malik. A real-time computer vision system for measuring traffic parameters. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 495–501. IEEE, 1997. 8

[3] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, pages 92 – 100, 1998. 13

[4] Norbert Buch, James Orwell, and Sergio A Velastin. Urban road user detection and classification using 3d wire frame models. *IET Computer Vision*, 4(2):105–116, 2010. 12, 29

[5] Norbert Buch, Sergio A Velastin, and James Orwell. A review of computer vision techniques for the analysis of urban traffic. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):920–939, 2011. 5, 7, 9

[6] Peter Carr, Yaser Sheikh, and Iain Matthews. Monocular object detection using 3d geometric primitives. In *The European Conference on Computer Vision (ECCV)*, pages 864–878. Springer, 2012. 12

[7] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011. 18

[8] Chung-Cheng Chiu, Min-Yu Ku, and Hung-Tsung Chen. Motorcycle detection and tracking system with occlusion segmentation. In *Eighth International Workshop on Image Analysis for Multimedia Interactive Services*, pages 32–32. IEEE, 2007. 4

[9] Hyunggi Cho, Paul E Rybski, and Wende Zhang. Vision-based bicycle detection and tracking using a deformable part model and an ekf algorithm. In *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1875–1880. IEEE, 2010. 4, 23

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005. 11, 14, 18, 36

[11] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9):1627–1645, 2010. 33

[12] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3d traffic scene understanding from movable platforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(5):1012–1025, 2013. 12, 35

[13] Ben Goldacre and David Spiegelhalter. Bicycle helmets and the law. *BMJ*, 346(7912), 2013. 2

[14] Chunhui Gu and Xiaofeng Ren. Discriminative mixture-of-templates for viewpoint classification. In *The European Conference on Computer Vision (ECCV)*, pages 408–421. Springer, 2010. 12

[15] Hossein Hajimirsadeghi, Jinling Li, Greg Mori, Mohamed H Zaki, and Tarek Sayed. Multiple instance learning by discriminative training of markov networks. In *29TH Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013. 6

[16] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008. 12

[17] Home Office Scientific Development Branch. Imagery library for intelligent detection systems (i-LIDS). https://www.gov.uk/imagery-library-for-intelligent-detection-systems, 2007. 5

[18] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334–352, 2004. 8

[19] Karim Ismail, Tarek Sayed, Nicolas Saunier, and Michael Bartlett. A methodology for precise camera calibration for data collection applications in urban traffic scenes. *Canadian Journal of Civil Engineering*, 40(1):57–67, 2013. 14, 16

[20] Karim Ismail, Tarek Sayed, Nicolas Saunier, and Clark Lim. Automated analysis of pedestrian-vehicle conflicts using video data. *Transportation Research Record: Journal of the Transportation Research Board*, 2140(1):44–54, 2009. 3

[21] Yangqing Jia and Changshui Zhang. Front-view vehicle detection by markov chain monte carlo method. *Pattern Recognition*, 42(3):313–321, 2009. 12, 29

[22] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, 1981. 11

[23] Shunsuke Kamijo, Katsushi Ikeuchi, and Masao Sakauchi. Vehicle tracking in low-angle and front-view images based on spatio-temporal markov random field model. In *8th World Congress on ITS*, 2001. 9

[24] V Kastrinaki, M Zervakis, and Kostas Kalaitzakis. A survey of video processing techniques for traffic applications. *Image and vision computing*, 21(4):359–381, 2003. 1

[25] Cheng-Hao Kuo and Ramakant Nevatia. Robust multi-view car detection using unsupervised sub-categorization. In *Workshop on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2009. 12

[26] Aliaksei Laureshyn, Håkan Ardö, Åse Svensson, and Thomas Jonsson. Application of automated video analysis for behavioural studies: concept and experience. *IET Intelligent Transport Systems*, 3(3):345–357, 2009. 4

[27] Anat Levin, Paul Viola, and Yoav Freund. Unsupervised improvement of visual detectors using cotraining. In *Ninth IEEE International Conference on Computer Vision (ICCV)*, pages 626–633. IEEE, 2003. 13

[28] Jinling Li, Hossein Hajimirsadeghi, Mohamed H Zaki, Greg Mori, and Tarek Sayed. Cyclists helmet recognition using computer vision techniques. *Transportation Research Record: Journal of the Transportation Research Board*, In Press, 2014. 6

