OBSTACLE DETECTION FRAMEWORK IN HYBRID CLASSIFICATION OF HETEROGENEOUS ENVIRONMENT BASED ON DATA MINING TECHNIQUES

A Thesis

SUBMITTED TO THE

TILAK MAHARASHTRA VIDYAPEETH,
PUNE

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

In Computer Science

Under the Board of Modern Sciences & Professional Skills



BY

Mrs. Padma Nilesh Mishra
(25315008502)

UNDER THE GUIDANCE OF

Dr. ANUP GIRDHAR

DEPARTMENT OF COMPUTER SCIENCE

SEPTEMBER 2021

CERTIFICATE BY THE SCHOLAR

This is to certify that the thesis titled, "Obstacle Detection Framework In Hybrid Classification Of Heterogeneous Environment Based On Data Mining Techniques" submitted by Mrs. Padma Nilesh Mishra, under the supervision of Dr. Anup Girdhar, P.R.N. Number 25315008502 for the award of Ph.D. degree of Computer science from 'Tilak Maharashtra Vidyapeeth' University embodies my original work and has not formed the basis for the award of any degree, diploma, fellowship, titles in this or any other university or any other similar institutions of higher learning.

Date: Signature

Mrs. Padma Nilesh Mishra,

Research Scholar, TMV.

Enrollment Number 25315008502

CERTIFICATE BY THE SUPERVISOR

This is to certify that the work incorporated in the thesis “Obstacle Detection Framework in Hybrid Classification of Heterogeneous Environment Based on Data Mining Techniques” submitted by Mrs. Padma Nilesh Mishra is an original piece of research work carried out by the candidate under my supervision. Literary presentation is satisfactory and the thesis is in a form suitable for publication. Work evinces the capacity of the candidate for critical examination and independent judgment. Such materials as has been obtained from other sources have been duly acknowledged in the thesis.

Date: Signature

Dr. Anup Girdhar, Ph.D.

DECLARATION

I hereby declare that this Ph.D. thesis entitled "Obstacle Detection Framework in Hybrid Classification of Heterogeneous Environment Based on Data Mining Techniques" was carried out by me for the degree of Doctor of Philosophy in Computer Science under the guidance and supervision of Dr. Anup Girdhar, Tilak Maharashtra Vidyapeeth, Pune, India. The interpretations put forth are based on my own experimentation and understanding. The books, articles and websites, which I have made use of are acknowledged at the respective place in the text. All content is plagiarism free. All the information gained is from genuine sources and the results are obtained following the detailed procedure. For the present thesis, which I am submitting to the University, no degree or diploma or distinction has been conferred on me before, either in this or in any other University.

Date: Signature

Mrs. Padma Nilesh Mishra,

Research Scholar, TMV.

Enrollment Number 25315008502

ACKNOWLEDGEMENTS

I wish to sincerely thank all those who have contributed in one way or another to this study. Words can only inadequately express my deep gratitude to my guide, **Dr. Anup Girdhar,** for his meticulous care, kindness and generosity. His fruitful comments and insightful suggestions have been a crucial formative influence on the present study. He has supported me in every possible way since the beginning of my research. His critical and careful reading of my writing has saved me from a lot of errors. Without his guidance and encouragement, my research would have never come out in the present form. I have seen in him an unpretentious and devoted scholar. Furthermore, it has been a memorable and enjoyable experience for me to work with him. Sincere thanks to **Mr. Nilesh Mishra** and my daughter **Jaynie Mishra** for cooperating with me and helping me at every stage. I wish to express my sincere gratitude to **Dr. Sunanda Yadav,** the Director of the place of research, for her expert guidance and invaluable suggestions.

Mrs. Padma Nilesh Mishra,

Research Scholar, TMV.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction to the Study……..…...1

1.2 Obstacle Detection Mechanism…….…...9

1.2.1 Appearance-based Methods for Obstacle Detection ……....9

1.2.2 Methods based on motion ……..15

1.3 Obstacle as Vehicle Detection Challenges…….……17

1.3.1 Camera Motions…….……...……..17

1.3.2 Camouflage ……..…….….18

1.3.3 Clutter ……..……..18

1.3.4 Jammed Road traffic ……..…...18

1.3.5 Dissimilar Forms of Automobiles…….…..…18

1.3.6 Dynamic Changes……..….18

1.3.7 Low Light Conditions ……19

1.3.8 Multiscale Problem…….....19

1.3.9 Noise …….....19

1.3.10 Obstacle Motion Speed ……..20

1.3.11 Stereo Matching ……20

1.3.12 Sudden Changes in Illumination …….…….20

1.3.13 Obstacle as Vehicle Orientation……...20

1.3.14 Obstacle as Vehicle Shadows……...21

1.4 Fully Automated Data Processing……..……23

1.5 Intelligent Transportation Systems…….....24

1.6 Automated Obstacle as Vehicle Monitoring…….25

1.7 Reasons for Using Computer Vision for Obstacle as Vehicle Detection…….....26

1.8 Reasons for Using Machine Learning for Obstacle as Vehicle Detection…….....27

1.9 Reasons for Using Data mining for Obstacle as Vehicle Detection ……..28

1.9.1 Image classification for obstacle as vehicle detection…….29

1.10 Autonomous Vehicles and Driver Assistance Systems……30

1.11 Obstacle Detection Model System……....31
1.12. Weather and Obstacle Detection Statistics……..32
1.13. Process of Obstacle as Vehicle Analysis……...…..33
1.14 Recognition and detection of obstacle as Vehicle Shapes……....35
1.15. Problem Statement…….…..…35
1.16 Challenges and Issues…….…..….36
1.17 Objective and Contributions……...…….….38
1.18 Motivation for the Study……..…….....39
1.19 Structure of the Thesis…….....40
1.20 Keywords used…….…….....41

CHAPTER 2: LITERATURE REVIEW

2.1 Analysis of Literature Survey…….....42
2.2 Obstacle as Vehicle Detection…….…....42
2.2.1 Appearance Based Methods……..….47
2.2.1.1 Feature based model……....49
2.2.1.2 Part Based Model…….……...49
2.3 Vehicle Classification…….……...51
2.4 Vehicle Tracking…….…..53
2.5 Machine Learning /Data mining…….……...53
2.5.1 Machine learning as a tool in data mining…….54
2.5.2 Types……..54
2.5.3 Data Mining Techniques……...….55
2.6 Gap Analysis from the Literature Review…….....61

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction Research Methodology……..……..…….…….....62

3.1.1 Objective of the Research Study……..……..…….……..…..62

3.1.2 Significance of the Research Study ……..……..…….....….63

3.1.3 Need of the Research Work……..……..……..…….…..63

3.1.4 Benefit of Research Work……..……..……..…….…….....63

3.2 Research Approaches……..……..…….....64

3.2.1 Mixed Approach……..……..……..……..…...…...64

3.3 Collection of Datasets……..……..……..…….…..92

3.3.1. Study on Existing Datasets ……..……..……...……....93

3.3.2 Testing the Existing Datasets……..……..…….…….....95

3.3.3 Proposed Heterogenous Dataset……..……..……..…….…….....98

3.4 Proposed Framework for Obstacle Detection Model System……..……..106

3.4.1 Obstacle as Vehicle Detection……..……..……..……..……..111

3.4.2 Extraction from the region of interests(ROI)…..…..…..…….…….....131

3.4.3 Techniques for Feature Extraction……..……..……..……..134

3.4.4 Global Representation Techniques ……..……..……..……...…...143

3.4.5 Approaches proposed by BoF-ODMS……..……..……..……...151

3.4.6 Classification Techniques……..……..……..……....159

CHAPTER 4: OBSTACLE DETECTION MODEL SYSTEM RESULTS AND PERFORMANCE EVALUATION

4.1 Performance Metrics……..……..……..……..……...184

4.1.1 Speed……..……..……..……..……...184

4.1.2 Accuracy……..……..……..……..……..184

4.1.3 Decision Capability……..……..……..……..……188

4.2 Best Parameters Selection……..……..……..…….…188

4.2.1 Improved Dictionaries…..……..……..……..……189

4.2.2 Optimized Classifiers…..……..……...……..……193

4.3 Obstacle as Vehicle Region of Interest…..……..……..……...205

4.4 Performance of SSVM-based BoF-ODMS…..……..……205

4.4.1 Accuracy…..……..……...…….208

4.4.2 Speed…..……..……...…211
4.5 Comparison of ODMS BoF-SD and ODMS -BoF-MD……...……212
4.5.1 Accuracy and Speed……..…212
4.5.2 Dictionary Training Time……213
4.6 Computation Time for Feature Extraction and Global Feature Representation for ODMS…….215
4.6.1 Histogram of Oriented Gradient Based Gist Feature (HOG-gist) Extraction for ODMS…….215
4.6.2 GIST Feature Extraction……...…219
4.6.3 SIFT Feature Extraction……..……220
4.7. Random Forest……...…….....227
4.8. Results Using ODMS- Ensemble of Multi-Class Support Vector Machine…..…..…236
4.9. Explanations on ODMS Performance……...……..243
4.10. Results for Obstacle detection Model System Using Ensemble of Multi-class SVM Classifiers……...……249
4.11. Misperception Matrix besides Class Recognition Percentage ……...…….…….251
4.12. Proposed Fast Multiclass Classification……...…255
4.12.1 Proposed Divide and Conquer Support Vector Machine (DCSVM) for ODMS…….....256
Chapter 5 Finding of Heterogenous Model, Conclusion and Future Work
5.1 Finding of Heterogenous Model…….……...……264
5.2 Limitations of the Study……...……...……..…..281
5.3 Conclusion & Future work……..……..……...……..281
REFERENCES……...……285

LIST OF TABLES

Table 1. 1. Proves in what manner color data remains castoff popular Obstacle as car illuminations proof of identity.

Table 1.2 Demonstrates a selection of the portion in mining-classifier mixtures

Table 1.3 Demonstrates a review of the vehicle as far as obstacle detection and tracking are concerned.

Table 2.1: Demonstrates the Representative Works Vision-Based Vehicle Detection

Table 2.2: Demonstrates the Attributes Used for Classification

Table 3.1 Qualitative obstacle comparison with approaches to vehicle detection.

Table 3.2 Summary of current directions and open issues

Table 3.3 Demonstrates the summary of the existing datasets addressing the task

Table 3.4 Demonstrates the Number of Images in Each Dataset.

Table 3.5: Demonstrates the Classification results for the models trained on different datasets.

Table 3.6 Demonstrations Comparison with the existing datasets with the proposed heterogenous Dataset.

Table 3.7 Demonstrations of Total Data Used for the Research Study

Table 3.8. Demonstrations some samples of training and testing dataset.

Table 3.9: Demonstrations the Typical Values1 for Highway Obstacle as vehicle Detection.

Table 3.10: Demonstrations the Typical Values2 for Highway Obstacle as vehicle Detection

Table 3.11: Terminology

Table 4.1 SURF and SSVM performance of BoF-SD (C-SVC)

Table 4.2: SURF and SSVM performance of BoF-MD (C-SVC)

Table 4.3: Calculation of computation time for HOG and GIST (seconds/100 images)

Table 4.4: Calculation of Computation time for SIFT and SURF (seconds/100 images)

Table 4.5 Demonstrates the RF-ODMS testing time (seconds/100 images)

Table 5.1 Demonstrates the Total Accuracy rate of the Hybrid Classification Techniques.

Table 5.2 Comparation of Proposed models with Accuracy and Performance

LIST OF FIGURES

Figure 1.1 (a)Demonstrates the Road accidents in the United States are roughly five times higher than in India. (b)India has four times the number of deaths as the United States.
Figure 1.2 Demonstrates the Word Statics Reports
Figure1.3. Demonstrates the aftermath of a tunnel traffic accident
Figure1.4. Demonstrates the Maharashtra State Highway Police statistics on death accidents, critically injured accidents, and minor injury accidents are displayed.
Figure1.5. Demonstrate the Fatalities-2019 data from Maharashtra's several
Figure1.6. Demonstrate the Process of Obstacle as Vehicle Analysis
Figure 1.7 Demonstrates the vehicles in various lighting and weather conditions.
Figure 3.1 Survey Form for Preparedness for Obstacle Detection Issues under heterogenous Climate
Figure. 3.2 Represents Road Weather, Visibility Issues in Express Highways causes for fatal accident.
Figure. 3.3. The growing number of object detection publications.
Figure.3.4. Demonstrates the Object Detection Route Map & Milestone detectors
Figure.3.5. Demonstrate the accuracy improvements of object detection on VOC07, VOC12 and MS-COCO datasets. Detectors.
Figure.3.6. Detector of objects from 2001 to 2019 in multi-scale evolution:1) feature pyramids and sliding windows, 2) detection with object proposals, 3) deep regression, 4) multi-reference detection, and 5) multi-resolution detection.
Figure 3.7. Demonstrates the Evolution of bounding box regression techniques in object detection from 2001 to 2019

Figure.3.8. Demonstrates the Evolution of context priming in object detection from 2001 to 2019: 1) detection with local context, 2) detection with global context, 3) detection with context interactives.
Figure.3.9. Demonstrate the Evolution of non-max suppression (NMS) techniques in object detection from 1994 to 2019: 1) Greedy selection, 2) Bounding box aggregation, and 3) Learn to NMS.
Figure 3.10 Demonstrates the sub-classes of a selected make the distribution of images is displayed
Figure 3.11 Demonstrates Distribution in CompCars & Stanforddb db- 51 of the number of images per model year.
Figure 3.12. Demonstrations the Number of top 20 images and models in the internet dataset collected.
Figure 3.13: Demonstrates the Quantitative and Qualitative data for the Research Study.
Figure 3.14. Demonstrations the Processes of Proposed Datasets
Figure 3.15.(a) Proposed Framework for Obstacle Detection Model System for Cities
Figure 3.15.(b) Proposed Framework for Obstacle Detection Model System for Highways.
Figure 3.16. Demonstrates the Concept of approaching obstacle as vehicle detection.
Figure 3.17 Demonstrates the Proposed approach for Obstacle as vehicle detection
Figure 3.18. Demonstrates vertical height a single measurement subtends, dz, is connected to the vertical angular resolution $\Delta\theta$.
Figure 3.19. Demonstrates the design specification for the highway states that the driver must be able to see an object height p = 0.5 feet on the road at distance L at a height h = 3.5 feet over the road surface.
Figure 3.20. Demonstrates The camera is normally inclined at an angle a distance L, to cross a planar road. The assumed lookahead distance is nominal surface L
Figure 3.21. Demonstrates the Left-right pair of stereos simulated.

