Vehicle classification based on structural and local features

Suryanti Yunita Anggrelly

2011

Suryanti, Y. A. (2011). Vehicle classification based on structural and local features.
Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/43101>

<https://doi.org/10.32657/10356/43101>

VEHICLE CLASSIFICATION BASED ON STRUCTURAL AND LOCAL FEATURES

SURYANTI YUNITA ANGGRELLY

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering

2011

Acknowledgment

Firstly, I would like to express my sincere gratitude for my thesis supervisor, Dr Jiang Xudong at Nanyang Technological University. His guidance and valuable advice have given me direction throughout this work. I appreciate all his contributions of time and ideas for my research work. I would also like to thank Dr Eng How Lung, my supervisor from Institute for Infocomm Research, for his support and helpful discussions for this thesis work.

The support and encouragement from my colleagues in Institute for Infocomm Research have been indispensable for this research work.

I am grateful for my brothers, sisters, and friends from Reformed Evangelical Church of Singapore for their support and friendship throughout the journey of completing my Master of Engineering study.

I especially would like to thank Carlos Wiyono Kurniawan for his love, help, and encouragement especially during tough times of this thesis work.

Finally, I am deeply thankful for my beloved parents, Sihar Simanullang and Erika Harianja, for their unconditional love, faithful prayers and encouragements. This thesis is dedicated to them.

Contents

Acknowledgements	i
Contents	ii
List of Figures	v
List of Tables	vii
Abstract	vii
1 Introduction	1
1.1 Motivation	1
1.2 Project Objectives	3
1.3 The Report at a Glance	3
2 Literature Survey	5
2.1 Feature Extraction Methods for Object Classification	7
2.2 Object Classification Methods	9
2.3 Vehicle Classification Methods	11
2.4 Summary	13
3 Our Proposed Method	15
3.1 Hierarchical Method for Vehicle Classification	16
3.2 Experimental Set Up and Datasets	18

4	Structural Configuration	20
4.1	Motivation	20
4.2	Structural Features on Vehicle Images	22
4.3	Algorithm Design	25
5	Local Features Configuration	28
5.1	Motivation	28
5.2	SIFT Keypoints Localization	29
5.2.1	Scale Space Construction	29
5.2.2	Detection of SIFT Keypoints Candidates	31
5.2.3	Removal of Unstable Keypoints	31
5.3	SIFT Descriptor Representation	35
5.3.1	Magnitude and Orientation Assignment	35
5.3.2	Computing the Local Image Descriptor	36
5.4	Vehicle Classification using SIFT Descriptors	38
5.4.1	SIFT Keypoints Observation	38
5.4.2	SIFT Keypoints Matching	41
6	Template Selection and Keypoints Reduction	45
6.1	Forward Template Selection Method	46
6.2	Keypoints Reduction	48
7	Experimental Results and Discussions	51
7.1	Structural Configuration Method	52
7.2	Local Features Method	53
7.3	Hierarchical Method	54
7.3.1	Hierarchical Method with Template Selection	55
7.3.2	Hierarchical Method with Keypoints Reduction	57
8	Conclusions and Future Works	59
8.1	Conclusions	59
8.2	Future Works	60

CONTENTS

iv

Bibliography

63

List of Figures

1.1	Color Variations in Vehicle Class Car	2
1.2	Design Variations in Vehicle Class Van	3
3.1	Hierarchical Method Diagram	16
3.2	Traffic Monitoring on a Three-Lane Highway	18
4.1	Color Intensity Value along Vertical Axis	22
4.2	Steps in Major Planes Detection	23
4.3	Different Number of Nodes Detected in Class Car	24
4.4	Different Number of Nodes Detected in Class Van	25
4.5	Histogram of the Locations of the Major Planes in the Training Data	26
5.1	Construction of Difference of Gaussian Images (Reproduced from [1])	30
5.2	Gaussian Blurred Images of a Car at Different Octaves and Sigma Values	32
5.3	Difference-of-Gaussian Images of a Car at Different Octaves	33
5.4	Detection of Local Extrema in One Interval (Reproduced from [1]) . .	34
5.5	Assignment of Orientation (Reproduced from [1])	36
5.6	Descriptors Representation of SIFT Keypoints (Reproduced from [1])	37
5.7	Keypoints Localization Stages for a Car Image	37
5.8	Keypoints Localization Stages for a Van Image	38
5.9	SIFT Keypoints from Vehicles of Class Car at Different Octaves . . .	39
5.10	SIFT Keypoints from Vehicles of Class Van at Different Octaves . . .	40
5.11	Matching of a Van to another Van and to a Car without Geometrical Constraint	42

5.12	Matching of a Van to another Van and to a Car with Geometrical Constraint	42
5.13	Mean of SIFT Keypoints Match Percentage of Class Car Observed in Different Octaves	43
5.14	Mean of SIFT Keypoints Match Percentage of Class Van Observed in Different Octaves	44
6.1	Keypoints Reduction Schematic Diagram	48
7.1	Classification based on Local Features using a Single Template without Geometrical Constraint	53
7.2	Classification based on Local Features using a single Template with Geometrical Constraint	54
7.3	Vehicle Classification Accuracy for Template Selection	55
7.4	Classification Accuracy using Proposed Template Selection Method Compared with using 100 Random Selection of Five Templates	56
7.5	Vehicle Classification Accuracy for Keypoints Reduction	57
7.6	Classification Accuracy using Proposed Keypoints Reduction Method Compared with using 100 Random Reduction of Keypoints	58

List of Tables

7.1	Classification Results using the Structural Configuration	52
7.2	Classification Results using Five Templates	55
7.3	Classification Results using Reduced Number of Keypoints	58

Abstract

Object classification research has been moving towards invariant features extraction and development of a robust framework for object modeling and recognition. However, only a few works have been reported in implementing them in a real-time traffic surveillance system, in particular for vehicle classification task.

We propose a hierarchical method using structural and local features for vehicle classification in an automated real-time traffic surveillance system. In the first stage, major planes in the vehicle image are extracted to build the structural configuration of the vehicles. Descriptors obtained using Scale Invariant Feature Transform (SIFT) algorithm are used as the local features in the second stage of classification. Each class of vehicles is represented by a number of images selected using our proposed template selection method. Keypoints from these templates are further reduced to remove redundant keypoints. The proposed method was tested on images taken from a real-time traffic surveillance database and performed well on the vehicle classification.

Chapter 1

Introduction

1.1 Motivation

Rapid increase in system level performance of computers, fast digital signal processors, and wide range availability of digital cameras and video recorders, has made computer vision affordable for wide scale real-time applications. The installation of surveillance cameras is nowadays common in banks or stores for crime prevention, on the road for monitoring traffic conditions, in the offices for people activity monitoring, and many others.

Research on image processing and computer vision has also been growing to support automation of the video surveillance system. For an automated and real-time observation of the people or scene, a surveillance system requires computer vision for object detection, tracking, recognition or event detection. Computer vision has further the research to integrate multiple views from many cameras to build more complete view of the surveillance target or area.

In this project, we focus on classification of vehicle images obtained from an automated real-time traffic surveillance system. The tasks in a traffic surveillance system include vehicle detection and segmentation followed by classification. Robust vehicle segmentation is necessary for object recognition and event detection. Segmentation

of vehicle from the scene can be done in various methods such as background subtraction, frame difference, level set method and so on. Some of the challenges faced in segmenting out the vehicle from the background come from the weather conditions such as changing illumination conditions, clouds, rains, glaring sunshine that cause reflection, and so on.

In addition, the shadow cast by the trees or poles along the road, will pose challenge to the background subtraction method for vehicle segmentation. The shadow of the moving vehicle during daytime will result in under segmentation and cause problem in vehicle classification. We have developed a vehicle detection module based on the contour and edges information to detect foreground object in the images taken from a real-time traffic surveillance system.

The segmentation result will be sent to the object recognition module in the system. There are two different categories in object recognition, namely object identification and classification. Object identification focuses on specific object recognition in the image while object classification generalizes object characteristics in order to group given objects into their classes.

As in any general object classification problem, challenges in vehicle classification comes from the intra-class variations of the vehicles. These variations come from the automobile manufacturers that differentiate design, shape, color and ornamentation of vehicles from the same class (Figure 1.1 and 1.2).

Figure 1.1: Color Variations in Vehicle Class Car

One of the challenges of vehicle classification is the different designs of vehicles in the same class. Another challenge is the similarity in the shape and textures of the

Figure 1.2: Design Variations in Vehicle Class Van

parts across different class of vehicles. Thus, we need to extract features that allow the intra-class variations and at the same time discriminate the inter-class variations.

Therefore, the classification module must be able to generalize features that characterize objects from the same class and at the same time learn discriminative features to separate different classes of vehicles.

1.2 Project Objectives

The primary objective of this thesis project is to develop a vehicle classification for a real-time traffic surveillance. Another objective is to build representation of each class of vehicle that would be sufficiently general to handle the intra-class variations of vehicles from the same class due to the different designs and colors.

Therefore, the algorithm should select repeatable and discriminative features of the training images in order to classify the testing images. In addition, the features should be invariant to illumination changes and other challenges introduced by the natural conditions in real time situation of traffic monitoring system. These objectives are accomplished by the proposed hierarchical method that combines the structural and local features properties of the vehicle images.

1.3 The Report at a Glance

The remaining chapters are organized as follow:

- Chapter 2 presents a literature survey in the area of feature extraction methods for object classification, general object classification and particularly vehicle classification
- Chapter 3 introduces our proposed hierarchical algorithm for vehicle classification system that incorporates structural and local features of the object.
- Chapter 4 elaborates on structural configuration part of our proposed hierarchical method for vehicle classification. The building of the model for each vehicle class using the training data as well as the objective function to classify the test images are described in detail.
- Chapter 5 gives detailed description on local features detection using SIFT algorithm and its application in our proposed method. We describes the scale space construction, the keypoints localization and descriptors representation of the SIFT method. Also, we discuss the matching algorithm using SIFT descriptors and the geometrical constraint that we find necessary for the matching of the vehicles.
- Chapter 6 further details on the template selection and keypoints reduction methods for local features model. Template selection method is required since more images are required to improve the representation of each class in the training data. All the keypoints in the templates will be used as local features in our proposed hierarchical method. The steps taken to remove some of these keypoints that might be redundant will be described in the keypoints reduction method.
- Chapter 7 discusses the experimental set up and data preparations.
- Chapter 8 presents the results of the experiments.
- Chapter 9 concludes the findings of these experiments.