[29] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981. 9, 31

[30] Xiaoxu Ma and W Eric L Grimson. Edge-based rich representation for vehicle classification. In *Tenth IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1185–1192. IEEE, 2005. 8

[31] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1):7–27, 2001. 11, 14, 18

[32] Antoine Messiah, Aymery Constant, Benjamin Contrand, Marie-Line Felonneau, and Emmanuel Lagarde. Risk compensation: a male phenomenon? results from a controlled intervention trial promoting helmet use among cyclists. *American journal of public health*, 102(S2):S204–S206, 2012. 3

[33] Erik Minge, Scott Petersen, and Jerry Kotzenmacher. Evaluation of nonintrusive technologies for traffic detection, phase 3. *Transportation Research Record: Journal of the Transportation Research Board*, 2256(1):95–103, 2011. 3

[34] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(7):971–987, 2002. 11, 14, 18

[35] Stuart R Perkins and Joseph I Harris. *Criteria for traffic conflict characteristics*. Research Laboratories, General Motors Corporation, 1966. 3

[36] Ross Owen Phillips, Aslak Fyhri, and Fridulv Sagberg. Risk compensation and bicycle helmets. *Risk analysis*, 31(8):1187–1195, 2011. 3

[37] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1201–1208. IEEE, 2011. 10

[38] DL Robinson. Bicycle helmet legislation: can we reach a consensus? *Accident Analysis & Prevention*, 39(1):86–93, 2007. 3

[39] Nicolas Saunier and Tarek Sayed. A feature-based tracking algorithm for vehicles in intersections. In *The 3rd Canadian Conference on Computer and Robot Vision*, pages 59–59. IEEE, 2006. 1, 4, 5, 8, 14

[40] Tarek Sayed, Mohamed H Zaki, and Jarvis Autey. Automated safety diagnosis of vehicle–bicycle interactions using computer vision analysis. *Safety science*, 59:163–172, 2013. 4

[41] Guruprasad Somasundaram, Vassilios Morellas, and Nikolaos Papanikolopoulos. Counting pedestrians and bicycles in traffic scenes. In *12th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2009. 4

[42] Xuefeng Song and Ramakant Nevatia. A model-based vehicle segmentation method for tracking. In *Tenth IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1124–1131. IEEE, 2005. 12, 29

[43] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2. IEEE, 1999. 8

[44] Min Sun, Hao Su, Silvio Savarese, and Li Fei-Fei. A multi-view probabilistic model for 3d object classes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1247–1254. IEEE, 2009. 12

[45] Åse Svensson and Christer Hydén. Estimating the severity of safety related behaviour. *Accident Analysis & Prevention*, 38(2):379–385, 2006. 3, 4

[46] Harini Veeraraghavan, Osama Masoud, and Nikolaos P Papanikolopoulos. Computer vision algorithms for intersection monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 4(2):78–89, 2003. 9

[47] Ian Walker. Drivers overtaking bicyclists: Objective data on the effects of riding position, helmet use, vehicle type and apparent gender. *Accident Analysis & Prevention*, 39(2):417–425, 2007. 2

[48] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176, Colorado Springs, United States, June 2011. 9, 37

[49] Simon Washington, Narelle Haworth, and Amy Schramm. Relationships between self-reported bicycling injuries and perceived risk of cyclists in queensland, australia. *Transportation Research Record: Journal of the Transportation Research Board*, 2314(1):57–65, 2012. 2

[50] Weilong Yang, Yang Wang, Arash Vahdat, and Greg Mori. Kernel latent svm for visual recognition. In *Advances in Neural Information Processing Systems (NIPS)*, pages 809–817, 2012. 36

[51] Mohamed H Zaki and Tarek Sayed. A framework for automated road-users classification using movement trajectories. *Transportation Research Part C: Emerging Technologies*, 33:50–73, 2013. 14, 16

[52] Mohamed H Zaki, Tarek Sayed, and Andrew Cheung. Automated collection of cyclist data using computer vision techniques. *Transportation Research Record: Journal of the Transportation Research Board*, 2013. 3

[53] Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, 2005. 13

[54] Xiaojin Zhu. Semi-supervised learning tutorial. In *International Conference on Machine Learning (ICML)*, pages 1–135, 2007. 13