Figure 3.22. Demonstrates the When the right image twist through the correct disparity function, result as the left image of the pair that looks like the original stereo pair's left one (the white area at the bottom-left indicates pixels where no match was possible).
Figure 3.23: Demonstrates the Region of Interest Extraction of Obstacle as Car for ODMS
Figure 3.24: Demonstrates the Incorrectly Computed ROI Data Stored in Trained Dataset
Figure 3.25: Demonstrates the generation step Scale-Space to get DoG images for an input image.
Figure 3.26: Demonstrates the SIFT descriptor computation around a key point.
Figure 3.27. Demonstrations the one-block SURF descriptor computing method in the area centered around the key.
Figure 3.28: Demonstrates the Computing the HOG descriptor over an input image or ROI for one of the sliding windows.
Figure 3.29: Demonstrates the (a) The LBP binary line for a pixel is computed, taking into account its 3x3 neighborhoods: Pixel-based comparisons are made; (b) A 9 / 9 operator MB-LBP, with a total of 8 sub-regions in a central area: comparisons between average sub-regional grey value.
Figure 3.30: Demonstrates the Grid-based Representation of obstacle as vehicle models: (a) Direct Division scheme; (b) Indirect Division scheme.
Figure 3.31: Demonstrates the two Direct Division methods: (a) Wide Direct Division (WDD); (b) Narrow Direct Division (NDD).
Figure 332: Demonstrates the Two Inside Division method (a) Symmetrical Inside Division (SID); (b) Full Inside Division (FID).
Figure 3.33: Demonstrates Different patterns of block combination: (a) Combining every two (1 × 2) adjacent blocks; (b) Combining every four (2 × 2) adjacent blocks .
Figure 3.34: Demonstrates the Fisher Vector A key Point-based patch descriptor computing pipeline.

Figure 3.35: Demonstrates the Cootes and models make Petrovic and Petrovic and Cootes features extraction and global representatives of vehicles.
Figure 3.36: Demonstrates the Feature's extraction and global representations of obstacle as vehicle
Figure 3.37: Demonstrates Overview of the BoF-based ODMS approaches proposed.
Figure 3.38: Demonstrates the Offline Dictionary Building in our BoF-ODMS approaches: (a) Single-Dictionary Building Scheme, (b) Modular Dictionary Building Scheme
Figure 3.39: Demonstrates A hyperplane P which divides two class 2D data points. The circles are equivalent to class L1 data points, while the diamonds are equivalent to those of label +1.
Figure 3.40: Demonstrates A the hyperplanes $P_{\_}$, P, and P_{+}.
Figure 3.41: Demonstrating the use of slack variables, in case of linearly inseparable data Figure 3.42: Demonstrates In IR2 (left), mapped to IR3 (Right), the linearly inseparable data points allow to learn the separation hyperplane.
Figure 3.43: Demonstrating the use the classification procedure for a text sample x with a Random Forest classifier
Figure 4.1: The effect on Accuracy and Speed of ODMS based on BoF-SD of varying S D(100 to 4000).
Figure 4.2: Effects for the precise processing and speed of BoF-MD-based ODMS, varying size of individual dictionaries (SIDs) from 20 to 200.
Figure 4.3: Effect on (a) average classification of the characteristic's subset size (S ss) and sub sample quantities (Nss)
Figure 4.3: Effects on (b) Processing speed of the feature subset sizes (S ss) and number of subsamples (Nss)
Figure 4.4: ACCR vs. AB-SVM Processing BoF-SD speed for the various sub-sample numbers (NSs) and subset feature sizes (S ss). Nss = 15 and Ss = 500 is the fastest compromise.

Figure 4.5 SURF and SSVM performance of BoF-SD (C-SVC)
Figure 4.6 Performance of SURF and SSVM performance of BoF-MD (C-SVC)
Figure 4.7: Class wise ACCRli 's with 95% Confidence Intervals for BoF-SD and BoF-MD with SSVM (over the ND = 10 different 80-20 Dataset splits).
Figure 4.9: Dictionary effect (S D) for single and modular dictionaries on dictionary training times (TDT r).
Figure 4.10 Process of Random Forest Algorithm for ODMS
Figure 4.11: Demonstrates the Working the Random Forest Algorithm for ODMS
Figure 4.12: Demonstrates the Recognition rate for RF- ODMS
Figure 413: Demonstrates the Recognition Rate for ABSVMB ODMS
Figure 4-14: Demonstrates the ODMS stimulating effects (a, b) obstacle as car underneath obstruction, (c, d) incompletely obtainable of photographic camera sight, (e) little illumination, (f) undesirable replication in hood
Figure 4.15: Demonstrate the Visualization of confusion matrices
Figure 5.1: Demonstrate the Review of Data
Figure 5.2: Demonstrate the Review of Data
Figure 5.3: Demonstrate the training dataset is labeled with actual vehicles' information
Figure 5.4: Demonstrate the testing dataset also has model information but it is only used to compare the ODMS system prediction to actual model.
Figure 5.5: Demonstrates the Samples from Cars and Not cars sets
Figure 5.6: Demonstrates the Train SVM Classifier
Figure 5.7: Demonstrates the Test prediction
Figure 5.8 Demonstrates the Test prediction of car model
Figure 5.9: Demonstrates Multiscale grid for obstacle detection

Figure 5.10: Demonstrates Obstacle model output from partial view images of car Predict the models
Figure 5.11: Demonstrates Real time testing using ODMS-DCSVM
Figure 5.12 Demonstrates the minimum, maximum and average recon ignition rate for the Ford different models.
Figure 5.13 Demonstrates the minimum, maximum and average reconignition rate for the Honda different models.
Figure 5.14 Demonstrates the minimum, maximum and average reconignition rate for the Hyundai different models.
Figure 5.15 Demonstrates the minimum, maximum and average reconignition rate for the Mahindra different models.
Figure 5.16 Demonstrates the minimum, maximum and average reconignition rate for the Mits different models.
Figure 5.17 Demonstrates the Total Accuracy rate of the Hybrid Classification Techniques

Acronyms and Abbreviations

GPS	Global Positioning System
GPU	Graphics Processing Units
HOG	Histogram of Oriented Gradients
PHOG	Pyramid Histograms of Oriented Gradients
SVM	Support Vector Machine
CNN	Convolutional Neural Network
ITS	Intelligent Transportation Systems
ADAS	Advanced driver-assistance systems
AVS	Autonomous Vehicle Systems
ALPR	Automatic License Plate Recognition
ODMS	Obstacle Detection Model System
CPU	Central processing unit
KDD	Knowledge Discovery in Databases
LIDAR	Light Detection and Ranging
RADAR	Radio Detection and Ranging
RGB	Red Green Blue
HSV	Hue Saturation Value
ODMS-ABSVM	Obstacle Detection Model System Asymmetric Bagging SVM
ODMS-DCSVM	Obstacle Detection Model System Divide and Conquer Support Vector Machine
SIFT	Scale Invariant Feature Transform
BoF	Bag of Features
mACCR	mean Correct Classification Rate
SURF	Speed-Up Robust Feature
SD	Single-Dictionary
MD	Modular Dictionary
ND	Number of Training and Testing Options
ODMS-SSVM	Obstacle Detection Model System Single Multi-Class SVM Classifier
GIST	Holistic Representation of Spatial Envelope
RF-ODMS	Random Forest- Obstacle Detection Model System
CBIR	Content Based Image Retrieval

ABSTRACT

Recognition of License Plate remains a straightforward computer vision problem. The issue has been systematically discovered besides is extensively measured towards remain a well-understood problem. Recognition of License plate recognition schemes remain extensively organized entirely diagonally the globe for a variety of situations. Recognition of License plate schemes accomplish poorly popular the circumstances wherever the license plate remains ambiguous, forged, or else damaged. Likewise, Recognition of License plate schemes cannot work if only partial license plate information is available or if only vehicle's description is available hit and run, hot pursuit, and amber alert are a few of the situations where an obstacle as vehicle's description is often available. An Obstacle Detection Model System (ODMS) scheme can deliver countless value in terms of obstacle as vehicle monitoring in addition identification built on Obstacle as vehicle presence instead of the automobiles' attached license plate. A ODMS scheme besides a Recognition of License Plate scheme can remain used toward adjunct respectively other. A real-time ODMS scheme is a significant module of numerous Intelligent Transportation System (ITS) claims, by method of automatic automobile observation, road traffic supervision, motorist aid schemes, road traffic performance examination, besides road traffic monitoring, etc. The Obstacle Detection Model System (ODMS) scheme can decrease the price aimed at such applications.

Current methods of Vehicle Detection have a unique set of challenges besides problems. An few of the tests remain connected to computer vision comprises image acquisition, dissimilarities in illuminations besides lighting, dissimilarities in weather, obstruction, shades, in adding replications, etc. A limited of the challenges remain due to the nature of the problematic, such as the large variety of vehicles, the inter-class besides intra-class comparisons, adding/removal of obstacle as cars models completed time, etc. The Obstacle Detection Model System is a multi-class classification/recognition therefore, selection of machine learning procedure aimed at vigorous in addition dependable ODMS scheme remains extra challenging task. In this thesis, we present a exclusive besides vigorous real-time ODMS scheme which can knob the challenges defined above in addition identify vehicles through high precision. Towards reliably overcome the ambiguity tasks, a global

features illustration method created on the Bag-of-Features example is planned. We abstract important features since dissimilar make-model classes in an optimized dictionary, over dual dissimilar dictionary building approaches. We signify dissimilar models as of individually

class through esteem to the learned dictionary. Present dual classification schemes grounded arranged multi-class Support Vector Machines (1) Single multi-class SVM-ODMS (2) Feature Bagging-based Collective of multi-class SVMs-ODMS. These classification schemes permit concurrent knowledge of the variances amid global representations of dissimilar classes besides the likenesses between dissimilar shapes inside a similar make-model class, toward additional overcome the multiplicity challenges intended by real-time application. Extensive tests presented with Proposed approaches produce greater results aimed at imageries that remained occluded, below low illumination, limited camera opinions, or else even non-frontal opinions, available in a recently published real-world dataset.

Extract image features subsequently Obstacle vehicle images formerly produce many distinct feature directions to show the dataset. Dual existing classification procedures, Random Forest in addition Support Vector Machine, in our work We likewise proposed a DC-SVM-ODMS & dual-level Support Vector Machine classification algorithm in our work. Thus, uses of a realistic dataset to test and evaluate the proposed ODMS system. The vehicles pictures in the dataset produce real- world circumstances by way of diverse weather circumstances, dissimilar illumination exposure, obstructed images (e.g., walkers), besides dissimilar inspecting angles, etc. The proposed ODMS scheme identifies Obstacle as vehicles happening the basis of make, model although the existing ODMS systems can individually recognize the make and model. Image feature descriptors stand used to denote the car dataset. Image features can remain generally separated interested in two groups: the first kind of features is extracted using the prominent and distinctive points/patches in the image and uses the selected points/patches to represent the image. Finally Proposed Fast Multiclass Classification Method which solved Overfitting problem when Models trained on a small dataset are additional likely to see patterns that do not exist, which outcomes in high variance and very high error on a test set. we have proposed the DC-SVM-ODMS algorithm to improve the accuracy for the proposed model the obstacles. The proposed

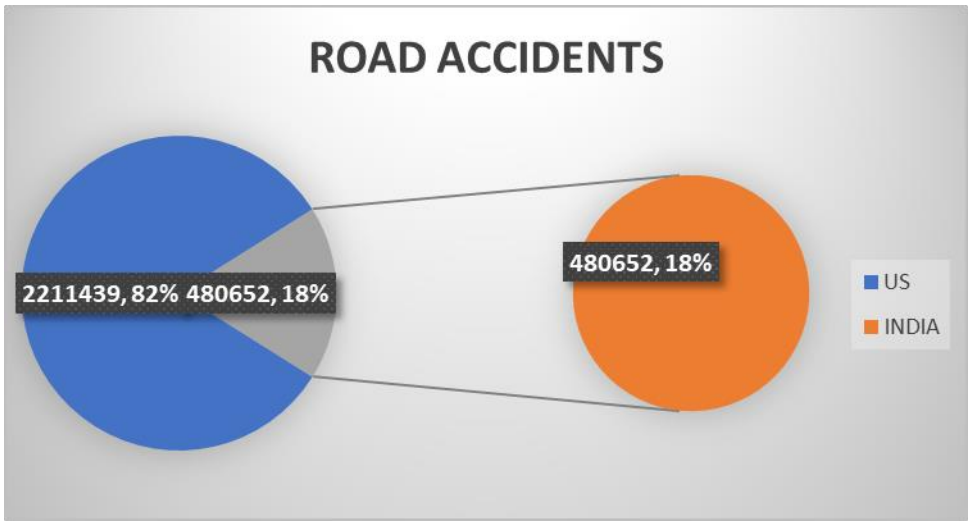
system works well in challenging situations where obstacles are partially occluded, partial out of the image frame or poorly visible due to low lighting. An Obstacle detection model system can offer great value in terms of obstacle. As obstacle as vehicle detection monitoring besides identification based on obstacle. Therefore, our Obstacle detection model system recognizes an obstacle and provides information about obstacle type information. Since the DC-SVM-ODMS provide better accuracy than remaining techniques.

Chapter 1: Introduction

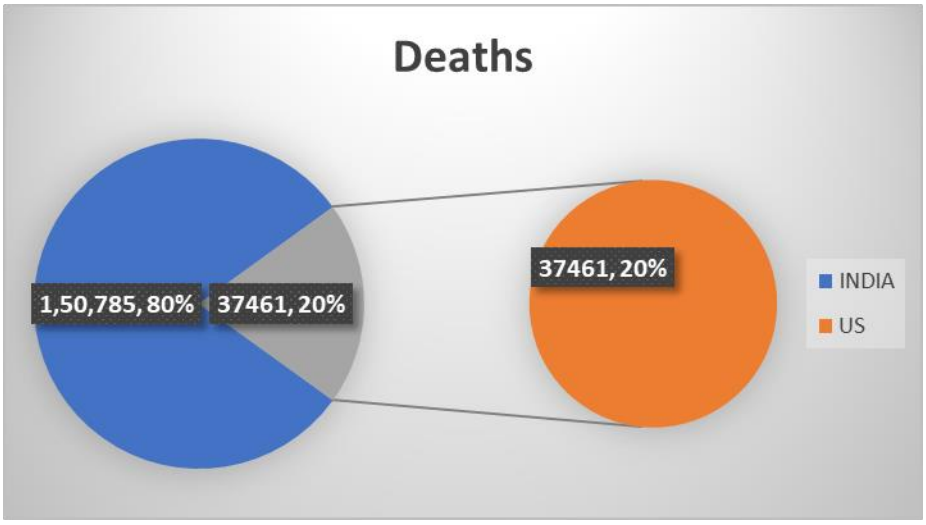
1.1 Introduction To the Study

Various autonomous car technologies have recently appeared. Several sensors in the car, such as lane maintenance, omnidirectional vehicle distance estimate, side vehicle recognition, and vehicle distance maintenance sensors, (Butakov et al 2015) have been combined to create a provision system for safe vehicle driving. This chapter describes a strategy for comprehending fully autonomous driving. The charge-coupled device (CCD) vision sensor is the most important of the sensors placed in automobiles to allow autonomous driving. The driver's visual tasks are the most important driving tasks. Visual information is used to analyse the road environment, the situation is recognised, and the vehicle steering task is finally defined by the driving task. As a result, image identification using vision sensors is important for autonomous vehicle safety support. A vehicle-type black-box device, which is a front and rear video recording device, is the most typically placed or installed sensor on vehicles among the sensors needed for safe driving assistance. The current vehicle black-box system, on the other hand, is only employed as a video recording device for accident detection. It will be feasible to promote safe driving with the black-box product equipped with a function for it. Image processing functions such as lane keeping, determining if the car in front is beginning or not, and traffic sign recognition are all included in most devices. Current black-box systems that recognise intelligent road conditions, on the other hand, can be used in situations when lighting and road conditions do not fluctuate considerably.