Chapter 2

Literature Survey

Visual object recognition can be divided into two different tasks, namely object identification and object classification. Object identification detects a specific object that is previously seen or learned by the system. Some tasks in object identification are specific object detection, object segmentation, and retrieval of all images from a database that contain the instances of a given object.

Murphy *et al.* [2] combined global and local features for object detection and localization. Taking the whole image as an input, they predict the rough location and scale of objects based on the gist of the scene, without running an object detector. Fragments were extracted from the image using sliding window classifiers applied to local features.

Meanwhile, object classification requires general descriptions of an object class from the training data to classify correctly a new image that has never been seen before. It makes use of the a priori knowledge of the object class characteristics. The standard framework to object classification is to determine what kind of features to extract, what type of approach to represent the object class and the type of classifier used.

The main difficulty faced by object classification is the large intra-class variability and therefore the need to generalize across these variations of objects belonging to the same class. It is a highly complex task to recognize or classify object from a single

image. Human visual system is able to learn the general characteristics of object class from the past observation over objects' variations. This is more challenging compared to identifying a specific object in the image, since the object can be described in a more precise manner.

In addition, the overall object classification algorithm is highly dependent on its feature selection method. Selection of good and discriminative features is very important in a classification system. Hall and Crowley [3] argued the significance of robust detection and identification of the significant class features to model object class for classification. They proposed a method to detect significant parts of the learned object and compute the generic features by unsupervised learning using clustering.

Features used in object classification system should capture unique and stable representation of an object class. They should be robust to scale, rotation, viewpoints, and illumination changes. In addition, they should be able to handle partial occlusion of the object.

Mikolajczyk and Schmid [4] designed multi scale Harris detector that has scale invariance property. Harris detector is based on the idea of auto-correlation to capture the structure of the local neighborhood features. Interest points were detected by applying threshold on the eigenvalues of the auto-correlation matrix. After identifying interest points in the image using Harris detector, the characteristic scale of each point was determined. Harris detector performs well in the presence of occlusion and clutter.

In addition, Schmid *et al.* [5] reported that edge based detector gives low performance in the repeatability experiment by subjecting the image into rotation and perspective transformation, while Harris-Laplace, Hessian-Laplace, LoG and DoG give good results. They also reported that scale-invariant detector can handle 40 degrees of viewpoint change. The Scale Invariant Feature Transform (SIFT) based descriptors were reported in [6] to outperform other interest point descriptors in image matching task. Yang and Newsam [7] used SIFT descriptors and Gabor texture features for classification of remote sensed imagery. SIFT descriptors were randomly sampled from the satellite images and clustered using the k-means algorithm. Each descriptor was assigned a label based on the closest cluster center. A single global

SIFT descriptor was computed based on the frequency count of these labels.

Furthermore, Dorko and Schmid [8] clustered SIFT descriptors and construct part classifiers using Support Vector Machine and Mixture of Gaussians. Feature selection was based on likelihood ratio and mutual information to obtain the most discriminative object part classifiers. SIFT was also used in semantic scene classification in [9], where Ayers and Boutell categorized photographs of home interior into a discrete set of classes using SIFT keypoint histograms. The importance of summarization of the keypoints using frequency information was introduced because of the many extraneous features. The experimental results shown that using SIFT keypoints histograms classified using Adaboost gives the best accuracy for 2 and 3 class problem compared to other combinations such as keypoints histogram with Support Vector Machine or with Nearest Mean classifiers.

Moreover, SIFT has been used in many object recognition systems for its robustness to scale, rotation and small illumination changes compared to other texture and appearance based approaches. Robustness to illumination change is especially important since different lighting conditions are expected throughout the day in a traffic surveillance system. Poor lighting is expected in early morning and evening in a traffic surveillance system. Strong shadow will also add noise into any interest point detector. Taking all these into consideration, we get the SIFT descriptors from the vehicle images as one of the features for the vehicle classification system.

2.1 Feature Extraction Methods for Object Classification

General approaches used in features extraction for object recognition can be categorized into three groups, namely the global-based approach, component-based approach, and hierarchical approach. The first approach considers the whole image as a single unit and recognition is performed on the features extracted from the entire image. Support Vector Machine or Neural Networks are some classifiers that are commonly used to discriminate between the different classes.

Global approaches have been successfully applied when the pose of the object was fixed, such as for face detection [10], shape-based object recognition, handwritten digit recognition, pedestrian detection [11], etc. However, using global features for object recognition was not robust to occlusion compared to using local features. Also, it lacks invariance and is highly dimensional since this approach considers the whole image.

Component-based approach has been widely used in object detection task. This approach has been intuitively and empirically more accurate and robust compared to global approach especially in handling partial occlusions and visual changes such as different pose, image rotation, illumination or noise ([12],[13]). Each part of an object should be less sensitive to changes in illumination change and vary less under different viewpoint changes compared to using the whole object.

Many works have been done related to component-based approach for object detection especially in selection of which parts of object and what type of features to use. In face detection field, for example, systems described in [14] and [15] selected features that are salient to human visual system such as eyes, nose, and mouth. Other works such as in [16] learned object parts or image patches to determine salient object parts automatically.

Ullman and Sali [17] represented class objects in terms of common image fragments that represent the variability of different objects that belong to a common class, such as a car or a face. Selection of best representative fragment is based on maximum mutual information of the fragments and the class they represent. Further, Epshtein and Ullman [18] proposed a top-down features extraction for object classification. First, they extract informative top-level fragments and the fragments are broken down successively into their optimal components.

A hierarchical approach to classify an object either as a person or a non person is presented in [13]. Mohan *et al.* [13] developed a general example-based algorithm for detecting objects in images by first detecting the object's components and then combining them with a classifier if the geometric configuration between these components is valid. This algorithm was robust in detecting any object composed of distinct identifiable parts that are arranged in a well-defined configuration such as

faces and cars, in complex and cluttered scenes. This results into a more robust method compared to using full-body person detection method since it is capable to locate partially occluded views of people.

Leung [19] built a hierarchical component-based car detector in street scene images. The classification scheme was developed based on learning similarities of features extracted by an interest operator. The classifier operates by first locating the keypoints and compared these against a corpus of car-specific keypoints learned from the training data.

2.2 Object Classification Methods

Object recognition has posed challenging tasks in computer vision research since it requires ability to generalize under large intra-class variation, changes in illumination and viewpoint, clutter, deformations, as well as partial occlusions.

There are generally three approaches for object classification, namely appearance-based approach, structure-based approach, and hybrid approach that combines the previous two approaches. Pattern of color intensity is the main feature in appearance based object recognition algorithm such as bag-of-words method. Inspired by method used in document categorization algorithm, an object is represented using some collection of building blocks without coherent geometric relations. Patches of building blocks were represented in vectors clustered to form code words dictionary in works by Vogel *et al.* [20] and Fei-Fei *et al.* [21]. All components of an object have equal probability in bag of words method.

Two main approaches in structure based object recognition, where spatial configuration of the object is primary, are implicit and explicit structure model. In implicit structure model, the structural information of the object was not taken into consideration systematically but the fragments are assembled like jigsaw puzzle to fit the object [22]. The attempt to record the relation between each part and the centroid of the object in [23] introduced implicit structure into object recognition.

Pictorial structure method proposed by Fischler and Eschlarger in [24] explicitly

imposed some spatial configuration that allows certain control of structural information. It has intuitive visual sense because it captures features that are common between instances of a class. Felzenszwalb [25] proposed an object modeled using graph that consists of nodes that represent parts, edges that denote spatial relations between parts and spatial prior on configuration of parts.

Exploring further, Agarwal and Roth in [12] described the structure using angular relation and distance between each pair of parts. However, this will not work well when the object is stretched since the relation is modeled based on constant displacement between parts of an object. In traffic surveillance case, we expect varying vehicle sizes depending on the camera position relative to the road. Relational structure between parts of the vehicle has to be invariant to vehicle size. In implicit structure model, scale invariance can only be achieved by exhaustive search over all sizes.

The last approach in object recognition is a combination of both appearance and structure based methods. An object class is defined in [26] as a collection of objects which share characteristic features or parts that are visually similar and having similar spatial configurations.

Salient features are extracted using automatic features detectors and modeled using probability density function. Constellation of parts model combines appearance and geometrical methods. Sparse affine-invariant feature points are detected and then modeled using Gaussian spatial model of how feature locations vary within category. While this method explicitly model shape variability, it does not model the appearance variability since appearance of a part is fixed before shape learning.

Taking this into account, Fergus *et al.* [27] proposed a general statistical framework for object class recognition. Their method learnt appearance and shape simultaneously. In their experiment, all aspects of objects: shape, appearance, occlusion and relative scale are modeled using flexible constellation parts. This generative model assumes independence of the local features observed given their assignments to parts in the model.

Region of interests were found using salient region operator developed by Kadir and Brady [28]. Locations and scale that is the radius of region in pixel are computed.

Then, to compute the appearance-based features, the pixels in the region obtained from previous step are normalized and then sub sampled to result in an 11x11 patch. These pixels are then projected onto PCA basis resulting in low-dimensional vector space. They combined feature locations, appearance (feature representations) and scale (vector of feature relative scales) using Bayesian decision for object recognition.

Heisele *et al.* [29] used Support Vector Machine (SVM) classifiers to learn discriminative components of objects. First, component classifiers detected facial components and second, a combination classifier learned the geometrical relation between the components and performed the final detection of the object. Their algorithm was applied on face classification - not vehicle classification. Also, they used textured 3-D head models to generate the training data. Our experiment deals with natural object images from a traffic surveillance data.

2.3 Vehicle Classification Methods

Vehicle classification problem is related to vehicle detection and recognition. Many works have been done in vehicle detection and tracking such as in [12], [30], [19], and [31]. For vehicle detection, Gupte *et al.* [32] developed a camera calibration tool to recover the location, length, width and velocity of the vehicle fragments from the image. Vehicles were classified into two categories, namely cars and non-cars using the actual dimensions of vehicles (length and height) obtained using the camera calibration tool. They reported 70% correct classification of the vehicles detected.

However, our experiment does not require actual dimensions of the vehicles. The relative dimension of each building part of a vehicle is however considered in building the structural configuration of the system.

Javed and Shah in [33] developed tracking and classification method in realistic scenarios using a single camera. Objects were classified based upon detected recurrent motion for each tracked object. They reported that different types of objects yield very different Recurrent Motion Image and thus can be easily classified as a single person, a group of persons or a vehicle.

Following the detection and segmentation steps, measurement vector was computed from each foreground blob and projected using Linear Discriminant Analysis (LDA) onto a lower dimensional space better suited for classification in [34]. The classifier used weighted nearest Neighbor method and Fuzzy -means (FCM) clustering to learn the training set. However, this system does not incorporate any shape information, which is one of the discriminative features in vehicle classification system.

Hinz [31] modeled a vehicle on a local level by a 3D-wireframe representation to describe the prominent and radiometric features of cars. The global level is based on a more generic knowledge about vehicles as part of vehicle queues. Xia [35] proposed a vehicle recognition system by fitting the model to the vehicle in the image to estimate the pose and shape parameters of the vehicle in the image. Generic shape of the model for each class was derived using Principal Component Analysis (PCA) on a set of training shapes. In a more detailed manner, Koller [36] developed a generic polyhedral vehicle, a three dimensional model parameterized by 12 length parameters.

In [37] and [38], Sullivan *et al.* built a parameterized vehicle model for different vehicle classes. These models were fitted to the sample data and subjected to a PCA to create deformable models to make the system more generic. They reported that using the six PCA parameters shows significant improvements to vehicle selectivity and offer greater robustness in vehicle tracking. Modeling vehicles using this approach may not be sufficient in our experiment due to the viewpoint of the surveillance camera used.

Similarly, Wu *et al.* [39] used wireline models to classify vehicles into six classes, namely cars, small carriage, large carriage, small truck, large truck, and others. These two works make use of the geometrical properties of vehicles for the classification task, which is a reasonable approach for rigid object classification. The 91% recognition rate for the later work was achieved using neural network classifier and thus lacking in semantic meaning of vehicle modeling.

Further, using computer graphics models viewed from the top has been explored for vehicle classification in [40]. Features were extracted using eigen window from edge image, that served as template for the classification. Similar work by Gupte *et*

al. [30] explored motion segmentation using edge detection on two consecutive frames to obtain clear outline of the vehicle.

Lastly, Ma and Grimson [41] developed a rich representation for object classes based on edge points and modified SIFT descriptors for vehicle classification. They explored explicit and implicit shape models as well as using shape information alone. Similar to their work, our system takes into account both appearance and structural properties of the object and combines them in a systematic classification framework.

2.4 Summary

Various methods in vision-based vehicle classification can be divided into two categories based on their feature selection methods, namely geometrical and local features. The wireline model, 3D-wireframe representation and general polyhedral vehicle model are some of the methods that fall on the first category. These methods construct a model for each class of vehicle based on the geometrical measurements of vehicle images in each class. This approach is reasonable for rigid object classification. In this work, we detect the major building blocks of the vehicle images to build the structural configuration of each class of vehicle. However, we consider the relative dimensions of the major building blocks instead of their actual dimensions.

The second category in the vehicle classification method builds their model based on local features such as edges, SIFT descriptors, etc. This method is more robust in handling partial occlusion and noise compared to global-based approach since the local features of an object will be less sensitive to illumination changes and vary less under different viewpoint changes compared to its global features. SIFT descriptors has been reported to be robust in handling scale, rotation and illumination changes for object detection and matching. Thus, we extract SIFT descriptors to represent local invariant features in addition to the structural features of the vehicle images.

Furthermore, the task in vehicle classification requires not only identification of a particular object but grouping similar objects into a specific class. This task requires generalization across variations of objects within the same class. Therefore, we proposed using templates to represent different classes of vehicles for the classification.

The templates are selected from the training images using forward selection method to determine the best templates based on highest classification accuracy using SIFT matching algorithm. In addition, we use the backward selection method to remove some redundant keypoints from the selected templates to increase computation efficiency.

Chapter 3

Our Proposed Method

First of all, we observed that vehicles from class *Van* consist of three major parts, namely the roof, the rear window and the trunk while vehicles from class *Car* consist of four to five major parts, namely the hood, the windshield, the roof, the rear window and the trunk. Thus, we extract the structural features of the vehicles that detect the major building parts of the vehicles to discriminate the two classes of vehicles in the first stage of our proposed hierarchical vehicle classification system.

The major building parts of each vehicle image are segmented based on the gradient of the intensity values along the vertical axis of that image. The locations of the major building parts detected in all images in the training data are used to build the structural configuration of each vehicle class. Nevertheless, the number and the locations of the major building parts detected from the vehicle images may not be stable and hence cause error in the classification. The instability may be caused by poor segmentation of the vehicles from the background. In over segmentation case, some major building parts are segmented out while in under segmentation case, the background may be detected as one of the major building parts of the vehicle.

Therefore, we need additional features from the vehicles images to further discriminate the two classes of vehicles. The features need to be invariant to scale changes since the sizes of the input images are not uniform. Mikolajczyk and Schmid [4]

designed multi scale Harris detector that has scale invariance property. After identifying interest points in the image using Harris detector, the characteristic scale of each point is determined.

3.1 Hierarchical Method for Vehicle Classification

Our proposed method can be summarized in the schematic diagram shown in Fig. 3.1. Our system takes into account both structural and local features of the object and combines them in a hierarchical vehicle classification framework. Images were obtained from a traffic surveillance camera placed on an overhead bridge. After object detection and segmentation, the vehicles images are used as the input to our vehicle classification system. The segmented image of vehicle is the input to our system. In this experiment, we focused on two classes of vehicles, namely *Car* and *Van*.

The first stage is to use the structural configuration to classify the vehicles into class *Car*, *Van* and *Uncertain*. In order to classify the vehicles into class *Car*, *Van* and *Uncertain* group, we need two threshold values for the values of the sum of the posterior probabilities for class *Car* and *Van*. By varying these two thresholds, we get different number of images that are correctly classified into class *Car* and *Van* based on their structural configuration. In the training stage, we determine the minimum threshold to classify the vehicles into class *Car* and *Van* without any error. Those vehicles which do not fall in either of the two categories were further classified using their local features.

Figure 3.1: Hierarchical Method Diagram

Subsequently, the local features of these vehicle images are extracted using SIFT

algorithm. A number of images were selected from the training data as the templates of each vehicle class using the template selection method. All keypoints from these templates are compared against all keypoints from the test image. We adopt Lowe’s method for object matching using SIFT descriptors in [1]. We obtained an image from each class in the training data set to be used as the template. Each image in the testing data was matched against the template from each class and the number of keypoints matched are counted. This number is normalized to the total number of keypoints being matched, both from the template image and the test image.

We extracted local features using SIFT algorithm. The local features extracted need to be invariant to scale changes since the sizes of the input images are not uniform. Mikolajczyk and Schmid [4] designed multi scale Harris detector that has scale invariance property. After identifying interest points in the image using Harris detector, the characteristic scale of each point is determined.

However, one drawback in global feature extraction such as this is its sensitivity to illumination changes, background clutter and occlusion. Thus, we cannot rely on global features alone in our object class modeling for our traffic monitoring system since illumination change is expected throughout the day. Region based feature extraction methods such as Scale Invariant Feature Transform (SIFT) address this problem. Therefore, we proposed using SIFT algorithm in the second stage of our proposed hierarchical method of vehicle classification system.

We adopt Lowe’s method for object matching using SIFT descriptors in [1]. We obtained an image from each class in the training data set to be used as the template. Each image in the testing data was matched against the template from each class and the number of keypoints matched are counted. This number is normalized to the total number of keypoints being matched, both from the template image and the test image. We classify the vehicles based on the maximum number of matching descriptors between the test image and descriptors in the template of each vehicle class.

However, we observe from our experiment that using only one image is not sufficient to classify the vehicles. One possible reason is the large variations in the designs of vehicles from the same class. Thus, we add more templates to match against

the test images. All SIFT keypoints' descriptors from these templates are compared against keypoints' descriptors from the test image.

The templates were selected from each vehicle class of the training data using our proposed template selection method. The number of keypoints increases as the number of templates used increases. However, we observe that some keypoints in different vehicles in the templates of the same class are similar and can be removed without affecting the classification accuracy. Hence, some of the keypoints from these templates are removed systematically using our proposed keypoints reduction method for more efficient and effective representation of the templates. This also means reduced computation time during the matching process.

The final classification result of the hierarchical method is defined as the total number of images being classified correctly both in the structural and local features configuration.

3.2 Experimental Set Up and Datasets

We consider a setup in traffic video surveillance system in which fixed overhead cameras are installed on the road side or on an overhead bridge to monitor the traffic condition along the highways. The placement of surveillance cameras on the overhead bridges results in top view of the vehicle images as shown in Figure 3.2.

Figure 3.2: Traffic Monitoring on a Three-Lane Highway

Foreground objects in the current frame are detected by subtracting the current

frame from the background color model and edge model based on the method explained in [42]. Color and edge models are considered as these two features provide complementary advantages. This yielded the segmented vehicle images for classification purpose.

Furthermore, we consider segmented objects whose centroid positions are within certain distance from the camera as input to our classification system. This is to prevent partial occlusion due to the vehicles' entrances and exits from the camera view. The sizes of the vehicle objects will vary according to their class, speed and processing speed to capture the images. The original image resolution is of VGA size (640 x 480 pixels). Vehicle size obtained after the background subtraction process is about 150 x 100 pixels.

In this project, we focus on classification of two types of vehicles, namely *Car* and *Van*. One of the reasons is because majority of vehicles on the road are cars, which include taxis. Also, our proposed method makes use of the structural features of vehicles to discriminate different classes of vehicles based on their building structures. Vehicles of class Van and other type of vehicles which are similar to it such as buses have distinguishable structural property from vehicles of class *Car*.

Some other types of vehicles have similar structural features as these two classes. For instance, a bus has similar structure as a van. For vehicles from class motorcycle or truck are distinguishable from other classes of vehicles based on their sizes. Therefore, it is possible to extend our proposed method to classify more types of vehicles on the road. Taking all these into consideration, we demonstrate our proposed method for vehicle classification on two main classes, namely *Car* and *Van*.

Furthermore, we only consider segmented objects whose centroid positions are within certain distance from the camera as input to our classification system. This is to prevent partial occlusion due to the vehicles' entrances and exits from the camera view. The sizes of the vehicle objects will vary according to their class, speed and processing speed to capture the images. The original image resolution is of VGA size (640 x 480 pixels). Vehicle size obtained after the background subtraction process is about 150 x 100 pixels. In this project, we focused on vehicles of class *Car* and *Van*. Experiments for more classes of vehicles can be carried out in similar ways.

Chapter 4

Structural Configuration

4.1 Motivation

Shape based object recognition since 1960's such as L.G Roberts [43] has recognized the importance of building a composite structure of object parts for a complete description of the three-dimensional structure of an object. Research work by Zerroug and Nevatia [44] exploited the projective properties of object parts using generalized cylinders as primitives to construct a 3-D description of an object.