Figure 1.1 (a)Demonstrates the Road accidents in the United States are roughly five times higher than in India. (b)India has four times the number of deaths as the United States. [Dash, Dipak K. “This Is What Makes Indian Road Deadliest, TOI]



(a)



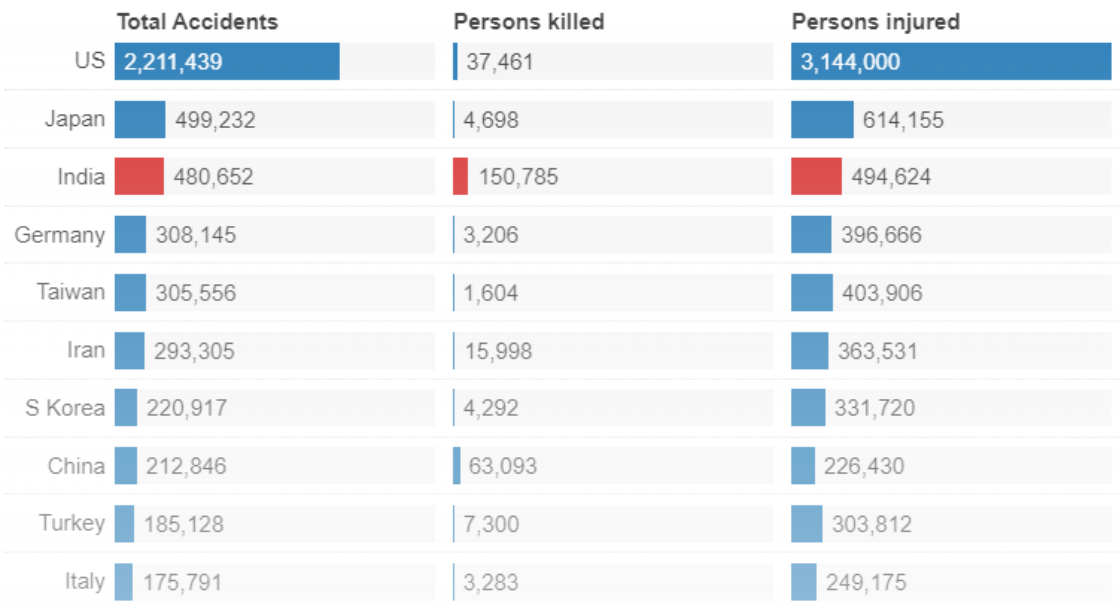
(b)

Road accidents in the United States are roughly five times higher than in India. India consumes steadily remained on the highest of list of nations through the maximum GDP. Meanwhile 2009, once World Health Organization available the opening Global Rank Account happening Highway Safety, India consumes remained by the highest of slope of nations through the maximum street fatalities. India's roadways account for one out of every eleven traffic deaths worldwide. India ranks fifth in the world in terms of automobile registrations, with about 3,000,000 added in 2019 and a street network

stretching out more than 6,000,000 kilometres, making it perhaps the most active country as far as street traffic.

Figure 1.2 Demonstrates the 2018 Word Statics Reports

[Dash, Dipak K. “This Is What Makes Indian Road Deadliest", TOI]

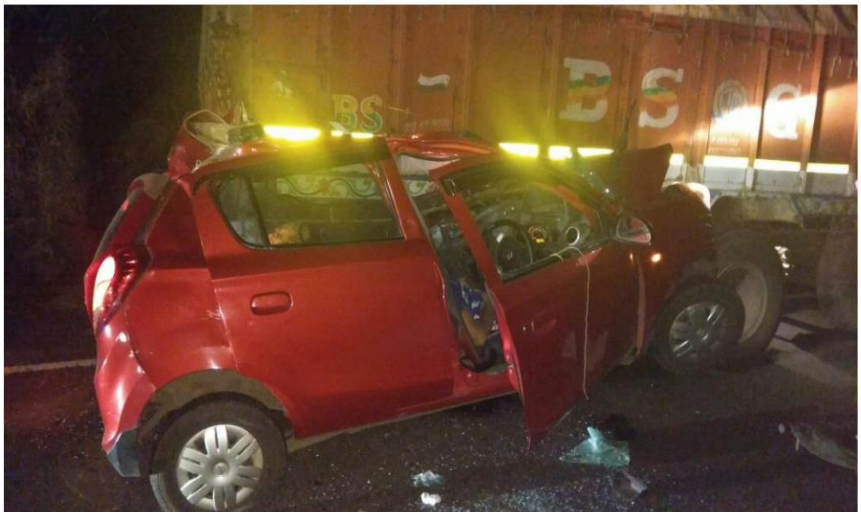


According to the recently released government road accidents in India 2019 data, India had a total of 449.002 road accidents in 2019, with 151,113 fatalities and 451,361 injuries. In rate terms, there was an abatement in each of the three, however a little one: in 2009, there were 3.86 percent less mishaps. A portion of this can be because of the new engine vehicles act, which came full circle in September 2019.It resulted in 20% fewer fatalities and 3.84 percent less injuries.

For example, in some environments, such as tunnels or bridges, it can be difficult to assess the right road status. Tunnel traffic accidents are becoming more common, and the number of fatalities is increasing. The aftermath of a tunnel traffic accident is seen in Figure 1.3. The bulk of traffic events in tunnels are caused by collisions with cars in front of them, as shown in Figure 1.3. Every year, almost 1.35 million people are killed in

automobile accidents. The global number of road traffic fatalities and injuries is expected to be cut in half by 2020, according to the 2030 Agenda for Sustainable Development. The majority of countries lose 3% of their GDP as a result of climate change to road accidents. Pedestrians, cyclists, and motorcyclists account for more than half of all road traffic deaths. Despite having about 60% of the world's vehicles, low- and middle-income countries are responsible for 93 percent of all road deaths. Children and young adults aged 5 to 29 years die most frequently as a result of road accidents. Treatment is a loss of profit for those who are murdered or disabled by their wounds, as well as relatives that need to leave on holiday, work, or school to properly focus on the afflicted. As a result of street car accidents, most countries lose 3% of their GDP.

Figure1.3. Demonstrates the aftermath of a tunnel traffic accident [Ravindra Joshi," Speeding car rams into truck near new Katraj tunnel, four dead]



Despite possessing only 1% of the world's autos, India accounts for 11% of global road deaths, the most in the world, according to a World Bank analysis. Every year, the country experiences approximately 4.5 lakh car accidents, with 1.5 lakh individual's dead. "India tops the globe in terms of road traffic fatalities and injuries. With 53 accidents every hour, each killing one person, it has 1% of the world's automobiles yet accounts for 11% of all road incident deaths. According to the report, during the last decade, 13 lakh

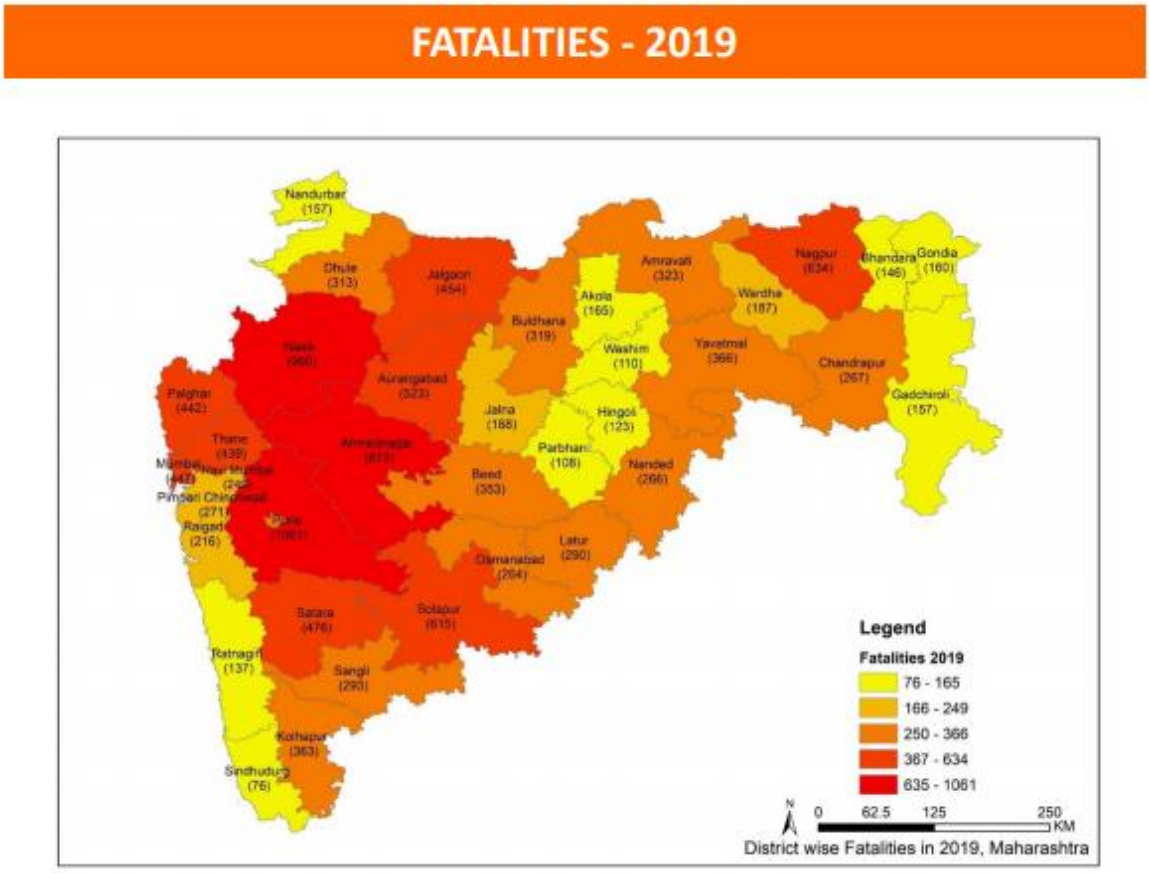
people have died and another 50 lakhs have been injured on Indian roadways. According to statistics, 70 percent of road accidents occur among India's youthful, productive population, ages 18 to 45.

Figure1.4. Demonstrates the Maharashtra State Highway Police statistics on death accidents, critically injured accidents, and minor injury accidents are displayed. [Roads Accidents in Maharashtra ,2019]

SR. NO.	YEAR	FATAL ACCIDENTS		SERIOUSLY INJURED ACCIDENTS		MINOR INJURY ACCIDENTS		WITHOUT INJURY ACCIDENTS	TOTAL ACCIDENTS
		NO. OF ACCIDENT	TOTAL KILLED	NO. OF ACCIDENT	TOTAL SERIOUSLY INJURED	NO. OF ACCIDENT	TOTAL MINOR INJURED		
1	2016	11780	12935	13273	22273	8554	13611	6271	39878
2	2017	11454	12511	12333	20767	7098	11477	5171	36056
3	2018	12098	13261	12648	20335	6585	11030	4386	35717
4	2019	11787	12788	12197	19152	5473	9476	3568	32925

Figure 1.4 shows the accident status from 2016 to 2019 according to the Maharashtra highway police's traffic accident analysis system. The number of fatal accidents climbs each year as the number of vehicles on the road increases, resulting in an increase in the accident rate. As the number of vehicles on the road grows year after year, so does the number of persons critically injure in fatal incidents. In 2016, the total number of badly injured people in fatal accidents increased to 22273. There were 8554 minor injured accidents in 2016, out of a total of 13611 minor injured accidents.

Figure1.5. Demonstrate the Fatalities-2019 data from Maharashtra's several districts [Roads Accidents in Maharashtra ,2019]



The Institute for Transportation Studies (ITS) defines ITS as “the application of advanced and developing technologies (computers, sensors, control, communications, and electronic devices) in transportation to save lives, time, money, energy, and the environment.” Vehicles, infrastructure, drivers, and other users interact dynamically in an ITS to achieve these aims. To improve transportation safety and security, governments, business, and academics are working together to develop and implement advanced ITS. Numerous significant ITS applications and segments, like ADAS, AVS, road traffic besides movement observing, traffic conduct investigation, traffic management, and so on, have benefited from the advancement of digital image sensors and computer vision techniques. Due to increased security concerns in ITS, vehicle identification and

categorization are of major relevance in these applications. The great variety in appearance between items of the same class, as well as the additional variability between instances of the same object due to variances in viewing conditions, make object detection difficult. An object detection system must be able to detect the presence or absence of an object, such as an automobile, in a variety of lighting conditions, scales, poses, and background clutter levels. Image categorization is a key topic in PC vision that fills in as a fundamental way to deal with organize pictures in a solo way. The situation contains measuring whether *a* twin comprises a particular type of object. The key obstacles in fundamental recognition tasks have been the considerable intra-class variability and variations in the global look of items within the same category. A category model should, in theory, be able to describe all of the items in the category while still being flexible enough to account for intra-class heterogeneity. Subordinate-level categories, such as identification of different kinds of birds, flowers, leaves, dog breeds, and aircraft models, have recently been a popular research topic in picture classification, owing to real-world applications. These issues are characterised as fine-grained visual classification tasks that aim to classify same-category items using just subtitle distinctions. The level of difference between distinct categories is the principal difference between fundamental level and powdered graphic description. The examination of knowing a bird from a plane is straightforward in light of the fact that there are a few visual markers. This isn't the situation, notwithstanding, when we try to recognize two sorts of birds with minor differences in appearance, such as differing beak colours. Methods that are more discriminatory than those utilised for general picture classification are required for the latter task.

In fine-grained classification, there are two fundamental obstacles to overcome. To begin, distinguishing minor differences in appearance that would establish the intended fine-grained classes. This requires an appearance include portrayal that holds basic segregation subtleties while disposing of unimportant information. Finding and finding the parts that contain discriminator subtleties is another problem. A reliable classifier would accentuate the distinctions in the representation of distinct categories while reducing the similar features in such applications. Despite these obstacles, fine-grained jobs can be extremely valuable if handled correctly. Because of its potential to make

computers compete with human professionals in visual comprehension in some verifiable applications, it is often regarded as one of the cornerstones in computer vision. Recent advancements in computer-based applications have resulted in the invasion of programming and logging devices into nearly every aspect of human activity. The advancements in programming have been fueled by ongoing advancements in fascinating registering fields such as:

- Advances in calculating power based on Moors' law (Moore et al 1965). Because of advancements in calculating power, small spectacular observation cameras with better picture quality and goal, as well as simple access to enormous capacity, are now available.
- Advances in test probe devices, as well as their availability and accessibility. Basic detecting gadgets, reconnaissance cameras, LIDAR/RADAR, movement and Global Positioning System (GPS) detecting, and other detecting and examining gadgets are examples of test gadgets (Dufour et al 2013).
- Advances in network technology, as well as advancements in programming phases, have prompted the growth of distributed computing innovation.
- Advances in computer vision have modified the old-style method to dealing with difficult-to-resolve difficulties. Advanced PC vision technologies have made it possible to treat massive datasets with more complicated computations in ways that were before unthinkable.
- The ImageNet was mentioned in special (Deng et al 2009) project, which created a picture dataset and enabled the WordNet movement, as well as the achievements made by the CIFAR research Centre.
- Advances in scaling and normalizing image segments, as well as increased use of Graphics Processing Units (GPU) for efficient processing of picture stumbling blocks and advanced separating, indicate significant improvements.
- Artificial Intelligence technologies, notably Machine Learning, have advanced. CPUs, memory, and capacity in multi-centers are more grounded. Advances in equal preparation and cloud stages increased the examination local area's overall capability to comprehend the advantages of current AI calculations.

1.2 Obstacle Detection Mechanism

1.2.1 Appearance-based Methods for Obstacle Detection

Obstacle as Vehicle appearances vary in size, form, and shading. In the case of Obstacle as Vehicle identification, appearance-based algorithms make use of previous data to divide the closer view (which includes the revenue objects) and the foundation (which includes the entire vehicle) (Barnich et al 2011). Because cars are rectangular, different features, such as evenness, shade, edge (even/vertical), and shadow, serve as important triggers for vehicle finding. Each element has its own set of advantages and disadvantages, and they can be used separately or in combination. Descriptors have been presented in light of these highlights, and they have proven to be effective in vehicle finding. (Sun et al 2006, Malley et al 2008), Obstacle as Vehicle finding tactics that are focused on appearance usually track two fundamental advances1. Hypothesis Generation anywhere the places of expected vehicles in an image are speculated, and 2 Hypothesis Verification somewhere examinations remain achieved toward confirm the incidence of cars popular image. Theory Generation Subordinate Features Now we review specialist low-level features e.g., concealing, shadow, equilibrium besides edge, aimed at the usage of appearance-based HG. These highlights are clear, powerful, and of exceptional worth in vehicle-related information extraction. a) Color: Color data gives rich data in video pictures, and has been used to distinguish visual highlights of vehicles, example, car illuminations (chen et al 2012), Traffic investigation photographic camera often effort with the RGB perfect, though the 3 frequencies remain deeply related besides the separate value of R, G, B trusts upon illumination powerfully (Yang et al 2011). To take care of this issue, change from from RGB color space to HSV (Malley et al 2010, Malley et al 2008), YCrCb (Wang et al 2015, Peng et al 2012), L*a*b (Cabani et al 2005) remains functioned towards feature the sore besides snowy blending color besides reduce the influence of improving alteration. Popular (Tsai et al 2007), new-fangled color change picture-perfect continued thoughtful to discriminate obstacle as car color subsequently static pictures. Table 1 demos in what way color information is castoff in car lights proof of identity.

Table 1.1. Proves in what manner color data remains used popular Obstacle as car illuminations proof of identity.

Color Space	Corresponding component in RGB color space	Application	Reference
RGB	Red	Tail-light detection at night	(P.Thammakaroon et al 2009)
HSV	Red	Tail-light detection at night	(R.O'Malley et al 2008)
HSV	Red and White	Tail-light detection at night	(R.O'Malley et al 2008)
YCrCb	Red	Tail-light detection at night	(D.Y. Chen et al 2012)
YCrCb	Red	Brake-light detection at daytime	(X. Wang et al 2015)
L*a*b	Red	Tail-light detection at daytime/adverse weather conditions	(E SKodras et al 2012)
L*a*b	Red	Brake light detection at daytime	(H.T Chen et al 2016)

a) Color: Color information in video images gives a wealth of information, and it has been used to detect visual elements of automobiles such as licence plates. License plates (V. Abolghasemi et al 2009), car lights .Surveillance cameras in the streets. The RGB paradigm is commonly used, however the three channels are extremely important. R, G, and B are all interrelated, and their separate values are affected by brightness strongly. Conversion from RGB color to CMYK color is used to tackle this problem. To highlight the red and white color information and lessen the effect, space to HSV (R. O'Malley et al 2008), YCbCr (D.Y. Chen et al 2012) (X. Wang et al 2015) Lab (D.Y. Chen et al 2012) is used.

b) Shadow: One of the common challenges in vehicle identification proof is seeing a shadow that indicates the vehicle's tasks and progress in the scene. As one type of nearby light changes, shadows can produce complications such as uniting, shape morphing, and item inadequacy (Prati et al 2015). Shadow assessment is a working examination region in various tests since it perceives hidden setup from moving objects (Tzomakas et al 1998). Overall, a covering prototypical built scheduled difference (Asaidi et al 2014), excellence (Horprasert et al 1999), nasty, besides

indecision of altogether shading portions can be used to recognize and eliminate vehicle shadow regions.

c) Symmetry: Symmetry is a term that refers to something's symmetry. From a frontal vantage point. It is frequently used to detect the vehicle location in photos through building a symmetrical prototypical (Collado et al 2004), detection regularity machetes (W. Liu et al 2007), or else centre opinions (Teoh et al 2012). Current video cameras, on the other hand, are deployed at varied angles and viewpoints, making symmetry a less-used trait in modern vehicle identification technologies.

d) Edge has been widely used in vehicle detection as a major source of contour information. Edge detection can be done in real time because of its cheap processing complexity. Traditional edge-based vehicle detection systems build horizontal and vertical edge maps using operators like Sobel, Canny, and Prewitt. For vehicle detection, Edge is frequently paired with other features. Multiscale edge fusion (K. Mu et al 2006) was utilized to locate target cars in a state-of-the-art vehicle detection technique. Edge is frequently used to offer contour information, while other features are used to extract vehicle candidates by supplying contextual information. Low-level feature extraction remains rapid besides informal, nevertheless the important difficulty remains that 1 feature not proficiently characterize altogether significant info. Single technique remains toward usage frequent characteristics towards perceive car portions, such by way of car numeral plates (C. Patel et al 2013) and vehicle lights, to tackle this problem (Skodras et al 2012, D.Y. Chen et al 2011) Based on attributes like color, edge, and blob detectors, a ROI (Region of Interest) is created. Vehicle candidates can be successfully located based on the visual parts. Another option is to combine features in video images to extract the entire vehicle(s). Contextual and contour information about cars can be extracted more effectively using a combination of two or more attributes.

1.2.1.1 Descriptors of Local Features

There has been a shift in recent studies from basic low-level structures towards

strong local feature forms. Feature signifiers remain considered aimed at a exact application purpose and enable for a quick and effective search of objects in photos. Descriptors should, in theory, be able to cope with a variety of objects and be resistant to changes in backdrop, as well as being invariant to geometric and photometric transformations (J. Li et al 2011). In the literature, a number of appearance features have been presented, of which 2 signifiers, HOG besides Harr-like structures, remain attractive additional joint in current car credit.

a) HOG: The Histogram of Gradients (HOG) is an object identification feature descriptor. HOG was first evaluated in the field of pedestrian detection and subsequently expanded to the field of moving vehicle recognition by capturing edge of gradient structure that is particularly typical of local shape. To extract vehicle features, (G. Yan et al 2003) used a mixture of two HOG vectors. Pyramid Histograms of Oriented Gradients (PHOG) characteristics were recovered from traffic photos as fundamental features in (N. Khairdoost et al 2003). Three distinct configurations linking flat (H-HOG), perpendicular (V-HOG), besides concentrical four-sided (CR-HOG) signifiers are described in (J. Arróspide et al 2006). were presented as a more cost-effective alternative to the original HOG. For vehicle verification, symmetric HOG vectors were produced by (M. Cheon et al 2003).

b) Haar-like Features: Viola and Jones presented Haar-like features (P. Viola et al 2001) to detect human faces. Edge, line, and center-surround characteristics are all common Haar characteristics. It has been utilized in several vehicle detection algorithms for feature representation since it is simple and easy to compute. Harr-like mining of feature technique remained applied towards represent a vehicle's boundaries besides buildings popular (X. Wen et al 2015). A 2D triangle filter based on Haar-like features was presented to detect cars in (A. Haselhoff et al 2009). Sivaraman (S. Sivaraman et al 2010) proposed a broad framework for learning based on Haar-like features. Haar-like features can be computed quickly. The dimension of this feature vector derived from photographs, on the other hand, is quite large (H. Wang et al 2014) The hardware storage was then reduced via a dimension reduction procedure.

c) Other Feature Descriptors: Despite the efficiency of HOG and Haar-like features in collecting contextual information at a regional level, certain other appearance features also demonstrate exceptional usefulness. To extract taillights, Gabor features (Q. Ming et al 2011) were employed. Gabor characteristics were utilized to detect automobiles based on rear view in (Z. Sun et al 2006). Accelerate The wheel on the back part of the car was extracted using Robust Features (SURF) (H. Bayet al 2006). (B.F. Lin et al 2011). Scale-Invariant Feature Transform (SIFT) features (D.G. Lowe et al 2015) were utilized to extract interest spots in a picture, which aids in the discovery of a vehicle's bounding box (X. Zhang et al 2011).

1.2.1.2 Hypothesis Verification (HV)

Following the generation of potential candidates from image sequences, the following stage remains towards examination the hypothesis' accuracy. HV remains measured by way of a 2-class design organization problematic by means of machine learning approaches: vehicle versus non-vehicle (X. Wen et al 2015) Positive photos (vehicles class) and negative photos (non-vehicles class) are collected from various sources, then classifiers are trained on them. Each image in the training samples is represented by one or more feature vectors. Classifiers use statistics to understand the features of car presence besides shape a decision link amongst the car besides non-car modules. Towards get finest appearance-based organization presentation, within class changeability necessity remain mined. SVM, Adaboost, besides Neural Networks remain 3 samples of car discovery classifiers. SVM remains a judicial classifier that studies the choice border amongst 2 modules by constructing a hyperplane. Adaboost (Y. Freund et al 1995) is a generative weak classifier that combines local feature descriptors to increase the performance of a simple classifier (N. Buch et al 2011). Neural Networks (NN) are a type of artificial intelligence (C.M. Bishop et al 1995), have been increasingly popular in recent years, and they can learn extremely nonlinear decision boundaries (Z. Sun et al 2006). Neural Networks, on the other hand, suffer from the time-consuming computation of parameter adjustment. CNN consume remained effectively working popular vision-based car discovery meanwhile the appearance of bottomless knowledge skill. For

car discovery, Profligate R-CNN remained utilized, by means of a area suggestion net importantly dropping the suggestion value. A shallow fully convolutional network termed the fast vehicle proposal network (FVPN) was suggested in (Y. Zhou et al 2015) towards localize altogether vehicle-like substances trendy actual.

Approximately sure feature-classifier couples consume newly become popular in vehicle detection under a variety of situations. The most commonly utilized couples are HOG features and SVM classifiers (J. Lan et al 2014) and Haar-like features and Adaboost classifiers (J. Lan et al 2014). Comparative studies of features, classifiers, and feature-classifier combinations have been conducted using existing techniques. In (P. Negri et al 2008), HOG and Haar-like properties were compared and used to for rear-view vehicle detection, build a cascade of boosted classifiers. SVM and NNs were contrasted as HV methods in both basic and complicated scenarios in (Z. Sun et al 2006). As an active learning framework, the HOG-SVM and Haar-Adaboost combinations were investigated in (S. Sivaraman et al 2014).

Table 1.2 Demonstrates a selection of the portion in mining-classifier mixtures

Features	Classifiers	Field of View
Gabor	Back propagation neural network (Qing et al 2011)	Rear view
	SVM (Z. Sun et al 2006)	Rear view
Haar-like	Artificial neural network (A. Mohamed et al 2015)	Side view
	Adaboost (X. Wen et al 2015)	Rear view
HoG	Adaboost (G. Yan et al 2016)	Rear view
	SVM (M. Cheon et al 2012)	Rear view
PHOG - PCA	GA – SVM (Khairdoost et al 2013)	Front/Rear view

In terms of orientation information, all features show pixels. Gabor structures proposal scale besides direction info, HoG structures compute gradient magnitude besides alignment toward figure histogram, and Haar-like features extract orientation information using rectangular filters. Simultaneously, we notice that the majority of combinations detect and classify cars in the rear view, with only a small part of detection occurring in the side view. Edge features have been used in several research to extract symmetric properties of vehicle looks, and symmetry can be highly useful in front/back perspective. In rear-view photos, both HOG-SVM and Harr-Adaboost combos perform well. Haar-like features, when paired with generative Adaboost classifiers, can provide a speedy detection performance since they are sensitive to edge and symmetric structures. Unlike the Haar-Adaboost combos, the HOG features are primarily concerned with extracting edge orientations. As a result, the field of view can be expanded beyond the usual front/rear view to include many views of the vehicle. HOG characteristics were used to identify four orientations (front, left, right, and distant) in (G. Yan et al 2016). HOG characteristics were chosen and taught to distinguish different vehicle orientations (P.E. Rybski, et al 2011)

1.2.2 Methods based on motion

There are primarily two types of information for vehicle detection and tracking: visual features and motion information (J.M. Collado et al 2004) Visual elements such as the vehicle's color and texture are recognised using video frame pixels, making this a useful way for describing and classifying the vehicle's look. Motion information, on the other hand, is kept in motion vector structures and is detected using a sequence of frames. Due to the merging of offered information, combining these two sources can be considerably more valuable. Vehicle identification and tracking techniques are classified as either monocular (single vision) or binocular (stereo vision) tactics (W. Liu et al 2007) Lane change assist with rear vehicle identification and tracking To recognize and track cars, monocular detection methods use a single camera, but binocular detection requires two cameras. In stereo vision vehicle detection, motion-based methods are more frequent than appearance-based methods (S.S. Teoh et al 2012). Vehicle identification systems for road surveillance

typically trust happening edge information heuristics, besides to approximately portion, they're constructed using machine learning methods similar Haar-like structures besides rise forms to create automatically learned appearances (A.L. Fytsilis et al 2016). However, the many methodologies for detecting moving vehicles in camera outputs can be grouped into the following classes as a broad classification.

1.2.2.1 Background Subtraction: Background Subtraction is a common technique for separating moving objects from static backgrounds. Through deducting the present frame of the video since a position still circumstantial, moving zones containing vehicle edges are recognised. Background modelling is the process of creating a reference background. The background model must be refreshed on a regular basis and must not contain any moving cars. A Combination of Gaussians method container remain castoff by way of an on-line estimate gage towards update the model. Moving components are defined as pixels in a frame that differ by more than a predetermined threshold. This approach is extremely sensitive to environmental changes.

1.2.2.2 Points/Corner Detection: In video frames, it relates towards the discovery of attention opinions or else car turnings. A angle remains the place wherever 2 boundaries of a car through separate instructions encounter hip a resident neighborhood of opinion. An attention opinion remains a distinct, effortlessly obvious advertisement trendy a audiovisual frame. The Harris & Stephens, Plessey, and Shi–Tomasi algorithms are only a few examples of point and corner detection algorithms.

1.2.2.3 Statistical Approach: By utilizing statistical properties of the pixels, the Statistical Approach is aimed to overcome the shortcomings of the background subtraction method. Cross-Correlation is a popular approach (CC). Foreground pixels that contain moving vehicles are recognised using the background subtraction approach by comparing each frame's pixels statistics to the background model's reference pixel. Statistical methods are well-known for their reliability in overcoming the effects of lighting fluctuations, noise, and shadows.

1.2.2.4 Supervised Learning: Instead, then keeping a collection of templates, the Supervised Learning approach recognizes cars by learning diverse viewpoints of them. These approaches employ a collection of training data to create a mapping function, or classification, that contains the expected vehicle form. For the detection process, numerous dissimilar knowledge algorithms such by way of Neural Networks ,Adaptive Boosting, decision trees, besides SVM consume remained established.

1.2.2.5 Temporal (Frame) Differencing By computing the pixel-by-pixel difference between subsequent frames in a video clip, the Temporal (Frame) Differencing methodology finds moving areas. When detecting moving vehicles, the number of frames employed is usually two or three. Moving vehicle detection remains recognized by way of a feasible technique hip bags someplace the photographic camera remains touching because it does not rely on a static reference. The situation must remain well-known that hip approximately schemes, the camera sign would remain assessed beforehand the car sign. Once a car changes gradually in a scenario, this strategy does not perform effectively.