Despite the lack of robustness to handle objects under real conditions, these early works remain significant in today's object recognition research due to several important principles such as the importance of shape, viewpoint-invariant, 3-D representations, symmetry feature, complex object decomposition, and hierarchical representation in defining object categories [45].

Vehicles can be categorized under rigid object class for their consistent geometric structure. Shape based recognition approach can be applied to model the vehicle object. Object decomposition, hierarchical parts representation and other similar shape-based representations can be used to describe the vehicle object. In vehicle classification problem, however, this approach requires certain viewpoint to obtain best features representation to discriminate different classes of vehicles.

In this project, we focus on classification of two types of vehicles, namely class *Car* and *Van*. One of the reasons is because majority of vehicles on the road are cars, which include taxis. Also, our proposed method makes use the structural features of vehicles that discriminate different classes of vehicles based on their building structures. Vehicles of class *Van* and other type of vehicles which are similar to it such as buses have distinguishable structural property from vehicles of class *Car*.

The major building parts of a van viewed from certain angle from the top generally consists of the roof, the rear window and the trunk. Meanwhile, there are generally four major parts that are visually apparent from a car in this view, namely the hood, the roof, the rear window and the trunk. Besides the difference in the number of building parts, these two classes of vehicles generally differ in the shape and size of each part too. These observations motivate us to experiment on these two classes of vehicles.

Some other types of vehicles have similar structural features as these two classes. For instance, a bus has similar structure as a van. Also, an SUV has major building parts that are similar to that of class van, even though they differ in shape. For vehicles from class motorcycle or truck are distinguishable from other classes of vehicles based on their sizes. Therefore, it is very feasible to extend our proposed method to classify more types of vehicles on the road. Taking all these into consideration, we demonstrate our proposed method for vehicle classification on two main classes, namely *Car* and *Van*.

For the experiment, we collected video data from the camera installed on an overhead bridge. This results in top view images of the vehicles. The top view of a car consists of distinguishable parts such as roof, hood, trunk, rear window, wind screen. These parts are arranged in fixed spatial configuration building the basic structure of the car. The top view of a van generally consists of roof, rear window and trunk. Therefore, these two classes can be differentiated using the major building parts of each vehicle class. The difference between the major building parts are not only on the number of parts but also on the shape and dimensions of each part as well as on the relative positions of each part to the other parts of the vehicle.

In summary, the structural configuration step attempts to discriminate various

classes of vehicles based on their outlining structures. The outlining structure of each vehicle class is formed by the major building parts of each vehicle class. These parts can be detected and represented as the major planes detected from the top view of the vehicles. The planes are labeled as nodes in the configuration. These nodes therefore make up the outlining structure of the model of each category of vehicles.

4.2 Structural Features on Vehicle Images

Detection of the major building parts of the vehicle image is based on the gradient of the gray level intensity values along the vertical axis of the image. Each vehicle image is firstly pre-processed using an average filter of size 3×3 in order to reduce image noise due to illumination on the vehicle surface and to reduce detail due to variations in vehicle surface texture or graffiti that might be found on the vehicle.

Next, we get the color intensity profile along the vertical axis of the vehicle object. We may observe the difference in the intensity profile of vehicles from the two classes as shown in Figure 4.1.

Figure 4.1: Color Intensity Value along Vertical Axis

Subsequently, we get the gradient value of this color intensity value. A transition point in the gradient intensity profile is marked if the value is larger than a certain

threshold. The threshold value is determined by taking the maximum value of the gradients into consideration. In this way, the threshold value is not fixed, but determined by the profile of gradient along the vertical axis of the vehicle. From our experiment on the training data, we select the threshold value to be $0.25 * y_{max}$, where y_{max} is the maximum value of the gradient values.

The next step is post processing of the detected transition nodes to remove some noise in the transition points detection. Any transition point that is too close to its neighboring points are merged.

The final transition points obtained are used to define the major building planes of the vehicle. The mid point between any two subsequent transition points is defined as the node. This node represents the major plane detected in the vehicle image. These few steps can be seen in Figure 4.2.

Figure 4.2: Steps in Major Planes Detection

Our first intention is to associate each major building part of the test image to the one in the model. However, the association between one plane to another needs additional features such as the size of each plane, the positions of the plane with respect to the rest of the planes, the locations of the planes, etc. In addition, this will require all the major planes being correctly detected which are not always possible for typical traffic surveillance whose image quality is quite low.

Some of the results obtained using gradient transition to find the nodes can be seen in Figure 4.3 for class *Car* and Figure 4.4 for class *Van*. From these two figures we observe that the number of planes detected may be different from one vehicle to another even though they are from the same class. We expect there are five major planes for vehicles from class *Car*. However, in some car images, the number of planes detected is more and in some other images there are less major planes being detected compared to the expected value.

An instance where more planes being detected can be seen in Figure 4.3(c). The extra transition point detected is caused by the significant color intensity value of the object on the back dashboard of the car. Figure 4.3(d) is an example for the less number of planes being detected. In this case, the rear bumper of the car is not detected due to the false detection of the car plate as the transition point. This false detection caused the trunk and rear bumper being detected as one plane.

Figure 4.3: Different Number of Nodes Detected in Class Car

Similarly for vehicles from class *Van*, the number of major planes detected may

vary around the expected three major planes for this class. For instance, the number of planes detected in the black Van as shown in Figure 4.4(b) is only two. This is mainly caused by the similarity of the main color intensity value of the van and its rear window.

Figure 4.4: Different Number of Nodes Detected in Class Van

Thus, we need to take into account the variation on the number of planes detected from the vehicles of the same class for the structural model. Each plane in the vehicle image is represented by its node location. The histogram of the nodes locations of all vehicle images in the training data (100 Cars and 100 Vans) is shown in Figure 4.5.

4.3 Algorithm Design

We model the locations of all nodes detected from vehicles in each class using mixtures of Gaussians as shown in Equation 4.1. This model denotes the class conditional probability distribution function for vehicles of class ω_i .

$$p(x|\omega_i) = \sum_k \alpha_k \cdot G\{x|\mu_k, \sigma_k^2\} \quad (4.1)$$

x is a random variable whose samples are all node locations detected from all images of class ω_i in the training data and $G\{x|\mu_k, \sigma_k^2\}$ is a one dimensional Gaussian Mixture

Figure 4.5: Histogram of the Locations of the Major Planes in the Training Data

model defined in Equation 4.2.

$$G\{x|\mu_k, \sigma_k^2\} = \frac{1}{\sigma_k \sqrt{2\pi}} e^{-(x-\mu_k)^2/2\sigma_k^2} \quad (4.2)$$

where μ_k and σ_k^2 are the mean and variance of the k^{th} Gaussian component respectively, α_k is the weight of the k^{th} mixture component and k is the number of components in the Gaussian mixture model.

Training samples of each class are used to estimate the means and variances of each component in the Gaussian mixture model for each class of vehicle. We set the number of Gaussian components for class *Car* $k = 5$ and for class *Van* $k = 3$. This is based on the fact that the number of reliable nodes detected from a car and a van is 5 and 3 respectively.

In classifying the vehicles based on their structural configuration, we need to compute the posterior probability of each vehicle class ω_i given an image represented by its structural configuration, as shown in the following equation:

$$\begin{aligned} P(\omega_i|x_n) &= \frac{p(x_n|\omega_i)P(\omega_i)}{p(x_n)} \\ &\propto p(x_n|\omega_i)P(\omega_i) \end{aligned} \quad (4.3)$$

where $p(x_n|\omega_i)$ is the conditional probability distribution of all node locations, x_n , detected from the test image given vehicle class ω_i and $P(\omega_i)$ is the prior probability of vehicle class ω_i .

Substituting Equation 4.1 for the conditional probability distribution, $p(x_n|\omega_i)$, and Equation 4.2 for the one dimensional Gaussian mixture model we obtain:

$$\begin{aligned} P(\omega_i|x_n) &= \sum_k \alpha_k \cdot G\{x_n|\mu_k, \sigma_k^2\}P(\omega_i) \\ &= \sum_k \alpha_k \cdot \frac{1}{\sigma_k \sqrt{2\pi}} e^{-(x_n - \mu_k)^2 / 2\sigma_k^2} P(\omega_i) \end{aligned} \quad (4.4)$$

Next, the objective function for the classification is defined in Equation 4.5 as the maximum sum of the posterior probabilities of all node locations, x_n , detected from the test image.

$$\hat{C} = \arg \max_{i=1,2} \sum_n P(\omega_i|x_n) \quad (4.5)$$

where n is the total number of nodes detected in the test image, and i is the index of vehicle classes. By substituting Equation 4.1 and 4.2 to the posterior probability function, the objective function can be computed as follow:

$$\hat{C} = \arg \max_{i=1,2} \sum_n \sum_k \alpha_k \cdot \frac{1}{\sigma_k \sqrt{2\pi}} e^{-(x_n - \mu_k)^2 / 2\sigma_k^2} P(\omega_i) \quad (4.6)$$

Chapter 5

Local Features Configuration

In addition to the structural features, the local features of the image are extracted to further discriminate the vehicles into the two classes, *Car* and *Van*. In the local features configuration, we firstly model the SIFT descriptors into two major parts. The first part is the locations of the keypoints. The second part is the keypoints magnitude, major orientation and the 128 elements SIFT descriptors. The vehicle images that we obtained are of varying sizes. Therefore, we need to normalize the locations of the keypoints with respect to the height and width of the image.

In this section, we firstly describe local features extraction from the vehicle image using SIFT algorithm invented by Lowe [46] and then discuss the implementation of SIFT matching algorithm in vehicle classification problem.

5.1 Motivation

SIFT algorithm has been used in many object recognition system for its robustness to scale, rotation and small illumination changes. Robustness to illumination change is especially important since different lighting conditions are expected throughout the day in a traffic surveillance system.

In object recognition research, SIFT is normally used for exact matching. Each

reference image in object prototypes has a set of keypoints together with their corresponding descriptors. Matching is done by comparing SIFT descriptors obtained in the test image with descriptors of the reference image. However, we are dealing with sub-class object categorization in vehicle classification. We need to observe if SIFT algorithm is able to generalize the features of objects in the same class and at the same time is discriminative for features from other object class.

5.2 SIFT Keypoints Localization

SIFT algorithm was invented by David Lowe [46] to detect and represent local invariant features in an image for object detection and recognition. These features are invariant to image scale and rotation and also robust to changes in illumination, noise, occlusion and small variation of viewpoint.