1.2.2.6 Optical Flow method: The Optical Flow approach detects stirring automobiles hip a act through calculation the deceptive way besides speed of each pixel hip the edge contextual gesture perfect, which remains a laborious procedure. This technique tin cutting a background motion model as well as isolated motion areas, making optical flow a powerful model for detecting motion in a difficult video sequence, even if the camera or background is moving.

1.3. Obstacle as Vehicle Detection Challenges

Current video-based systems are sensitive to operational environment factors such as weather and lighting, resulting in lower accuracy and reliability. The primary issues of vehicle speed measurement applications are described in this section (P. Barcellos et al 2014) (I. Cabani and colleagues, 2005) (D.Y. Chen and colleagues, 2005)

1.3.1 Camera Motions: When there is motion in the field of view of the camera, such as in recordings acquired through unbalanced before vibrant photographic camera, dispensation the video container remain problematic. The result of this marvels remains gesture distortion in a video scene, which makes both detecting and tracking steps additional problematic. Through time-based de-blurring before

forecasting a single motion blur kernel for the entire image, motion blur can be avoided. Environmental factors such as winds are to blame for this phenomenon.

1.3.2. Camouflage

Creation precise classification besides segmentation remains problematic owing toward the resemblance of approximately cars' arrivals toward the still contextual In tracking applications, this is extremely significant. For temporal differencing algorithms, camouflage is a particularly difficult problem. Mathematical morphological techniques can help to solve this problem.

1.3.3. Clutter

The difficulty of distinguishing cars from the backdrop scene is complicated by the presence of clutter in the background. In backdrop subtraction and texture-based detection approaches, this type of issue is very essential. Segmentation is a common technique for dealing with clutter popular a audiovisual frame, which involves assembling limited picture info besides implanting it with an active contour model based on region.

1.3.4. Jammed Road traffic Circumstances

Car obstruction besides road traffic overcrowding remain 2 predominant occurrences in urban traffic photos. When a car passes behind other vehicles in front of a camera, this is known as vehicle occlusion. Occlusion, both partial and complete, can alter the background frame computation process. When a vehicle detection system is reliant on motion data, crowded traffic circumstances might have a significant impact on the outcomes.

1.3.5. Dissimilar Forms of Automobiles

Since around remain numerous car kinds, uncertainty the discovery scheme is dependent on visual elements such as headlights and bumpers, certain vehicles may be incorrectly classified. This is the most typical supervised approach vehicle detecting problem.

1.3.6. Dynamic Changes

A moving object, such as a parked vehicle moving out of a parking lot, reveals the background in this case. To put it another way, while some sections of the scene may have movement, they should be considered background. Since the greatest new edges

remain likewise comprised trendy the calculation of stirring areas, background subtraction approaches are frequently sensitive to dynamic changes, but temporal differencing methods are extremely adaptive to them.

1.3.7. Low Light Conditions

Cars cannot be spotted by their visual features at night or in low light settings such as tunnels, and the only component of them that can be recognised is the headlight/taillight bulbs. As a result, other portions that do not create light or are exposed to light from a near distance appear dark. Due to the lack of visual features, identification and tracking processes may be hampered. Another difficulty is to link detected lighting so that each car can be considered separately. Such scenarios can be handled by image binary conversion with an appropriate threshold value. Vehicle headlights/taillights, as well as poor illumination that can produce a lot of noise, can pose a lot of problems for night videos for the detecting task, there are too many obstacles.

1.3.8. Multiscale Problem

A multiscale problem is a condition in which the resolution, size, or viewing angle of a vehicle differs due to the perspective effect. In a video frame, the viewpoint result controls different automobile–photographic camera detachments. Once a car is in near immediacy toward the camera, the situation is thought tall scale besides consumes a massive image scope, tall quality, besides a extensive viewing viewpoint. Once a automobile remains far-off missing since the camera, schedule ding the additional pointer, the situation is low-slung climbed, through a minor image scope, low-slung excellence, besides a unimportant angle since the photographic camera. As a result, vehicles that are far away from the camera have a blurred effect and unidentifiable details such as the licence plate and headlights. Due to the time-varying nature of vehicle attributes, the multiscale problem makes vehicle detection difficult. As a result, removing the inherent perspective effect from acquired frames is an important task for video systems, which remains typically proficient through choosing an suitable ROI, congregation extra spatial-temporal info through extra sensors, before analysing chronological data of a order of video edges.

1.3.9. Noise

In most cases, the audiovisual torrent remains covered through sound. Contextual deduction methods aimed at audiovisual shadowing necessity contract through signs that consume remained spoiled through numerous foundations of noise, such by way of device noise before density objects. Together discovery besides following responsibilities might be hampered by noise.

1.3.10. Obstacle Motion Speed

The detection task is greatly influenced by the speed of moving vehicles. The time-based differencing technique determination flop towards notice the ranges of the entity keep a unchanging area unknown the article changes same gradually. A fast-moving vehicle, happening the additional indicator, shrubberies a track of impression area trendy the forefront part. Obsessing irregularities popular the noticed signal are caused by intermittent movement of objects.

1.3.11. Stereo Matching

This is primarily a problematic popular stereo-vision car discovery. The problematic tin remains well-defined by way of discovery the finest only opinion post amid 2 distinct audiovisual frames besides identical them by means of a inequality chart as quickly as feasible.

1.3.12. Sudden Changes in Illumination

During the day, the intensity of light changes, which affects the appearance of the backdrop scene and can lead to false positive vehicle detections, which implies that non-vehicle objects are mistaken for automobiles an audiovisual arrangement. A quick change since gray towards bright sunshine remains a even occurrence that can cause abrupt lighting changes.

1.3.13 Obstacle as Vehicle Orientation

Some detection methods rely on a single perspective of cars, but a video clip contains several orientations. Some frequent approaches include cameras that look onward, scheduled the side-view glass, privileged a affecting car, happening highest of car besides looking available the cross-traveler's opening. The vehicle stance is influenced by a variety of parameters such as camera positioning and angle, resulting in a wide range of vehicle appearances. This problem is particularly critical in vehicle recognition based on appearance.

1.3.14 Obstacle as Vehicle Shadows

Cast shadows always accompany moving cars in sunny weather. These shades container be measured a portion of the car, besides they might take an influence happening the detection procedure. Forming tracker removal before retaining advantage discovery to distinct sleuth since automobiles remain 2 typical ways that can improve vehicle detection accuracy in such situations. Following background reduction, shadows cast by foreground objects frequently hinder future processing processes.

TABLE 1.3 Demonstrates a review of the vehicle as far as obstacle detection and tracking are concerned.

Challenge	Description	Detection Field Affected	Tracking Field Affected
Camera Motions	The camera's field of vision is shaky.	Background Subtraction Statistical Approach	LK/KLT Condensation
Camouflage	Some vehicles' appearances are similar to the static background.	Background Subtraction Points/Corner Detection Statistical Approach Supervised Learning Temporal Differencing	Boundary/Edge-based The Kalman Filter TLD
Clutter	There is congestion in the background scene.	Background Subtraction Statistical Approach Supervised Learning	Mean-shift LK/KLT Condensation
Congested Traffic Conditions	In the foreground, there is a vehicle obstruction and a traffic jam.	Background Subtraction Points/Corner Detection Statistical Approach Supervised Learning Temporal Differencing	Boundary/Edge-based The Kalman Filter TLD
Different Shapes of	Dependence of the detecting step on	Supervised Learning	Boundary/Edge-based

Vehicles	vehicle visual characteristics.		Condensation
Dynamic Changes	A moving object reveals the background.	Background Subtraction Points/Corner Detection	The Kalman Filter
Low Light Conditions	Videos taken at night and in other low-light situations	Background Subtraction Supervised Learning	Condensation
Multiscale Problem	Because of the perspective effect, there is a disparity in vehicle sizes or viewing angles.	Points/Corner Detection Supervised Learning Temporal Differencing Optical Flow	Boundary/Edge-based Mean-shift The Kalman Filter TLD
Noise	Various types of video sounds	Background Subtraction Points/Corner Detection Supervised Learning	Boundary/Edge-based Mean-shift TLD
Object Motion Speed	Obstacle as Vehicles that are either slow or too quick in a video scene	Temporal Differencing Optical Flow	LK/KLT
Stereo Matching	The best search for a single point correspondence between two video frames feasible.	Temporal Differencing Optical Flow	-
Sudden Changes in Illumination	The impact of changing light intensity	Background Subtraction Temporal	Condensation

Obstacle as Vehicle Orientation	Various perspectives on the obstacle as a vehicle	Points/Corner Detection Supervised Learning Optical Flow	Boundary/ Edge-based TLD
Obstacle as Vehicle Shadows	Scene lighting and illumination	Background Subtraction Points/Corner Detection	Boundary/Edge-based

1.4 Fully Automated Data Processing

The demand for Fully Automated Data Processing was spurred by the rapid economic growth of computing technologies. The invention, implementation, and optimization of data processing is referred to as automated data processing. Fully Automated Data Processing systems efficiently gather, store, and analyses huge volumes of data with minimal human input, and then send the retrieved data to the appropriate audience. Advanced Data Processing Systems Provide Benefits in The Following Areas:

- Speed: Speed for example, it took 18 months to analyses the social media photographs from the 2011 Vancouver unrest. Automatic picture data processing should only take a fraction of this time. Productivity: Humans are subject to fatigue, confusion, bias, loss of focus, and other influencing elements leading to lower productivity.
- Reliability and Accessibility: An automated system can work nonstop for 24 hours a day, seven days a week. It also ensures that duties are successfully and consistently done.
- Cost: Normally, the major motivation for automation is not cost. Automation saves operational costs by enhancing productivity, dependability, and speed, despite the expensive initial development and setup costs.

Computer vision is useful for Automated Data Processing because it allows businesses to automatically recognize and process the contents of photos and videos.

1.5 Intelligent Transportation Systems

Intelligent Transportation Systems (ITS) have gotten closer to reality in recent years thanks to a slew of innovative technology and solutions. Numerous significant ITS applications besides mechanisms, such by way of Advanced ADAS, AVS, road traffic besides movement 24-hour care, road traffic behaviour examination, road traffic organization, smart parking, besides self-guided automobile schemes, benefit from the advancement of digital image sensors and computer vision techniques. Other deployments can be found in both the public and private sectors. Outstanding to amplified safety worries in ITS then difficult subdivisions such by way of battered ad or one-to-one care for corruption stoppage besides care, identification and categorization of vehicles is of major importance in these applications. A 2014, around remained 907 squillion nearside automobiles besides 329 million viable automobiles proceeding the highway. Trendy contrast, nearby were 678 million traveller cars besides 248 million profitable automobiles listed international trendy 2006, according to statistics. This is a 33.7 percent increase in passenger vehicles and a 32.6 percent increase in commercial vehicles. If current trends continue, there will be around 1.7 billion registered automobiles on the road worldwide in 2035. This necessitates the implementation of a system that can effectively and reliably identify cars and be used in a variety of scenarios such as incident detection, behaviour analysis, and comprehension. AVS can be used to identify the difficulties of vehicle detection, identification, classification, and tracking in general. We may obtain additional meta information, such as vehicle make, model, logo, production year, maximum speed, acceleration, and so on, by using fine-grained classification of cars in conveyance and public security. Significant research has been done throughout the years to address these problems. Though, categorizing cars hooked on satisfactory groups such by way of brands besides reproductions takes individual recently increased purchase, besides nearby stand motionless many obstacles towards speechless. The part of this thesis is to use state-of-the-art computer vision-based techniques

to develop unique approaches to handle the issues in automated obstacle Detection Model System (ODMS).

1.6 Automated Obstacle as vehicle Monitoring

By way an outcome of improved safety consciousness aimed at access control schemes popular car parks lots, constructions, besides limited locations, the necessity for vehicle identification and categorization has become more common in recent years. In such high-risk situations, an AVS system based on surveillance video footage can substantially aid security officers in categorizing vehicles based on their look and tracking them throughout a multi-camera network. Furthermore, act enforcement organisations characteristically need glance at besides 24-hour care a distrustful automobile as of lots of road traffic photographs grounded on general imageries of car attributes since an object popular post-event investigation of automobile-related crimes. Using ODMS technology in this situation would save a significant amount of time, resources, and manpower, allowing the crime scene to be cleared faster. Besides, for submissions such as electrical clang assembly, vision grounded AVS schemes might remain cast-off by way of an addition to present schemes to put on dissimilar charges to dissimilar kinds of cars in a cost-effective and automated manner. In an intelligent transportation system, statistics of vehicle flow, associated with vehicle models, are more useful in traffic control or traffic monitoring. During high traffic hours, the travel time between observed points can be approximated to offer a detailed traffic status. Traditional vehicle identification systems rely on manual human observations or automatic licence plate recognition (ALPR) technology to identify vehicle makes and models. Both approaches are prone to failure and have a number of drawbacks. Human observers have a tough time remembering and distinguishing between the many different car manufacturers and replicas. Happening the additional pointer AVS schemes that trust happening licence plates have a number of drawbacks. For starters, most surveillance cameras aren't set up to record licence plates, hence plate recognition performance on images/video data recorded by these cameras suffers significantly. Furthermore, due to varying lighting circumstances, licence plates can easily fabricate, damaged, changed, obscured, or undetectable. Furthermore, in some regions, having the licence plate on the front or back of the vehicle may not be required. As a result, if the ALPR system isn't set up to look for licence plates

from both sides of the car, it may fail. This could result in the registry getting incorrect information about the vehicle's make or model. To discourse the overhead inadequacies trendy old-style car identification besides classification schemes, the ODMS scheme's acknowledgement of the brand besides perfect of the car can addition licence plate appreciation schemes through as long as an advanced level of robustness in contradiction of deceitful usage of licence plates before unfortunate image value, thereby increasing security.

1.7 Reasons for Using Computer Vision for Obstacle as Vehicle Detection

Computer vision is an interdisciplinary field that studies how computers can be built to perceive digital images or movies at a high level. It aims to automate things that the human visual system can perform from an engineering standpoint. Computer vision is concerned with the development of visual capabilities for a device, such as a computer, using complex algorithms and vast datasets. Researchers in computer vision have discovered methods for analyzing, comprehending, and classifying visual data provided by a picture or a succession of images. In order to reap the benefits of Automated Data Processing, we frequently require automated picture and video processing. To extract relevant and contextual information, computer vision algorithms capture, process, and analyze photos and videos. The development of very sophisticated applications was made possible by advances in computer vision technology like as image acquisition (cameras) and algorithms, as well as the arrival of multi-core CPUs and GPUs. The key to automating a wide range of operations based on visual context is computer vision. Automated computer vision applications are utilized for a variety of purposes, including:

Visual surveillance systems that operate automatically

- Automated obstacle as vehicle detection
- Autonomous obstacle as vehicles and driver assistance
- Analysis and diagnosis of medical imaging
- Human-computer interaction based on gesture recognition and motion
- Visual inspection in industrial and robotic applications that is automated

Even in the presence of complications such as background, occlusion, noise, coloring, scale, rotations, and so on, humans can easily deduce valuable information from a visual scene by performing face recognition, recognition of human activities and

gestures, object detection and recognition, and so on. Handcrafted visual models of objects and scenes were used in many early computer vision applications. In a confined setting, the handmade technique works, but vision programmers must manage unlimited changes to achieve human-like performance in broad vision tasks.
Consider a computer vision application that must recognize a chair in a series of photographs; backdrop variations, lighting conditions, scale, coloring, transformations, and occlusion are just a few of the aspects that affect the recognition process' performance. It's also impracticable to create a model for every possible case by hand. A computer vision system must be able to abstract the features of a chair, such as the legs and flat cushion, while removing the noise, such as colors or lights. Another necessity is the flexibility to add and remove other parts like as arms or a chair back without sacrificing generality or distinctiveness.