There are two key issues in feature extraction using SIFT algorithm, namely keypoints localization and descriptor representation. Keypoints localization in SIFT method involves detection of extrema across different scale space functions and removal of unstable keypoints due to low contrast and edges response. This is performed by searching for stable features across all possible scales using scale space function. The next step is to get the descriptor representation for each keypoint from a set of orientation histograms on a 4×4 pixel neighborhoods around the keypoint.

5.2.1 Scale Space Construction

Prior to keypoints localization is detection of locations and scales that are invariant to scale change of the image. This is performed by searching for stable interest points across all possible scales using scale space function. These points are expected to be found at the extrema of the Laplacian images in the scale-space representation. The construction of the scale-space representation can be summarized in Figure 5.1.

The image is convolved with Gaussian filters incrementally, introducing smoothing parameter σ at different scales of the scale-space representation $L(x, y, k\sigma)$ according

Figure 5.1: Construction of Difference of Gaussian Images (Reproduced from [1])

to:

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y) \quad (5.1)$$

where $I(x, y)$ is the vehicle image observed, $*$ is the convolution operator in x and y , and $G(x, y, \sigma)$ is a 2D-Gaussian function with standard deviation σ as defined in Equation 5.2.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5.2)$$

The convolved images are grouped by octave. Each octave of scale space consists of $s + 3$ number of images, where s is an integer number of intervals so that the value of $k = 2^{1/s}$, in order to form $s + 2$ Difference of Gaussian images.

The Difference of Gaussian image is computed from the difference of two adjacent scale space images separated by a constant multiplicative factor k as shown in Equation 5.3. The subsequent scale is generated by resampling the Gaussian image that has twice the initial value of σ . Then, the processes are repeated until all the number of pyramid levels is obtained.

$$D(x, y, \sigma) = L(x, y, k_i \sigma) - L(x, y, k_j \sigma) \quad (5.3)$$

In our experiment, we set the number of sampling to be 3 scales per octave based on experimental results by Lowe in [1] that this value generates the most stable keypoints that are repeatable for object matching. Figure 5.2 shows a car image that is being blurred by convolution with the Gaussian filter using different smoothing parameters σ .

The Difference-of-Gaussian (DoG) for the car image observed can be seen in Figure 5.3. Subsequently, all interest points that correspond to extremum points in all DoG scale-space functions are detected on these Difference of Gaussian images.

5.2.2 Detection of SIFT Keypoints Candidates

Candidates for the keypoints are defined as local maximum or minimum points in the DoG images across scales. Since the DoG filter is used, these candidates will be detected on a region with different intensity compared to its surrounding area. They are obtained by comparing each pixel to its eight neighbors in the current DOG image and to its nine neighboring pixels in the scale above and below as shown in Figure 5.4. The extrema points in the first and last images of the octave are not searched since not all the 26 neighbors are known.

Further processing is required since some of these candidates are unstable, such as points along an edge or those having low contrast due to their sensitivity to noise. Also, the keypoints along an edge in DOG image will be unstable to small amount of noise, since DOG function will have a strong response along edges.

5.2.3 Removal of Unstable Keypoints

Detection of interest points in a DoG images will likely to give strong response on corners as well as blobs in the image. However, it is also expected that some interest points will be detected along the edges, which are not desired for being prone to even small amount of noise. In order to remove the unstable keypoints, the following steps

(a) 1st Octave

(b) 2nd Octave

(c) 3rd Octave

(d) 4th Octave

Figure 5.2: Gaussian Blurred Images of a Car at Different Octaves and Sigma Values

(a) 1st Octave

(b) 2nd Octave

(c) 3rd Octave

(d) 4th Octave

Figure 5.3: Difference-of-Gaussian Images of a Car at Different Octaves

Figure 5.4: Detection of Local Extrema in One Interval (Reproduced from [1])

are taken:

a. Interpolation of nearby data for accurate position

To improve matching and stability, the interpolated location of the extremum is taken. The interpolation is done using Quadratic Taylor expansion of the DOG scale-space function. This Taylor expansion can be written as:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (5.4)$$

where D and its derivatives are evaluated at $x = (x, y, \sigma)^T$, which is the offset from the particular point. In the experiment, the Hessian and derivative of D are approximated using differences of neighboring sample points that results on a 3×3 linear system.

b. Discarding low-contrast keypoints

Unstable extrema with low contrast can be removed by using:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (5.5)$$

From experiments, Lowe suggested to discard extrema with a value of $|D(\hat{x})| < 0.03$.

c. Eliminating edge responses

The keypoints along an edge in DOG image will be unstable to small amount of noise, since DOG function will have a strong response along edges. The edges are detected by a 2×2 Hessian matrix. Principal curvatures are computed from the Hessian matrix:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (5.6)$$

where the derivatives are estimated by taking differences of neighboring points.

Several keypoints are removed after discarding the low contrast keypoints as well as those keypoints along the edges in the DoG images. After removing these unstable keypoints, scale-invariant local descriptors of the remaining keypoints are computed.

5.3 SIFT Descriptor Representation

SIFT descriptor representation is computed in such a way that it is highly distinctive and partially invariant to variations such as change in illumination or 3D viewpoint. The first step in descriptor representation is magnitude and orientation assignment followed by taking the histogram of neighborhood pixels around the keypoint. Invariance to image rotation can therefore be achieved as the keypoint descriptors can be represented relative to this orientation.

5.3.1 Magnitude and Orientation Assignment

The computations of gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ are performed in the Gaussian blurred image L with the closest scale for scale invariance. Using pixel differences, gradient magnitude and orientation are computed as follow:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (5.7)$$

$$\theta(x, y) = \arctan \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (5.8)$$

From the gradient orientations within a keypoint region, all samples are added into its histogram after being weighted by its gradient magnitude and by a Gaussian circular window with σ that is 1.5 times that of the keypoint scale. Highest peak in the orientation histogram corresponds to dominant directions of local gradients (Figure 5.5). Peaks that are within 80% of this highest peak are assigned to the same keypoint but different orientations. Therefore, for location where there are multiple peaks of similar magnitude, multiple keypoints will be created at the same location and scale.

Figure 5.5: Assignment of Orientation (Reproduced from [1])

5.3.2 Computing the Local Image Descriptor

In this last stage, a set of orientation histograms on 16×16 pixel neighborhoods are computed. For each 4×4 sub region, a histogram of all the gradient direction of all keypoints is quantized into 8 bins. The basic idea is shown in Figure 5.6 for a 2×2 region. Thus, each keypoint has in total $4 \times 4 \times 8 = 128$ elements of feature vector. The histograms are then normalized to unit length to reduce the effects of illumination change and stored as the SIFT descriptor.

Figure 5.6: Descriptors Representation of SIFT Keypoints (Reproduced from [1])

Interest points detected in an instance of vehicles from class *Car* and *Van* can be seen in Figure 5.7 and Figure 5.8 respectively. The final SIFT keypoints are drawn using arrow representation in the right most column of the figures. The length of the arrow represents the magnitude of the keypoint and its direction represents the major orientation of the SIFT descriptors.

Figure 5.7: Keypoints Localization Stages for a Car Image

The descriptors in final keypoints detection will then be matched against stored descriptors of the model of each vehicle class stored in the templates to determine the vehicle class. In the next section, we explore the feasibility of using the SIFT descriptors for vehicle classification.

Figure 5.8: Keypoints Localization Stages for a Van Image

5.4 Vehicle Classification using SIFT Descriptors

Generally SIFT has been reported to be robust for object detection and object matching in images with high resolution. However, the task in vehicle classification is not object detection nor matching but grouping similar objects into the same class. In the first experiment, we aim to observe the SIFT keypoints of different vehicles in the same class. Subsequently, object matching using SIFT algorithm is explored for vehicle classification. Geometrical constraint is introduced to the matching. Finally, the matching between vehicles from the same class and their matching to vehicles from the opposite class are compared and discussed.

5.4.1 SIFT Keypoints Observation

Vehicle classification algorithm requires generalization across intra-class variations of objects belonging to the same class. In order to see the variations of keypoints detected from vehicles of the same class, we firstly select a few vehicle images that look similar in color and shape from the training data to observe the nature of SIFT keypoints.

The SIFT keypoints detected in different octaves of the gaussian pyramid for class *Car* and *Van* are shown in Figure 5.9 and Figure 5.10 respectively.

Figure 5.9: SIFT Keypoints from Vehicles of Class Car at Different Octaves

We observe that even though the SIFT descriptors are not exactly the same for all vehicles of the same class, some patterns of the keypoints represented by the arrows across different vehicles appears to be consistent. In Figure 5.10, for example, the long arrow on the back screen of the vehicles in class *Van* is detected consistently at Octave # 4.

Figure 5.10: SIFT Keypoints from Vehicles of Class Van at Different Octaves

In the following section, we explore using the SIFT matching algorithm for vehicle classification. The matching of vehicles within the same class is compared to the matching of vehicles from different classes.

5.4.2 SIFT Keypoints Matching

We explore using the SIFT matching algorithm for vehicle classification. The matching of vehicles within the same class is compared to the matching of vehicles from different classes. We adopt Lowe’s method for object matching using SIFT descriptors in [1]. In its matching process, SIFT uses Euclidean distance to find the first and second nearest descriptors in the database. In order to discard false matches arising from background clutter, the match is only accepted if the ratio of the nearest neighbor distance to the second nearest neighbor distance is greater than 0.8.

Subsequently, a few options have been proposed for searching the nearest descriptors. The first option is to compare each descriptor against all the descriptors in the database and find the nearest one. Long search time is expected for this search option. The second option is called Best Bin First (BBF) algorithm where each descriptor is compared to at most Q nodes to find the nearest one. This option may not be the optimal solution but it is suggested that it gives good trade off between short search time and success probability.

We selected a random image from each class in the training data set to be used as the model for that class. Each keypoints of the image in test data was matched against the keypoints from the template image in each class. The number of keypoints matched are counted. This number is normalized to the total number of keypoints being matched, both from the template and test image. However, different parts of vehicles are falsely associated and matched as we can see in Figure 5.11. This may be due to the fact that different parts of vehicle have similar shapes and textures.

Therefore, instead of directly comparing the whole descriptors of the local features, geometrical constraints should be imposed to avoid poor matching. Only those keypoints which are within certain normalized distance from keypoints of the template will be considered in the matching. From our experience, we use the relative distance = 0.1 of the image size as the constraint for the keypoints matching. The matching of keypoints with this geometrical constraint is shown in Figure 5.12.