Computer vision programs can currently detect and monitor the movement of a single item class in an unlimited environment, such as people, human faces, or autos. Smart cameras can now do human face recognition (R. Subban et al 2015), image-based online search (Adrakatti at al 2016), surveillance applications (Hampapur et al 2003), and driver assistance features thanks to this capability (Saravanan et al 2010).

Since the 1990s, researchers have been focusing on single object recognition. Ren et al 2010) reported a 63.7 percent object identification rate for 42 commonplace objects in 2009, with no background cluttering or occlusion. For a database of 1000 classes, the ImageNet project achieved an object recognition rate of more than 95% in 2015. In 2015, (Lake et al 2017) outperformed humans on a small range of vision-based activities. The use of very sophisticated learning models, as well as a better knowledge of the composition and design of modern learning models, is credited with improving identification rates and adaptability to recognizing cryptic and partial objects. It's crucial to note that though Machine Learning methods are frequently used in Computer Vision jobs, the two names are not interchangeable. To achieve the intended goals, Computer Vision may employ pattern matching or other processes.

1.8 Reasons for Using Machine Learning for Obstacle as Vehicle Detection

Machine learning is defined as "a set of strategies for automatically detecting patterns in data and then using those patterns to forecast future data or make various types of uncertain decisions." Machine learning automates the process of developing an analytical model based on data analysis and then using that model to generate data-driven judgments, classifications, or predictions. Machine learning methods allow computers to learn from data without having to be explicitly programmed. Machine learning algorithms can learn and adapt to new trends in incoming data on a constant basis.

Many study areas, including natural language processing, targeted advertising, sentiment analysis, recommendation systems, bioinformatics, and others, use machine learning. The depth and complexity of recent machine learning models have enhanced machine learning's ability to solve challenges previously believed impossible. Machine learning algorithms can currently be utilized to deliver a very accurate answer to issues with a lot of ambiguity.

Machine learning methods are useful in computer vision applications because they increase automation. First, features are retrieved from input photos or videos, and then the extracted features are used to train the model using a learning algorithm. During the training phase, a series of photos with various backgrounds, lighting, scaling, and other factors are employed; machine learning algorithms generate models based on their similarities and differences, resulting in flexible and resilient vision applications. Object detection, recognition, and tracking, segmentation, object and picture categorization, picture registration, 3D reconstruction, optical character recognition, and other computer vision operations all use Machine Learning methods.

1.9 Reasons for Using Data mining for Obstacle as Vehicle Detection

Data mining is a popular technology that has a lot of potential to help businesses. Data mining techniques can forecast future trends, behaviors, and knowledge-based decisions. The practice of obtaining valuable information from enormous amounts of data is known as data mining. In other words, data mining is the process of extracting knowledge from data. The classification of data mining systems is based on a variety of factors, including data kinds and data models. By leveraging previously classified data,

data mining creates classification models and identifies the predicted pattern. In each situation of class, classification issues are utilized to identify the features of the group. Data mining can lead to the discovery of new information and the generation of a huge number of rules. It should be applicable to any method, methodology, or repository of knowledge. Relational databases, object-oriented databases, data warehouses, repositories, transactional databases, semi structured databases, and unstructured databases are all studied using data mining. To classify novel items, data mining and classification algorithms are commonly utilized. Image mining is a subset of data mining and is one of the most important research fields in computer science. Image classification is a machine learning classifier that is taught using training data. Unlabeled or unknown vectors can be classified by the trained classifier. Data mining approaches, such as classification, sequential patterns, and prediction models from many forms of data, are at the heart of the KDD process. Image mining is a computational procedure for locating and locating valuable information in large amounts of data. Image mining is not the same as traditional mining. Image mining techniques, such as supervised image classification and unsupervised image classification, are very active research areas. Image mining extracts low-level information including shape, color, and texture. For image mining, the extracted characteristics denote visual content. A classifier is trained in image classification using a training set. Unlabeled or unknown trained classes are classified using the trained classifier. Visual mining is the process of extracting useful image material from a large data set of images.

1.9.1 Image classification for obstacle as vehicle detection

Its method is based on the nature of the data analyzed and automatically classifies all pixels in images that are used in a hybrid fashion. Image mining is used in image retrieval, image processing, machine learning databases, and data mining to extract knowledge and image data relationships from images. Image mining differs from image processing and low-level computer vision in that it involves extracting patterns from a large amount of data. Image processing and computer vision are used to extract certain features from a single image. Image mining is nothing but applying existing data mining

algorithms on images. Between an image database and a relational database, there are some distinctions. The techniques for image mining look into the best framework for picture mining. Classification, clustering, image retrieval, image indexing, association rule mining, neural network, and object identification are examples of image mining algorithms. Unsupervised classification and supervised classification are two types of image classification. A collection of pre-classified photos is provided by the supervised classification.

1.10 Autonomous Vehicles and Driver Assistance Systems

Transportation of products and people is a critical activity in today's globe. It makes a significant contribution to economic prosperity and quality of life. Apart from the monetary cost, pollution, resource consumption, exhaustion from driving and traffic congestion, and personal safety hazards from accidents are some of the negative consequences. The automobile industry is constantly attempting to mitigate negative consequences, such as improving vehicle and road safety. Driver Assistance Systems are utilized to improve mobility efficiency and active and integrated safety. To improve traffic safety, a wider market penetration of Driver Assistance Systems is essential (Lu et al 2006). The following are a few examples of developing Driver Assistance Systems (Bengler et al 2014):

- Cruise Control with Adaptive Settings
- Autonomous Parking
- Collision Avoidance and Detection Systems
- Detection of Drowsiness in Drivers
- Intelligent Speed Adaptation
- Lane Departure Alert Systems

Driver assistance is the most basic form of driver automation, while driverless vehicles are the most advanced. In order to plan and execute a safe driving algorithm, autonomous cars require sensing and perception of the environment. Autonomous cars are capable of perceiving the environment and navigating without the assistance of people utilizing a variety of sensors such as RADAR, LASER, GPS, ultrasonic, odometry, cameras, and so on. Control systems assess sensor data and extract information in order to spot impediments, comprehend road signage, determine

appropriate routes, and finally drive the vehicle to its destination. Although highly autonomous vehicles are not yet commercially available, every automobile manufacturer is developing its own autonomous vehicle. Google began working on autonomous vehicles in 2009, and in 2017 it launched its early rider programmed, inviting people of Phoenix, Arizona, to participate in a public trial of self-driving cars. In June 2017, Audi stated that the next Audi A8 will be capable of self-driving at speeds of up to 60 km/h. It is the first mass-produced car to reach such a high level of automated driving.

1.11 Obstacle Detection Model System

Vision-based automatic car sorting hooked on replicas remains a serious chore aimed at AVS besides extra ITS submissions, besides the situation container remain thought of as a difficult multi-class fine-grained image classification problem, with each "class" representing a specific vehicle. Most projects begin with an automobile discovery phase that makes ROIs that comprise the automobile's forward-facing before rear-view opinion, separated as of the contextual. The structures derived from these ROIs are subsequently used by vehicle categorization systems.

In many police operations, such as responding to a hit-and-run, an amber alert, or a high-speed chase, the vehicle detection and model are often available regardless of lighting conditions. An eyewitness may also recognise the licence plate number; however, it is not always visible or only partially visible. An ODMS system is extremely useful for vehicle tracking and identification that is based on the vehicle's look rather than the connected licence plate.

Authorities can use the ODMS system to search for all similar vehicles in a given location during a given time period based on the vehicle's description or partial number plate. Road traffic organization, motorist support organizations, road traffic behaviour examination, road traffic specialist care, and other applications all require vehicle classification and identification.

The goal of this study is to look into feature extraction and supervised learning in the context of a vehicle model detection system. Because they were recorded in various weather situations, with variable lighting exposures, partial occlusion (e.g., pedestrians), and varying viewing angles, the vehicle photographs used in our work mirror the real-

world situation. The overarching goal is to see if supervised learning can handle the applied computer vision problem of detecting the make, model, and generation of automobiles despite the issue environment's strict constraints. The proposed ODMS system can detect vehicles based on model under heterogenous climatic condition.

1.12. Weather and Obstacle Detection Statistics

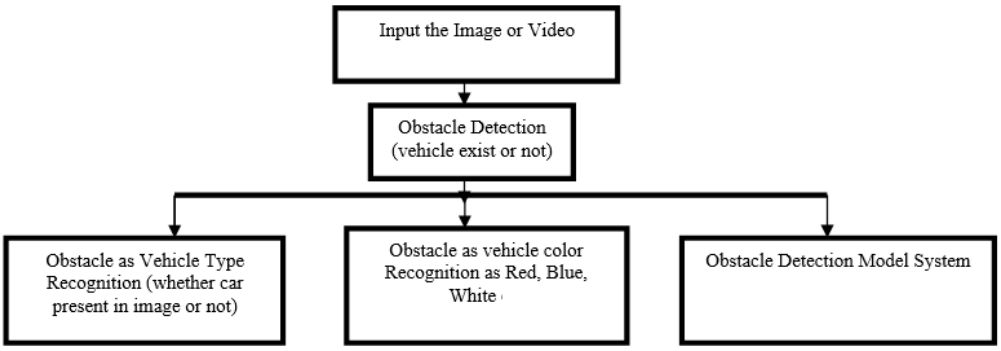
Given the complexity of urban structures, static and dynamic items such as crosswalks, parking lanes, sidewalks, zebras, pedestrians, and cyclists, great attention should be dedicated to resolving system failures in urban settings. Only on a well-structured route, such as motorways, can a UV-disparity based resolution build a real-time map of the surroundings. It does not, however, perform well in a complicated setting such as a city. According to a database, out of 610 disengagements, just about 16% were due to gloomy weather, while the remaining 1% were due to rainy and snowy days. The majority of errors have nothing to do with adverse weather, as 82 percent of disengagements occurred in pleasant weather. As a result, in addition to shadows and reflecting surfaces that are still difficult to discern, the impact of rain (high humidity condition) and snow (perhaps extremely reflecting during the day) would be a unique scenario to investigate (Kim et al 2015, Favar, F et al 2018). Light conditions and various weather conditions are two types of environmental elements that have a significant impact on visibility. It was determined that poor visibility in unusual weather could influence drivers who experience threat-related emotions such as fear and worry (Azam et al 2016). Under adverse weather circumstances such as rain, fog, or snow, the vehicle must be handled with absolute safety. Laser sensors seems to be the most suited hardware for autonomous driving, given their strong capabilities. However, it has a significant flaw in its perception performance under adverse weather conditions such as rain, fog, and snow (Rasshofer et al 2011) It is raining. Small water droplets make up both rain and fog. These droplets are likely to behave as reflectors during obstacle detection, causing false alarms (Rasshofer et al 2011). In rainy situations with rain intensity less than 20 cm difference, LIDAR's laser path and reflection rate on material surface changes function well. Rain is thought to be a periodic noise that alters the image of the photographed item. By using an automatic rain removal methodology, any coefficient value less than 0.5 is replaced by 0, considerably improving the quality and accuracy of the captured item. An image of an object was

acquired before and after the removal technique was implemented, as shown in the example. The distinction is obvious (Azam et al 2016). The fog is thick. During foggy weather, the link between temperature and visibility is in inverse proportion, according to (S. Hasirlioglu et al. 2017) research. This attenuation refers to the molecular intensity of water in fog, whereas visibility refers to how far the laser can reach in that situation. Fog causes the laser sensors' detecting range to be reduced considerably at lower temperatures (McKnight et al 2014). According to the Chinese Academy of Social Sciences, haze and fog have been the most important problem in northern China since 1961. (Wang, et al 2014). During the worst instances, visibility in Harbin was below 50 metres due to coal-fired heating systems and inadequate automobile emission standards (Wang, et al 2014). Experiments were carried out in both real foggy conditions and simulated laboratories, demonstrating that the LADAR and vision cameras had some difficulties scanning and assessing the driving environment and surroundings accurately (Zhu et al 2015). In order to remedy this scenario, an infrared sensor can be used to detect objects in foggy situations. Snowfall During driving in snowy winters, the sensors of LIDAR/RADAR can be obscured by snow. To prevent snow accumulation, either use the camera system behind the window or spin the laser-detector at a high angular speed. In snowy weather, cameras and LIDAR optical sensors have shown to be extremely challenging to use (Rasshofer et al 2005). If optical sensors can't observe important surfaces and label both the road ahead and the surroundings around it, the AV car will hardly stay in one lane. Furthermore, due to reflections, a build-up of snow on and beside the road might cause the LIDAR beams to act as a "phantom obstacle," causing a false assessment of the driving environment (Radecki et al 2016). The Localizing Ground-Penetrating Radar (LGPR) technology, which keeps the vehicle in its lane by placing certain specific tubes and a radar sensor under the chassis to emit and receive signals, can be utilised to tackle this issue during harsh weather such as severe fog or snow. r projects a pulse of electromagnetic radiation.

1.13. Process of Obstacle as Vehicle Analysis

Automated vehicle analysis is becoming increasingly crucial in various scenarios as the number of vehicles grows. The methodology of projecting the worldwide vehicle numbers is inexact.

Figure1.6. Demonstrate the Process of Obstacle as Vehicle Analysis



However, research have revealed an exponential rise. The present worldwide car count is projected to be above 1.2 billion, with research predicting that this number will exceed 2 billion in 2035 or 2040. Automatically analysing and monitoring cars for various uses is highly desirable. The vehicle analysis hierarchy is illustrated in the diagram. The detection of vehicles is the first step in vehicle analysis. We can classify the vehicle based on its class (car, bus, truck, etc.), color (white, black, red, grey, etc.), or model once it has been spotted (Toyota Corolla, Hando Accord, Ford Fusion, etc.). Automatic vehicle analysis is used in a variety of applications, including autonomous vehicles and driver assistance, surveillance, law enforcement, intelligent transportation systems, and smart cities. It is hard for humans to manually watch, evaluate, and analyses the growing number of automobiles, particularly in urban areas. A computer vision application, unlike a human operator, can monitor traffic:

- For an extended period of time,
- There are fewer errors owing to tiredness.
- Less expensive
- At a higher rate,
- At a performance level that is scaled to meet a desired performance/cost ratio.

A ODMS system can detect of a vehicle from a variety of angles and distances, and in a variety of weather conditions. It is anticipated that ODMS will be useful for Autonomous Vehicles and Driver Assistance, Surveillance, Law Enforcement,

Intelligent Transportation Systems, Smart Cities, etc. Machine learning algorithms are expected to play a vital role in the development of these fields.

1.14 Recognition and detection of obstacle as Vehicle Shapes

The goal of this study is to look into feature extraction and supervised learning approaches in the context of the Vehicle Detection Model System. Because they were recorded in various weather situations, with variable lighting exposures, partial occlusion (e.g., pedestrians), and varying viewing angles, the vehicle photographs used in our work mirror the real-world situation. The overarching goal is to see if supervised learning can solve the applied computer vision problem of identifying car models given the issue environment's tight constraints. The proposed ODMS system recognizes vehicles on the model, systems can only identify the model. We are using the term to describe the Vehicle Detection Model System having the same physical appearance.

1.15. Problem Statement

Vehicle as obstacle detection is becoming more vital in the road traffic management system as the number of vehicles in the country grows. The detection of obstacles such as vehicles has recently piqued the interest of the research community. Researchers are attempting to delve deeper into the subject in order to arrive at a degree of accuracy that is acceptable. Detecting objects is done via machine learning. There are numerous strategies for achieving this, however in order to find the best model among the suggested models, this thesis will look into how to detect an obstacle as a vehicle while comparing proposed models and recommending the best one that gives the best accuracy and performance. Due to no Street lights entire ghats section expressway Extreme incurable accidents consume remained stated happening the thruway popular 2016, popular the 151 public pass away.

1. The majority of accidents in the ghat area occur when automobiles slam into the motorist in front of them, according to accident trends. One reason for this could be that the driver was unable to distinguish the obstructions as automobiles.

2. CCTV deployed on roadways can identify various traffic accident causes such as vehicle over speeding, wrong way driving, collision, and accident. According to

Maharashtra data, accident rates are higher during the monsoon, which leads to accidents due to poor visibility.

3. Unpredictable and uncontrollable lighting conditions

4. Consider the following case. The RTO currently analyses each scanned licence plate in Mumbai to centralized and local databases. At the end of the day, all licence plates that do not match the databases are removed from the system. As a result, identifying the vehicle's owner may be problematic in some situations. Occlusions challenging scenarios, such as low-to-heavy occlusions (due to pedestrians or other objects), partially hidden vehicle views, closely adjacent vehicles, and so on.
5. Modified Appearance: when a vehicle's original appearance is altered, such as by painting a different pattern on the hood or trunk, modifying the bumpers or grill, adding spoilers, or installing neon lights beneath the bumpers, among other things.

1.16 Challenges and Issues

When compared to other objects, vehicles have various distinct characteristics. They present a more varied and difficult collection of problems, as well as a variety of new research topics in fine-grained image categorization. In vehicle detection, there are two types of challenges: (1) problem of multiplicity and (2) Problem of ambiguity (J.-W. Hsieh et al 2014).

The problem of multiplicity arises from the fact that different years of the same make's vehicle models have varied shapes and/or appearances. The shape of automobile is typically reshaped by dissimilar periods toward chance marketplace strains. The problematic of diversity trendy the dataset's model photographs. Aimed at instance, the modifications among photos of Honda Civic since 1986 towards 2015 stand variations popular the outline of the appendage illuminations, plenteous, or else, in some cases, a non-compulsory accumulation of rearmost coddler.

The vagueness problematic can remain separated hooked on deuce groups: inter-class similarity besides intra-class inconsistency. The initial vagueness concerns automobiles from several manufacturers that have a visually alike form or presence, that is 2

dissimilar brand prototypical programs partake equal forward-facing or back perspectives. The back-view appearance of the "Ford Escape 2010" and the "Mazda Tribute 2008," for example, is aesthetically identical. The latter type of uncertainty arises from the similarities of several models from the same manufacturer. The “Toyota” “Altis” and “Corolla” models, for example, have a striking external resemblance.

When characteristics are computed from the background, another important issue can occur. This may have an impact on the learning mechanism. Although many other fine-grained applications may encounter similar concerns, vehicle detection is one of the most difficult fine-grained classification challenges because to the enormous number of automobile models, which includes multiple car brands and models depending on the year. As a result, this application has the potential to nurture more advanced and reliable computer vision models and algorithms.

The detection of obstacle as vehicle problem can be treated as a multi-class image classification problem, where each class represents a specific vehicle’s make and model. However, more challenging and diverse problems are associated with vehicle detection system as compared to other image classification problems. A few of the challenges are listed below (G. Pearce et al 2011).

1. Image acquisition in a natural setting.
2. Unpredictable and uncontrolled lighting conditions.
3. Unpredictable and changing weather conditions.

A vast range of vehicle looks are offered. Though VDMS systems have lately attracted interest, significant research is still needed to produce systems that are robust to real-world situations and perform in real time to solve the aforesaid drawbacks. With advances in human face recognition and image search technologies, the idea of automatically recognising the kind, make, or model of a vehicle seen by a camera may seems extremely straightforward. ODMS, despite its seeming simplicity, has its own set of obstacles and concerns, which are listed below: There is a wide range of vehicle types (sedans, vans, SUVs, trucks, and so on), makes and models, and colours.

1. Multiplicity: when the shapes, styles, or sizes of vehicles in the same class differ
2. Ambiguity: when the shapes or designs of cars in different classes are similar

3. Heterogeneous Views: Viewpoints, scales, rotation, camera position, and camera pitch angle can all be changed.
4. Lighting Conditions: images captured under various lighting conditions (bright sunlight, glare on vehicles, shadows of objects upon vehicles, dusk, night, etc.)

Figure 1.7 Demonstrates the Vehicles in Various Lighting and Weather Conditions.



Obstacle as detection System: Models, Methods, and Techniques

1. Weather Conditions: a variety of weather conditions include sunny, cloudy, rainy, foggy, snowy, and so on.
2. Occlusions: difficult conditions include low-to-heavy occlusions (due to pedestrians or other objects), partially concealed vehicle views, and closely neighbouring cars, among others.
3. Modified Appearance: when a vehicle's original appearance is altered, such as by painting a different pattern on the hood or trunk, modifying the bumpers or grill, adding spoilers, or installing neon lights beneath the bumpers, among other things.

4. Dataset Availability: Researchers are unable to compare Automated Vehicular Surveillance approaches because to a lack of publicly accessible benchmark datasets.

1.17 Objective and Contributions

We plan to use computer vision algorithms to highlight obstacles such as vehicle detection systems and traffic control systems with intelligent functions in this dissertation. For the implementation, proof of concept, and testing of computer vision classification algorithms, some video cameras will be used. Our research's overarching goals are to improve the research community's collective awareness of the potentials

and limitations of the methods we utilize. The objectives defined for this thesis are the following:

1.17.1 To detect cars with a high degree of precision.

1.17.2 To investigate various feature extraction approaches. We are concentrating on GIST image features because it treats the image as a whole entity and computes features based on the entire image, whereas other feature extraction approaches attempt to pinpoint salient interest points in the image and compute features based on these places.

1.17.3 To improve the performance of robust features and produce efficient dictionaries for scale-invariant feature transforms.

1.17.4 To compare the performance of two existing categorization techniques in terms of model recognition. In this paper, we also proposed a ODMS-ABSVM classification approach & ODMS-DCSVM.

1.17.5 To get a thorough understanding of the properties of comparison models, as well as the reasoning behind performance discrepancies, particularly in the areas of accuracy and speed.

As a result, we employed obstacle detection to test and assess traditional machine learning algorithms such as Support Vector Machine and Random Forest. We contributed to the creation of complicated feature extraction, including scaling and normalization, in order to do so. We also proposed the ODMS-ABSVM & ODMS-DCSVM classification strategy, which generates a new feature representation and improves recognition rate. In terms of scaling and normalization approaches, (Trabelsi et al 2017). As a result, we believe our contribution to machine learning research has improved.

1.18 Motivation for the Study

Computer Vision and Machine Learning are fields that have a wide range of applications, with sensors based on image capture and video streaming becoming increasingly popular. The result can be used for a variety of things, including counting vehicles by class, detecting occurrences, and identifying vehicles that are acting strangely.

They employ information and communication technologies to the transportation sector through intelligent solutions that minimize accidents, CO2 emissions, and enhance traffic

conditions, while ensuring values like sustainability, efficiency, and safety. It has been working on the topic of vehicle detection and tracking and has developed a solution. This suggests that the findings of this thesis in the areas of ITS and obstacle detection may be of interest to the company. According to the World Health Organization, around 1.35 million people die each year as a result of traffic accidents, making traffic injuries the top cause of death worldwide. This rise is due to increased traffic congestion, public safety concerns, and government measures to improve and modernize traffic management. In the interest of modern government infrastructures traffic management systems, which also demonstrates a favorable attitude toward the place and environment in which our problem is embedded, because the input video data used in this work will be from the considered area. Currently, to detect the presence and absence of vehicles on the road, when humans recognize vehicles while viewing a video, there is a desire to make the work easier for humans, but not necessarily to remove them totally from the process.

1.19 Structure of the Thesis

The following is the structure of this thesis:

Chapter 1 Introduction

Chapter 2: Literature Review

The second chapter contains a complete literature review on offer literature review. Image processing and machine learning / data mining approaches were used to conduct a detailed review of vehicle as obstacle recognition and classification studies. Researchers have been employing vehicle detecting algorithms to have a better grasp of the obstacle. Many ways for discovery consume remained obtainable, counting appearance-based replicas besides motion-built replicas and the profits of these plans besides what remain the breaches that these plans are unable to fill .

Chapter 3 addresses the research methods employed in our study, as well as the complete system design and a description of the dataset used in this study. an proposed a Computer Vision strategy for autonomous obstacle as vehicle detection as the optimum solution due to the minimal infrastructure adjustments necessary. Computer vision methods do not necessitate any vehicle modifications. Vehicle identification and tracking, as well as recognition of vehicle parameters such as color

and model, are all part of automatic vehicular analysis. Vehicle analysis can be used to produce a variety of applications, including electronic toll collection, intelligent parking lots, and autonomous vehicles. It covers thorough descriptions and discussions of feature extraction and global representation methodologies in well-known ODMS studies, as well as the suggested BoF-based ODMS approaches with various classification approaches.

Chapter 4. provides a detailed examination of the Results and Performance Evaluation for the Proposed Obstacle Detection Model System, as well as comments on its performance. Two machine learning algorithms and four image feature extraction algorithms were used in the research covered by this thesis. ODMS employs two machine learning methods: Support Vector Machine and Random Forests, as well as four image features: SIFT, SURF, HOG, besides GIST. All two machine learning algorithms were used with the four feature extraction systems. SIFT and SURF use local image features that have been grouped into visual dictionaries to generate a global feature vector using a Bag of Features (BoF) approach. Multiple dictionaries of various sizes are used in an attempt to find the most appropriate size. Based on the HOG parameter configuration, developed different HOG image feature vectors of varying sizes. Finally, in this work, we proposed an ODMS-ABSVM and an ODMS-DCSVM classification to improve the recognition rate. We compare the outcomes of all ODMS system variants. In terms of accuracy and computation time, it assesses the efficiency and performance of the proposed ODMS approaches.

Chapter 5 Finding of Heterogenous Model, Conclusion and Future Work concludes by summarizing the work done in this thesis, and presents future directions of research for Obstacle Detection Model System.

1.20 Keywords used: Keywords used include Obstacle Detection, Computer Vision Approach, Obstacle Detection Model System, Support Vector Machine, and Random Forests. SIFT (Scale Invariant Feature Transform), SURF (Speed-Up Robust Feature), HOG (Oriented Gradient Histogram) and GIST (Bag of Features) (BoF). mACCR (mean Correct Classification Rate): (a) Single-Dictionary Construction Scheme; (b) Modular Dictionary Construction Scheme. ND number of training and testing.

Chapter 2: Literature Review

2.1 Analysis of Literature Survey

The purpose of literature survey is to understand the previous work done in the chosen topic. Many papers were read and analyzed to understand in which directions other researchers have moved in this field. When we started to do research on this topic, there were many areas, of which we had little or no knowledge. Detailed review of vehicle as obstacle recognition and classification research using image processing and machine learning / data mining techniques. Detailed understanding of the vehicle as obstacle detection techniques that researchers have been using till now. What are the benefits of these techniques and what are the gaps that these techniques are not able to fill. Many approaches have been proposed for detection which includes appearance-based models and motion-based models. Thus, the knowledge about each detection methodologies is must to measure pros and cons of each.

The literature review is described with the following sections:

1. Study of different techniques used for Obstacle as Vehicle Detection.
2. Study of Appearance Based Methods.
3. Detailed study of Feature based model methods used for object detection
4. Review on Part Based Model views vehicle as isolated into front, side and back parts to enhance the detection in occlusion and edge of camera.
5. Study of different Attributes used for vehicle Classification
6. Review of various Vehicle Tracking algorithms used in vehicle as obstacle tracking algorithm
7. Study of different machine learning/data mining algorithm used for vehicle as obstacle detection

2.2 Obstacle as Vehicle Detection

Obstacle detection is used for roads, highways, parking otherwise any other place to identify the number of vehicles existing on the spot. This determination will help the

surveillance to evaluate the traffic vehicles, average speed besides grouping of vehicle. There are amount of obstacle detection methods are obtainable to detect also classify them. Obstacle Detection exists a method that discoveries examples of world obstacles like pedestrians, faces, vehicles besides buildings in pictures or in videos.

(Shukla et al 2015) have discussed about Traffic Analysis with Incident Detection, Segmentation Methods and Vehicle Tracking Methods. However, the Approaches with adding study tools to detect, segment, besides track the vehicles. Thus, this paper depends on advanced information of features also their motion behaviours aimed with vehicle identification, which define the difficulty in identifying vehicle shapes, colours, and types. However, the author has extended his work using the optical flow method and Background Subtraction technique which benefits to conclude the speed of the vehicle through the video sequence.

(Bashkir et al 2014) proposed an automatic vehicle counting system for processing of videos recorded from stationary cameras in roads e.g., CCTV cameras installed near traffic intersections junctions and tally the quantity of cars passing a spot in a specific period for additional group of vehicle data. Authors proposed a simple approach to tackle the problem of Gaussian mixture model-based obstacle detection, a non-predictive regional tracking besides a counting of tracked obstacles founded on simple rules. On the basis of improvements, the authors developed a unique procedure for vehicle information recognition also tracing by Gaussian mixture model and blob detection methods.at first, they differentiate the forefront as of contextual in frames by learning the background. Thus, the foreground detector detects the obstacle and a binary computation is done to define rectangular areas about each detected obstacle. To identify the moving obstacle properly and to remove the noise specific morphological processes must applied. Then the ultimate counting is complete by tracking the spotted obstacles and their regions. Thus, the limitation of developed method is that for every camera data feed a considerable amount of tuning of the parameters is required to achieve the best performance. Also, it requires somewhat more processing time in highly danced traffic conditions.

(Changlasetty et al 2013) offered a system intended for identification and classification of vehicles wherever vehicles are traced via using width, length also parameter area mined using image techniques. Thus, Traffic data was recorded by stationary camera. Background of the image is extracted via using mixture model. For the classification LABVIEW and neural networks is trained toward for classifying vehicles with data monitoring WEKA tools. It is also used in intelligent transport system cites. Feed-forward neural network remains trained toward classify vehicles using data mining WEKA toolbox.

(Ding et al 2015) presented forward the obstacle detection algorithm based on laser radar data of three dimensions in the hybrid cross-country environment. Firstly, they use the method of the maximum then blurred lines toward split each laser-line projection in x-y plane. Then, K-means clustering remained applied to integrate the blurred line and also use the corner detection algorithm to accurately relocate the endpoint of the integration segment. And third step is related to the unidirectional graph of the Markov random field, which helps in marking and detecting the obstacle area. Lastly, the function is calculated by means of analysing line features and solved by the graph cut. The line segments are finally classified into two categories: the ground and the obstacle.