In direct comparison of all the descriptors, the numbers of matched keypoints from the test image to both Class *Van* and *Car* templates are quite similar. However, when there is geometrical constraint imposed, the number of Van keypoints being matched

Figure 5.11: Matching of a Van to another Van and to a Car without Geometrical Constraint

Figure 5.12: Matching of a Van to another Van and to a Car with Geometrical Constraint

to the template from class *Car* is largely reduced. Hence, this geometrical constraint is necessary to ensure the vehicle classification accuracy.

Further, we compare the matching between vehicles from the same class using the SIFT matching algorithm with the results of matching to vehicles from the opposite class. First of all, the SIFT descriptors from all vehicles from class *Car* and *Van* are obtained. Next, for each vehicle in class *Car*, we compare its SIFT descriptors against the descriptors of each remaining vehicle in class *Car*. The number of matched descriptors is computed and normalized to the average number of keypoints from the two cars that are being compared. This percentage value is denoted as the matching score for the two cars.

Subsequently, we take the average value of the matching scores of each Car to the rest of the Cars in the training data. This average value is denoted as the intra-class matching percentage of each Car.

In addition, the SIFT descriptors of each vehicle in class *Car* are compared to the descriptors of all vehicles in class *Van*. The number of matching descriptors is computed and normalized to get the inter-class match percentage for class *Car* against class *Van*. The same steps are repeated to get the intra-class and inter-class match percentage values for class *Van*. These two percentage values are compared in every octave level of the SIFT descriptors as shown in Figure 5.13 and Figure 5.14 for class *Car* and *Van* respectively.

Figure 5.13: Mean of SIFT Keypoints Match Percentage of Class Car Observed in Different Octaves

For the two classes of vehicles, the mean value of the intra-class matching percentage is generally higher than the inter-class matching percentage in every octave level. In other words, each vehicle of one class is more similar to the rest of vehicles in the same class compared to all vehicles in the opposite class. The SIFT descriptors can thus be used to discriminate these two classes of vehicles.

Figure 5.14: Mean of SIFT Keypoints Match Percentage of Class Van Observed in Different Octaves

Nevertheless, there are some vehicles whose inter-class matching values are higher than their intra-class matching. This type of vehicle should not be used as a reference in classifying new image during the testing stage for it will introduce error in the classification. Therefore, in the next chapter we propose a template selection method to systematically select a number of images that best represent the vehicles in the training data as the templates.

Chapter 6

Template Selection and Keypoints Reduction

In automatic model selection process, there are a few methods that can be considered. Forward selection method starts with no variable in the model. Then, variable is added to the model one at a time. At each step, each variable is tested for inclusion in the model. The most significant variable in each step will be added to the model. The steps are continued until there is no addition of significant variable to the model or until a stopping criterion is met.

Another approach is the backward selection method. The initial model consists of all the variables of interest. Subsequently, the least significant variable is excluded from the model. This step is repeated until all the variables left in the model are statistically significant. Next, the stepwise selection method alternates between forward and backward selection method. The variable is added in or removed from the model depending on the criteria set by the user for entry or removal. The selection procedure stops when a stable set of variables is obtained.

In this experiment, we use the forward selection method to determine the best templates to represent the two classes of vehicles for the classification. The whole images in the training data were considered in the template selection. Starting from an empty set of images in the template, an image was progressively added.

Meanwhile, we use the backward selection method for keypoints reduction. All keypoints obtained from the template images were considered as the set features and progressively some of these features were eliminated from the set.

6.1 Forward Template Selection Method

In the local features configuration, we use a single image as the template to represent each class of vehicle. All keypoints in this template are compared against all keypoints from the test image to classify this image into any of the two classes. However, using a single template as the model could not capture all important features to represent the object class, which can increase the accuracy of the classification.

The more templates used as the models to classify the test data, the better the classification result should be. However, we can not increase the number of templates indefinitely due to computational time needed to check all the templates for classification. Also, these templates may have similar keypoints descriptors and thus they are redundant for the classification. Hence, given a set of N images in the training data from each class of vehicle, we select m images from each class as the templates to represent the two classes of vehicles in the training dataset.

The first step in selecting multiple templates for classification is to select the first pair of images (c_1, v_1) that best represent the two classes of vehicles in the classification. From each vehicle class, X_i , an image is selected randomly and used as the template to classify the rest of images in the training data set. Subsequently, the SIFT keypoints from these two images, (c_{11}, v_{11}) , are detected. The SIFT matching algorithm with geometrical constraint mentioned in Section 5.4.2 is implemented to classify the remaining images in the training data. The classification accuracy using this pair of images is recorded. The pair is then put back into the training data set.

Subsequently, a new pair (c_{12}, v_{12}) is selected randomly as the template for classification, where $c_{12} \neq c_{11}$ and $v_{12} \neq v_{11}$. Using SIFT descriptors from this new pair, we classify the images in the training data set and record the classification accuracy. These steps are repeated for all images from class *Car*, $\{c_{13}, c_{14}, \dots, c_{1n}\}$, as well as from class *Van*, $\{v_{13}, v_{14}, \dots, v_{1n}\}$, where n is the number of images in each class of

vehicles. The pair with the highest classification accuracy is denoted as the first best pair of images (c_1, v_1) to represent the training data for classification. This pair is added to the templates and is removed from the training data in subsequent steps.

The next step is to select the second pair of images to add to the templates. A new pair of images, (c_{21}, v_{21}) , is selected randomly from the training data and added to the existing templates, $\{(c_1, v_1)\}$. The SIFT descriptors from these four images, $\{(c_1, v_1), (c_{21}, v_{21})\}$, are used to classify the remaining images in the training data. The classification accuracy is recorded and this new pair, (c_{21}, v_{21}) , is put back into the training data set.

Subsequently, another pair of images (c_{22}, v_{22}) is selected randomly, where $c_{22} \neq c_{21}$ and $v_{22} \neq v_{21}$, and added into the existing templates, $\{(c_1, v_1)\}$. The SIFT descriptors from these two pairs of images are used to classify the rest of images in the training data. Similarly, the classification accuracy is recorded and the pair of images is put back into the training data set. These steps are repeated for all images from class *Car*, $\{c_{23}, c_{24}, \dots, c_{2n'}\}$, and class *Van*, $\{v_{23}, v_{24}, \dots, v_{2n'}\}$, in the training data set, where $n' = n - 1$ is the current number of images in each vehicle class. The pair which together with the current images in the template results in the highest accuracy is kept as the second best pair (c_2, v_2) . Thus, we now have two pairs of images, $\{(c_1, v_1), (c_2, v_2)\}$, in the templates to represent the training data set.

The experiment to select the next best pair to add into the templates is repeated until there is no significant increase in the classification accuracy when another pair of images is added. After a number of tests, the images in the template data set are $\{(c_1, v_1), (c_2, v_2), (c_3, v_3), \dots, (c_m, v_m)\}$, where m is the number of pairs of images to represent each vehicle class in the training data. These templates are then used to classify images in the test data based on their SIFT descriptors.

However, it is expected that there will be several similar keypoints detected across different images from the same class in the templates. Thus, removal of these keypoints and keeping only one representative should not change the classification accuracy of the training dataset. Less keypoints also means less computation time in the matching stage. Therefore, some of redundant SIFT keypoints in the templates

are removed to increase computation efficiency of the matching process. The following section describes our proposed method to eliminate some of these less significant keypoints from the templates.

6.2 Keypoints Reduction

From the template selection step, we obtain m images as the templates to represent the training data set, namely $\{c_1, c_2, c_3, \dots, c_m\}$ from class *Car* and $\{v_1, v_2, v_3, \dots, v_m\}$ from class *Van*. The schematic diagram of the steps taken to remove some keypoints from the templates is shown in Figure 6.1.

Figure 6.1: Keypoints Reduction Schematic Diagram

Firstly, we obtain all SIFT keypoints on the images from class *Car* templates, $\{(\mathbf{x}_{c11}, \mathbf{x}_{c12}, \mathbf{x}_{c13}, \dots, \mathbf{x}_{c1k_1}), (\mathbf{x}_{c21}, \mathbf{x}_{c22}, \mathbf{x}_{c23}, \dots, \mathbf{x}_{c2k_2}), \dots, (\mathbf{x}_{cm1}, \mathbf{x}_{cm2}, \mathbf{x}_{cm3}, \dots, \mathbf{x}_{cmk_m})\}$, and from class *Van* templates, $\{(\mathbf{x}_{v11}, \mathbf{x}_{v12}, \mathbf{x}_{v13}, \dots, \mathbf{x}_{v1l_1}), (\mathbf{x}_{v21}, \mathbf{x}_{v22}, \mathbf{x}_{v23}, \dots, \mathbf{x}_{v2l_2}), \dots, (\mathbf{x}_{vm1}, \mathbf{x}_{vm2}, \mathbf{x}_{vm3}, \dots, \mathbf{x}_{vml_m})\}$, where $\mathbf{x}_{cmk_m}$ is the k_m^{th} keypoint in vehicle Car, c_m ,

and k_m is the total number of keypoints in vehicle Car, c_m . Similarly, $\mathbf{x}_{v_{ml_m}}$ is the l_m^{th} keypoint in vehicle Van, v_m , and l_m is the total number of keypoints in vehicle Van, v_m .

In addition, we obtain the SIFT keypoints from all images in the training data including all the template images. These keypoints are denoted as $\{(\mathbf{x}_{c_{11}}, \mathbf{x}_{c_{12}}, \mathbf{x}_{c_{13}}, \dots, \mathbf{x}_{c_{1k_1}}), \dots, (\mathbf{x}_{c_{m1}}, \mathbf{x}_{c_{m2}}, \mathbf{x}_{c_{m3}}, \dots, \mathbf{x}_{c_{mk_m}}), \dots, (\mathbf{x}_{c_{n1}}, \mathbf{x}_{c_{n2}}, \mathbf{x}_{c_{n3}}, \dots, \mathbf{x}_{c_{nk_n}})\}$ from class *Car* templates and $\{(\mathbf{x}_{v_{11}}, \mathbf{x}_{v_{12}}, \mathbf{x}_{v_{13}}, \dots, \mathbf{x}_{v_{1l_1}}), \dots, (\mathbf{x}_{v_{m1}}, \mathbf{x}_{v_{m2}}, \mathbf{x}_{v_{m3}}, \dots, \mathbf{x}_{v_{ml_m}}), \dots, (\mathbf{x}_{v_{n1}}, \mathbf{x}_{v_{n2}}, \mathbf{x}_{v_{n3}}, \dots, \mathbf{x}_{v_{nl_n}})\}$ from class *Van* templates, where n is the total number of vehicles in each class in the training data, k_n is the total number of keypoints in vehicle Car, c_n , and l_n is the total number of keypoints detected in vehicle Van, v_n .

Each SIFT keypoint, $\mathbf{x}_{c_{nk_n}}$ or $\mathbf{x}_{v_{nl_n}}$, has 128 descriptor elements, the locations of the keypoints, as well as the major orientation and scale. In our experiment, the locations of the keypoints were normalized to the dimensions (i.e. the width and height) of the image from where the keypoints are detected.

Any keypoint that is highly discriminative should be kept in the template. However, those keypoints that may cause confusion in the classification may be removed. In order to remove these keypoints from the templates we compare the similarity between the templates keypoints to the keypoints from all training images of the opposite class. The similarity is measured using the Euclidean distances between any two SIFT keypoints descriptors.

The inter-class distance between keypoint, $\mathbf{x}_{c_{mk_m}}$, from vehicle Car templates to all keypoints, $\mathbf{x}_{v_{nl_n}}$, from all vehicles of class *Van* in the training data is denoted as $\mathbf{d}_{c_{mk_m}, v_{nl}}$. Subsequently, we compute the minimum of these distances and use this value as the distance threshold for keypoint $\mathbf{x}_{c_{mk_m}}$ as shown below:

$$d_{c_{mk_m}, th} = \arg \min_{l_n} (\mathbf{d}_{c_{mk_m}, v_{nl}}) \quad (6.1)$$

where $d_{c_{mk_m}, th}$ denotes the threshold value for keypoint $\mathbf{x}_{c_{mk_m}}$ and l_n is the total number of keypoints from images in class *Van*.

Furthermore, the distance from each keypoint $\mathbf{x}_{c_{mk_m}}$ to every keypoint in class *Car* templates, $\mathbf{x}_{c_{m'k_{m'}}}$, is computed. These distances are the intra-class distances

of the keypoint, $\mathbf{d}_{c_{mk_m}c_{m'k_{m'}}$. Subsequently, we count the number of keypoints in class *Car* templates whose intra-class distance to keypoint, $\mathbf{x}_{c_{mk_m}}$, is smaller than the distance threshold, $d_{c_{mk_m},th}$. This count is assigned to the keypoint as its index number, $i_{c_{mk_m}}$. We repeat these steps for all keypoints in class *Car* templates. The keypoint with the highest index number, $i_{c_{mk_m}}$, is removed from the keypoints set in class *Car* templates.

In addition, we carry out the same steps on all keypoints in class *Van* templates to remove one keypoint with the highest index number, $i_{v_{ml_m}}$, from keypoints set in class *Van* templates. After the first keypoint from each class is removed, the above steps were repeated to remove the next keypoint from each class templates. The stopping criteria for the keypoints removal is if the highest index numbers for both class are equal to zero, i.e. there is no keypoint in the same class whose distance is smaller than the minimum distance to the opposite class. This also implies that the current set of keypoints are necessary to discriminate the two classes of vehicles.

Lastly, the remaining keypoints in each vehicle class templates will be compared against all keypoints in the test image for vehicle classification based on local features.

Chapter 7

Experimental Results and Discussions

The experimental results are organized as follow: First of all, we classified the vehicles in the test data set using only their structural configuration. Next, we used SIFT matching algorithm that has been used for exact matching of objects to classify the vehicles into class *Car* and *Van*. One image was selected from each class in the training data as the templates. We compared the classification results using geometrical constraint with the one without geometrical constraint.

Subsequently, the images in the test data were classified using our proposed hierarchical method. Vehicle images were firstly classified based on the structural configuration into three categories, namely *Car*, *Van*, and *Uncertain*. Vehicles in *Uncertain* category were classified further using SIFT keypoints in the local features stage. More templates were used to classify the images in the local features stage. These templates were selected from the training data using our proposed template selection method. We then compare the classification accuracy using more templates and classification using a single template in the local features stage of our proposed hierarchical method.

To further test the performance of our proposed template selection method, we classify the images in the test data using the same number of templates selected randomly from the training data and compare the classification result with our proposed

template selection method.

Lastly, we presented the classification accuracy of the training data in each step of the keypoints reduction method. The classification accuracy using our keypoints reduction method with the classification results using random removal of the keypoints were presented and compared.

We collected 1500 images of vehicles of class *Car* (1000 images) and *Van* (500 images) from a traffic surveillance system for our experiment. Out of this pool of data, 100 images from each class (*Car* and *Van*) are used as the training data. The remaining 1300 images (900 images for class *Car* and 400 images for class *Van*) are used as testing data.

7.1 Structural Configuration Method

Firstly, we classified all vehicle images in the test data set based on their structural configuration. The structural configuration for each class is described using the Gaussian mixtures model described earlier. The classification of vehicle images in the test data is based on the maximum sum of the posterior probabilities of all node locations, x_n , detected from the test image. The confusion matrix for vehicle classification based on the structural configuration is shown in Table 7.1.

	Car	Van
Car	93.00%	7.00%
Van	25.50%	74.50%

Table 7.1: Classification Results using the Structural Configuration

The accuracy using only structural configuration for classification is 87.31%. Even though the classification accuracy using the proposed structural configuration method is reasonably good, the misclassification rate for vehicle of class *Van* is rather high. In the next section, we classified the vehicle images in the test data based on their local features.

7.2 Local Features Method

Next, the test images were classified based on their local features using SIFT matching algorithm. We tested the algorithm using an image from each class of the training data as the template. For each test, one image from class *Car* and one image from class *Van* were selected randomly. The classification accuracy for 100 tests is shown in Figure 7.1. For this experiment we did not impose the geometrical constraint on the keypoints matching. The mean value of the classification accuracy is only 49.94%, which is only as good as random classification of the images.

Figure 7.1: Classification based on Local Features using a Single Template without Geometrical Constraint

Then, we carried out experiment with geometrical constraint where only those keypoints in the test images that are within relative distance = 0.1 from each keypoint in the template are considered for the matching. The results can be seen in Figure 7.2. The mean value of the classification accuracy with imposing geometrical constraint is 77.54%, which is higher compared to matching the keypoints without any geometrical constraint as in the previous stage (Figure 7.1). Therefore, imposing geometrical constraint on the matching of keypoints increases the accuracy of vehicle classification based on SIFT descriptors as the local features.

Figure 7.2: Classification based on Local Features using a single Template with Geometrical Constraint

However, the average value of the classification accuracy is lower compared to using the structural configuration, which is 87.31%. In the following section, we classified the vehicles in the test data using both structural and local features in hierarchical manner.

7.3 Hierarchical Method

In this last section, we classified the vehicles using our proposed hierarchical method. In the first stage, vehicles in the test data set were classified using the structural configuration. The vehicles in the Uncertain group are further classified using the SIFT matching algorithm. Keypoints are extracted from the templates obtained from the training data and compared to the keypoints from the test image in the local features stage.

7.3.1 Hierarchical Method with Template Selection

We apply the template selection method described in Section 6.1 to our training data. The classification accuracy is increased as the number of image pairs in the templates is increased as shown in Figure 7.3.

Figure 7.3: Vehicle Classification Accuracy for Template Selection

We select $m = 5$ images from the training data as the templates based on the least number of images required to achieve maximum classification accuracy. The SIFT descriptors from these five pair of images will then be used to classify the vehicles in the test data set. The classification accuracy is 96.23% (Table 7.2). The classification result using hierarchical method is better compared to only the structural configuration (accuracy = 87.31%) or based on local features (accuracy = 93%).

	Car	Van
Car	95.44%	4.56%
Van	2%	98%

Table 7.2: Classification Results using Five Templates

Further, a systematic selection of templates is expected to give better accuracy in classifying the vehicles in the test data set compared to a random selection of m

templates from N images in the training data. In order to assess the performance of our template selection method, we classify the test data using the same number of images that are selected randomly from the training data set.

We ran 100 random selections of five templates and used these templates to classify the images in the training data using the hierarchical method. Subsequently, we used these random selections of templates in the local features stage of our hierarchical method to classify the test images. The classification accuracy for each random selection can be seen in Figure 7.4.

Figure 7.4: Classification Accuracy using Proposed Template Selection Method Compared with using 100 Random Selection of Five Templates

The mean value of the classification accuracy for 100 random templates selection is 93.57%. The classification accuracy using our proposed template selection method is thus better compared to random selection of the same number of templates from the training data.

In the following section, some of redundant keypoints from these five templates were removed systematically while maintaining the classification accuracy in the training data. The remaining keypoints were then used to classify the images in the test data.

7.3.2 Hierarchical Method with Keypoints Reduction

The objective of this experiment is to remove n keypoints using the proposed keypoints reduction method. Firstly, all keypoints from the templates obtained from the template selection method were extracted. There are 269 keypoints for the five templates of class *Car* and 312 keypoints for templates of class *Van*. In each test, one keypoint is removed from each class template (Figure 7.5).

Figure 7.5: Vehicle Classification Accuracy for Keypoints Reduction

At keypoints index = 150, the classification accuracy of the remaining training data is about 95.8%, which is only about 4% lower compared to using all of the keypoints to classify the same set of data (accuracy = 100%). At this keypoints' index, the number of keypoints being removed from class *Car* is 57 keypoints and from class *Van* is 150 keypoints. The total keypoints removed is about 35% of the templates' keypoints. Thus, we can remove about 35% of the keypoints from the templates with only slight decrease in the classification accuracy of the training data.

Next, we need to test the performance of the proposed keypoints reduction method on the test data. We compared the remaining 65% keypoints from the selected templates with all SIFT keypoints from each test image in the local features stage of the hierarchical classification. The confusion matrix of the classification is shown in Table 7.3.

The classification accuracy using our proposed keypoints reduction method is 94.62%, which is only lower by 1.67% compared to using all keypoints in the templates (96.23%).

	Car	Van
Car	93.89%	6.11%
Van	3.75%	96.25%

Table 7.3: Classification Results using Reduced Number of Keypoints

Further, we compare the performance with random removal of the same number of keypoints from the templates. The results can be seen in Figure 7.6. The mean value of the classification accuracy using random keypoints reduction is 93.37%. Removing keypoints using our proposed keypoints reduction method results in higher classification accuracy compared to using random reduction of keypoints.

Figure 7.6: Classification Accuracy using Proposed Keypoints Reduction Method Compared with using 100 Random Reduction of Keypoints

Chapter 8

Conclusions and Future Works

8.1 Conclusions

We have proposed a hierarchical method for vehicle classification system that incorporates both structural and local features of an object. Firstly, we observe the discriminative structural features of vehicles viewed from a camera placed on an overhead bridge monitoring the highways. The structural configuration of the vehicle finds the major planes in the vehicles based on the gradient of the color intensity values along vertical direction. Subsequently, a Gaussian mixture model for each class of vehicle in the training data is built from all the locations of the major planes detected from all vehicles in the class. The classification of images in this stage is based on sum of the posterior probabilities of all node locations of the major planes detected in the image.

Next, vehicles are classified using local features described using SIFT descriptors. We investigate the feasibility of using SIFT descriptors as the local features for vehicle classification. The classification is based on the maximum number of keypoints matched to a single template that is manually selected to represent each class of vehicle. Geometrical constraint imposed on the matching of SIFT keypoints improved the classification accuracy.

Further, we tested our proposed hierarchical method to classify the images in the training data. The classification result using hierarchical method is better compared to classifying the vehicles only based on their structural configuration or local features only. Subsequently, we extend our algorithm by selecting multiple templates to represent each class of vehicle. We propose a template selection method to select best representative images from each vehicle class in the training data. Classification using these templates obtained from our template selection method gives better accuracy compared to using only a single template or using the same number of templates but selected randomly from each class of vehicle.

Lastly, we remove a number of keypoints from the selected templates systematically while maintaining the classification accuracy of the training data when all of the keypoints were used. The remaining keypoints from these templates were then used to classify the test data in the local features stage. Using only about 65% of the keypoints, we do not change much the classification result of the training data as well as the result for the test data.

8.2 Future Works

In this work, we have presented a hierarchical method to classify vehicles into two classes, namely Car and Van, using structural and local features. This method should be extended further to more classes of vehicles such as Truck, Bus, Hatchback, Multi-purpose Vehicle (MPV), Motorcycle, etc.

In addition, the data used in this experiment is collected from a single camera placed on an overhead bridge across the highway. From this camera view, the structural configuration of different types of vehicles may be distinguished based on the number of major building parts of the vehicles. The future work may explore on other camera views, such as the side or front view. However, the structural configuration based on the major building parts may not be applied directly due to the particular structures of vehicles viewed from different angles.

Furthermore, object modeling in this work is based on hierarchical model of vehicles, which consists of structural configuration as the major building parts and the

local features as the discriminating details of different types of vehicles. This model may be used to build more comprehensive templates for different models of certain vehicle classes. Also, extending this hierarchical model into different object classes will be an interesting topic to explore in the future.

Publications

Submitted Paper:

- S.Y. Anggrelly, Xudong Jiang, and How-Lung Eng, “Vehicle classification based on structural and local features,” *Pattern Analysis and Applications*, 2010.

Bibliography

- [1] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.
- [2] K. Murphy, A. Torralba, D. Eaton, and W. Freeman, “Object detection and localization using local and global features,” *Towards Category-Level Object Recognition*, vol. 1, 2005.
- [3] Daniela Hall and James L. Crowley, “Computation of generic features for object classification,” in *Scale-Space*, 2003, pp. 744–756.
- [4] K.Mikolajczyk and C.Schmid, “An affine invariant interest point detector,” in *Proc. the 7th European Conference on Computer Vision*, 2002, vol. 1, pp. 128–142.
- [5] Cordelia Schmid, Roger Mohr, and Christian Bauckhage, “Comparing and evaluating interest points,” in *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*. January 1998, IEEE Computer Society Press.
- [6] Krystian Mikolajczyk and Cordelia Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [7] Yi Yang and Shawn Newsam, “Comparing sift descriptors and gabor texture features for classification of remote sensed imagery,” in *ICIP*, 2008, pp. 1852–1855.

- [8] Gy. Dorkó and C. Schmid, “Selection of scale-invariant parts for object class recognition,” in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, 2003, vol. 1, p. 634, IEEE Computer Society.
- [9] Brian Ayers and Matthew Boutell, “Home interior classification using sift key-point histograms,” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 1–6, 2007.
- [10] Kah-Kay Sung and Tomaso Poggio, “Example-based learning for view-based human face detection,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, January 1998.
- [11] Constantine Papageorgiou and Tomaso Poggio, “A trainable system for object detection,” *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.
- [12] Shivani Agarwal and Dan Roth, “Learning a sparse representation for object detection,” in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, London, UK, 2002, pp. 113–130, Springer-Verlag.
- [13] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio, “Example-based object detection in images by components,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 349–361, 2001.
- [14] T. K. Leung, M. C. Burl, and P. Perona, “Finding faces in cluttered scenes using random labeled graph matching,” in *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, Washington, DC, USA, 1995, p. 637, IEEE Computer Society.
- [15] Bernd Heisele, Tomaso Poggio, and Massimiliano Pontil, “Face detection in still gray images,” in *A.I. memo 1687, Center for Biological and Computational Learning*, 2000.

- [16] Paul Viola, “Complex feature recognition: A bayesian approach for learning to recognize objects,” Tech. Rep., Cambridge, MA, USA, 1996.
- [17] Shimon Ullman and Erez Sali, “Object classification using a fragment-based representation,” in *BMVC '00: Proceedings of the First IEEE International Workshop on Biologically Motivated Computer Vision*, London, UK, 2000, pp. 73–87, Springer-Verlag.
- [18] Boris Epshtein and Shimon Ullman, “Feature hierarchies for object classification,” in *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Washington, DC, USA, 2005, pp. 220–227, IEEE Computer Society.
- [19] Brian Leung, “Component-based car detection in street scene images,” M.S. thesis, Massachusetts Institute of Technology, 2004.
- [20] Julia Vogel and Bernt Schiele, “Natural scene retrieval based on a semantic modeling step,” in *ACM International Conference on Image and Video Retrieval*. 2004, pp. 207–215, Springer Verlag.
- [21] Fei-Fei Li and Pietro Perona, “A bayesian hierarchical model for learning natural scene categories,” in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 524–531.
- [22] Eran Borenstein and Shimon Ullman, “Class-specific, top-down segmentation,” in *Proceedings of the 7th European Conference on Computer Vision-Part II*, London, UK, 2002, vol. 2351/2002, pp. 109–124, Springer-Verlag.
- [23] Bastian Leibe and Bernt Schiele, “Interleaved object categorization and segmentation,” in *British Machine Vision Conference*, 2003, pp. 759–768.
- [24] M. A. Fischler and R. A. Elschlager, “The representation and matching of pictorial structures,” *IEEE Transactions on Computers*, vol. 22, no. 1, pp. 67–92, 1973.

- [25] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, “Pictorial structures for object recognition,” *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, January 2005.
- [26] M Weber, *Unsupervised Learning of Models for Object Recognition*, Ph.D. thesis, California Institute of Technology, 2000.
- [27] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003, vol. 2, pp. 264–271.
- [28] T. Kadir and M. Brady, “Scale, saliency and image description,” *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, November 2001.
- [29] Bernd Heisele, Thomas Serre, Massimiliano Pontil, Thomas Vetter, and Tomaso Poggio, “Categorization by learning and combining object parts,” in *in NIPS*. 2001, pp. 1239–1245, MIT Press.
- [30] S. Gupte, O. Masoud, and P. Papanikolopoulos, “Vision-based vehicle classification,” in *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE*, October 2000, pp. 46–51.
- [31] Stefan Hinz, “Integrating local and global features for vehicle detection in high resolution aerial imagery,” in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2003, pp. 119–124.
- [32] Surendra Gupte, Osama Masoud, Robert F. K. Martin, and Nikolaos P. Papanikolopoulos, “Detection and classification of vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, pp. 37–47, 2002.
- [33] Omar Javed and Mubarak Shah, “Tracking and object classification for automated surveillance,” in *ECCV ’02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, London, UK, 2002, pp. 343–357, Springer-Verlag.
- [34] Brendan Morris and Mohan Trivedi, “Improved vehicle classification in long traffic video by cooperating tracker and classifier modules,” in *AVSS ’06: Proceedings*

- of the IEEE International Conference on Video and Signal Based Surveillance*, Washington, DC, USA, 2006, p. 9, IEEE Computer Society.
- [35] Xia Limin, “Vehicle shape recovery and recognition using generic models,” in *Proceedings of the 4th World Congress on Intelligent Control and Automation*, 2002, vol. 2, pp. 1055–1059.
- [36] Dieter Koller, “Moving object recognition and classification based on recursive shape parameter estimation,” in *Proc. 12th Israeli Conf. Artificial Intelligence, Computer Vision, and Neural Network*, 1993, pp. 359–368.
- [37] G.D. Sullivan, A.D. Worrall, and J.M. Ferryman, “Visual object recognition using deformable models of vehicles,” 1995, p. xx.
- [38] Geoff Sullivan, “Model-based vision for traffic scenes using the ground-plane constraint,” pp. 93–115, 1995.
- [39] W. Wei, Q. Zhang, and M. Wang, “A method of vehicle classification using models and neural networks,” in *IEEE Conference on Vehicular Technology*, 2001, vol. 4, pp. 3022–3026.
- [40] Tatsuya Yoshida, Shirmila Mohottala, Masataka Kagesawa, and Katsushi Ikeuchi, “Vehicle classification system with local-feature based algorithm using cg model images(special issue on information system technologies for its),” *IEICE Transactions on Information and Systems*, vol. 85, no. 11, pp. 1745–1752, 2002.
- [41] Xiaoxu Ma and W.E.L. Grimson, “Edge-based rich representation for vehicle classification,” in *IEEE International Conference on Computer Vision*, 2005, vol. 2, pp. 1185–1192.
- [42] How-Lung Eng, M. Thida, Boon-Fong Chew, K. Leman, and S.Y. Anggrelly, “Model-based detection and segmentation of vehicles for intelligent transportation system,” in *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, 3-5 2008, pp. 2127 –2132.

- [43] L. G. Roberts, “Machine perception of three-dimensional solids,” *Optical and Electro-optical Information Processing*, pp. 159–197, 1965.
- [44] Mourad Zerroug and Ramakant Nevatia, “From an intensity image to 3-d segmented descriptions,” in *ECCV '96: Proceedings of the International Workshop on Object Representation in Computer Vision II*, London, UK, 1996, pp. 11–24, Springer-Verlag.
- [45] Sven Dickinson, “The evolution of object categorization and the challenge of image abstraction,” in *Object Categorization: Computer and Human Vision Perspectives*, Sven Dickinson, Aleš Leonardis, Bernt Schiele, and Michael J. Tarr, Eds., pp. 1–37. Cambridge University Press, 2009.
- [46] David G. Lowe, “Object recognition from local scale-invariant features,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. August 1999, vol. 2, pp. 1150–1157 vol.2, IEEE Computer Society.