(Gang et al 2013) proposed an improved algorithm which combines three frame differential method and image edge detection algorithm to get the complete profile of the moving target, fill the regions of the target with a series of mathematical morphology and connectivity detection remove small-scale noise, and make the complete moving region appear. Author proposed method that proved effective via experimental. Thus, the suggested algorithm still has some limitation in consideration environment with strong light and obvious shadow. It still needs further improvement. For backgrounds with changing conditions and jitter, the effect of this method to detect moving targets is not very ideal and authors need to further research and exploration.

(Chitaliya et al 2012) proposed a vehicle classifier by the discrete Curve let transform to increase the effectiveness of classifier, 3 class structures considered with detail to the ratio of length and width of the vehicle or person going on the road. Every image is pre-

processed with unsharp filtering which offers edge also converts it into a binary image. Binary image remains converted into distinct curve let heaving low frequency also high frequency. Feature vector matrixes remain drawn also Eigen value is estimated. The experiments conceded out on dissimilar types of vehicle images. The outcomes of the classifier display the efficiency to handle the real time dataset.

(Hassannejad et al 2015) developed a monocular algorithm for detection of moving obstacles in roundabouts used in Advanced Driver Assistance Systems. Thus, the proposed algorithm fuses motion-based approach also appearance-based approach on the way to income monocular system limitations in obstacle detection. It processes images sequence which is extracted from video stream of a monocular camera mounted on front left guard, adjacent to the head lamp. The algorithm was customized on the characteristics of traffic movement in roundabouts also it benefits to require a safe entrance into the roundabouts. Firstly, the algorithm spots the independent moving features of the images then clusters to discin the moving obstacles. After, it uses Haar-like features alongside Soft-Cascade AdaBoost to spot the vehicles which are moving towards/away to/from the car. The result was tested on selected validation set also it used in some real drives.

(Zhan et al 2014) offered a novel fine-grained vehicle recognition system that uses the Collective feedback scheme aimed at detection-classification-tracking in surveillance video. The researcher takes data 200cars heaving 3 categories besides some negative tasters. The Harr-like feature are extracted from the image offline also SVM be used as a classifier intended for interclass variation. The detector localizes the motion obstacles in each frame excluding that must been observed and also tracked, and the best matching obstacle through the assumed initial car is recognized by the classifier and remain tracked in the subsequent frames till it is not present. The experiment result demonstrations that this approach can successfully locate the finest matching car from the frames when the target car appears.

(Deb et al 2011) developed an automatic driver assistance system toward to alert a driver about driving environment. The common method used toward vehicle detection is active sensor such as radar-based system, Light Detection and Ranging also acoustic based.

Radar based system can realize 150 meter forward in fog or rain. To discern the location of vehicles, the three methods exist i.e. knowledge based, stereo based and motion based. Thus, the vehicle identification is done on the basis of symmetry, colour, shadow, corners, vertical and horizontal edges, texture, and vehicle light. Thus, considering the stereo vision system, the three methods used are disparity map, inverse perspective mapping and motion based and the location are finding through using template based and appearance.

(Kaur et al 2015) Authors has drawn assessment among existing and also proposed methods of lane detection. Lane detection methods exist proved to be helpful in avoiding accidents on highways. The system able to detect the lane marks even in the presence of high level of noise in the input image. The proposed technique by author has proved as more efficient than that of existing technique but still there is need for improved canny and also the threshold for fixed aimed to find different binarization technique to make adaptive and most accurate result.

(Radhe et al 2014) presents an algorithm for on-road vehicle detection problem using cascade of boosted classifiers. This approach filters different non-target (negative) samples in dissimilar steps of cascaded structure according to their level of similarity through target obstacle class. Then matched the performance of local then global texture features in grouping with boosted Haar like features. The finest performance for on-road obstacle detection is achieved by Adaboost with Haar-like feature along with SVM then histograms of oriented gradients (HOG) features. The outcomes tell that the obtainable approach improves the detection efficiency.

(Boroujeni et al 2012) contributions for a highly efficient and accurate method for detecting obstacles in support with crowded also the simple scenes. The system was built on obtaining single camera color images besides through no depth information present. An adaptive method for background detection remained originally employed also the foreground sections of the scene remained further examined with two types of optical flow, Horn-Shuck and Lucas-Kanadead and thus steps were carried out on each color channel and also the results were combined together. Thus, the Two benchmark

sequences, Bahnhof and Jelmoli exist for testing system. Likewise, numerous sequences remained obtained from a hallway through obstacles placed in the pathway of the moving cart mounted using a camera. Finally, outcomes demonstration very accurate besides robust extraction of obstacles even though a speed increase of above 80% is accomplished in with standard optical flow methods alone. The outcomes indicate that targeted optical flow eliminates background outliers therefore making it more appropriate for obstacle detection.

Two models for vehicle detection includes Appearance based strategies and Motion based techniques (Shukla et al 2015). Appearance based strategies uses highlights like shape, color shading and size of vehicles to recognize vehicle. Motion based detection models detects an object relative to its surroundings. Different detection methods are listed in Table 2.1

2.2.1 Appearance Based Methods

Appearance based strategies utilizes prior knowledge to segment the foreground and static background. Appearance based approach can be feature based or part based.

Table 2.1: Demonstrates the Representative Works Vision-Based Vehicle Detection

Appearance Based Methods	
Feature Based Model	
.SIFT	Local edge orientation, invariant scaling, translation rotation (Tian et al 2015)
SURF	Local feature around point of interest (Lin et al 2009)
HOG	Edge of gradient structure that represents local shape (N.Dalal et al 2005)
HAAR	Edge, time and center-oriented features (H. Bai et al 2006)
Part Based Model	Model by spatial relation between different parts like top, window and wheels (Hoieum et al 2007)
Motion Based Model	
Frame Differencing	Moving vehicles detected using pixel wise difference between consecutive frames. (Li et al 2011)
Background Subtraction	Parametric Model Non-Parametric Model
Parametric Model	
Frame Averaging	Averaged frame is subtracted from consecutive frames. (Kanhere et al 2008)
Single Gaussian	Background is computed in the form of cumulative running average and standard deviation (Chen et al 2006)
Median Filter	Estimated by finding median value for each pixel from a set of frames. (Li et al 2011)
Gaussian Mixture Model	Distributions are estimated as background or foreground by evaluating its stability. (Haque et al 2008)
Non-Parametric Model	

KDE	Recent data are used to estimate probability of each background pixel (Wan et al 2008)
Code book	Dynamic code words distinguish background from foreground. (Mittal et al 2004, Kim et al 2005)

2.2.1.1 Feature based model

Feature based models utilize physical features of the vehicle. Low level component approach utilizes low level features like shading shadow, symmetry and edge. Low level features have been supplanted by string descriptors. Various component descriptors approaches have been proposed Scale Invariant Feature Transformation (SIFT), Speeded Up Robust Features (SURF), Histogram of Gradients (HOG) and Haar like features.

- Scale Invariant Feature Transformation (SIFT): In SIFT features are extracted with a filtering approach. The features so extracted are invariant to illumination, projection scaling, translation and rotation (Tian et al 2005).
- Speeded Up Robust Features (SURF): SURF is a scale and orientation invariant detector. The computational complexity of SURF is reduced in contrary to SIFT as Gaussian filters are replaced by Box filters. (Messelodi et 2005)
- Histogram of Gradients (HOG): HOG computes the image gradient directional histogram which integrates gradient and edge information. The fundamental highlight of this approach is its illumination, geometric invariance and computational effectiveness. (Dalal et al 2005)
- Haar like features: Haar like features are formed using sum and differences of rectangles over an image portion to highlight gray level distribution of adjacent regions. Two, three or four filters can be used at any point and scale to extract the features.

The output is computed by adding the pixel values for grey and white regions and the difference between two sums are normalized. (H. Bai et al 2006).

2.2.1.2 Part Based Model

In part-based model the vehicle is segmented into smaller parts and spatial relations between them are modeled. Part based models' views vehicle as isolated into front, side and back parts
to enhance the detection in occlusion and edge of camera.

2.2.1.2.1 Motion Based Models: Motion based methods extract moving vehicles based on motion from background. Motion based methods includes Temporal frame differencing and Back ground subtraction.

1. **Frame differencing:** Frame differencing is least complex and quickest method. Pixel wise difference is figured between two back-to-back frames. Moving foreground regions are determined using a threshold value (Munroe et al 2005). The detection can be improved using three consecutive frames. Dual inter frame subtraction followed bitwise AND is performed to extract the moving object (Li et al 2011).

2. **Background subtraction**: Foreground objects are separated by computing pixel wise distinction between the present image and the static background image (Gaunpte et al 2002). The information about the background is accumulated to produce the background model. Background can be parametric or non-parametric.

- **Parametric Models:** Parametric model uses a unimodal probability on each pixel and update the distribution parameters

1. Frame averaging: Frame averaging is a conventional averaging technique where a set of frames are averaged. The resulting background model will be subtracted from consecutive frames this technique has high computational efficiency but has tail effects (Wan et al 2008). The accuracy depends on N which comes at the cost of memory requirements.

2. Single Gaussian: Background is modeled recursively using single Gaussian. This improves robustness and reduces the memory requirements. The background is computed recursively in terms of cumulative running average and standard deviation (Morris et al 2006). Each pixel is distinguished into background or foreground

depending on the pixel position in Gaussian distribution. This model reduces the cost but tail effect still persists.

3. Median Filter: The background is evaluated by locating the median value for each pixel from a set of frames (Wan et al 2008). This technique is adopted when background pixels will not vary rapidly with time. A recursive approximation approach estimates the median using recursive filter which increases or decreases by one whenever input pixel has a value greater or less than estimate and is not changed if equal.

4. Gaussian Mixture Model: Gaussian mixture models each pixel as a blend of two or more Gaussian temporally updated (Lin, et al 2009). The stability is evaluated to estimate the distribution as table background or short-term foreground process.

- **Non parametric Models:** Non parametric background models use pixel history to construct a probabilistic model of the observation using recent samples of pixel values and do not consider pixel value as particular distribution (Wan et al 2008). Kernel Density Estimation (KDE) and codebook are non-parametric models (Bouttefroy et al 2009).

1. Kernel Density Estimation (KDE): the nonparametric KDE characterize a multimodal probability density function (Mittal et al 2004). The probability of each background pixel is estimated from recent samples using Parzen window.

2. Codebook Model: Parameters represented by probabilistic function is replaced by a set of dynamically handled code word to model background (Kim et al 2005). Quantization or clustering is applied after this. Each codebook contains a number of code words. The new pixel is classified as background if the value of the pixel belongs to the code word range else classified as foreground.

2.3 Vehicle Classification:

The detected foreground may correspond to different objects in the scene. It is required to isolate and recognize the object of interest. The number of classes depends on the features and classification used. Vehicle classification involves extracting vehicle static attributes that includes color, license plate, logo and is summarized in Table 2.2.

Table 2.2: Demonstrates the Attributes Used for Classification

Color	RGB, HSV	(Yang et al 2013)
License Plate	Texture, Edge features	(Munroe et al 2005)
Logo	Edge, Features	(H .Yang et al 2013)
Type	Shape, Appearance	(Hoieum et al 2007)

Color Recognition: Color of a vehicle is an important attribute that has wide applications that include security and crime prevention issues. It includes RGB and HSV techniques (H .Yang et al 2013).

License Plate Recognition: Automatic recognition of license plate is done in 3 major steps: license plate localization, plate character segmentation and recognition. Localization of license plate use the edge, color, texture or features combination. Character segmentation depends on the issuing country due to different standards with respect to color, size and aspect ratio. Character recognition makes use of classifiers such as artificial neural network or support vector machine.

Logo Recognition: Vehicle logo plays and important role in vehicle classification as it provides information about the make and model. Logo detection is a critical step for logo recognition. Edge detection and morphological filtering are some techniques used for logo detection (H. Yang et al 2013).

Vehicle type classification: Vehicle type classification can be based on appearance or shape. Shape features include size, dimension and aspect ratio etc., The curves associated with 3D ridges where used to classify the vehicles in to 3 classes SUV, car and mini bus with an accuracy of 88% (D. shan et al 2005). Oriented counter point model was proposed to identify the vehicle type. The edges in the four-vehicle orientation from the front view were integrated with voting algorithm and Euclidean edge distance with an accuracy rate of 93.1%.

Appearance based techniques utilizes visual features such as edge, gradient and corner. This approach can distinguish a wide variety of vehicles with varying time complexity and accuracy depending on feature selected. K means uses edge features to distinguish five classes of vehicles (Munroe et al 2005).

Vehicle recognition still faces many challenges including variation in the illumination, camera field of view, similarity in vehicle appearances and large number of vehicle types.

2.4 Vehicle Tracking

Vehicle tracking is employed to obtain the vehicle trajectory by identifying motion dynamic attributes and characteristics to predict its position in subsequent frames (B. Tian,et al 2015) Vehicle tracking can be categorized into three models: motion-based tracking, region based and feature based tracking.

1. Model Based Tracking:

Model based tracking uses prior knowledge to create a geometric model of vehicle, which can be 2-D or 3- D appearance model. These models are used to match with moving regions and describe vehicle motion (Kim et al 2005). This method is accurate and robust but costs high. Different methods to model vehicles include multi view model and deformable template.

2. Region Based tracking:

Region based tracking detects vehicle outline as associated regions within rectangular, oval or any other simple shape which can characterized by area, coordinates, centroids, edges or intensity histogram etc. Tracking is performed by utilizing the association between the region characteristics. The regions were matched with Kalman filtering in shape-based approach (Scott et al 2009) Camshift algorithm is applied for video object based on histogram back projection.

3. Feature Based Tracking:

The detected features are used to perform matching in consecutive frames. The vehicle features are followed in a changed space instead of pixel space. Several techniques use combination of SIFT, HOG and Haar like features for tracking. Feature based tracking perform well in occluded conditions. The challenge is to appropriately choose features that efficiently represent the vehicles.

2.5 Machine Learning /Data mining

Data mining is considered to be one of the popular terms of machine learning as it extracts meaningful information from the large pile of datasets and is used for decision-making tasks. Machine learning is one of the applications of Artificial Intelligence (AI) which enables the computers to learn on their own and perform tasks without human

intervention. There are numerous applications of machine learning algorithms in the field of computer vision. With the help of machine learning, formulation of some of the most complex problems have been performed easily. Various computer programs which were previously programmed by humans, sometimes by-hand, are now being programmed without any human contribution with the help of machine learning. In the recent years, due to remarkable increase in the availability of humongous sources of data and feasibility of computational resources, machine learning has
become predominant with wide range of applications in our daily lives.

2.5.1 Machine learning as a tool in data mining

System learns through data fed to it to develop its own ability to make decisions and solve problems like a human mind. The algorithm that is used to achieve this ability is trained on data or test data. This learning-by-doing approach is what requires data mining. Data mining might be used to get the test data ready. This test data is used to train a given machine learning algorithm.

2.5.2 Types

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning

Supervised learning is considered to be the most elementary class of machine learning algorithms. As the name suggests, these algorithms require direct supervision. In this type of learning, the data labelled / annotated by humans is spoon-fed to the algorithm. This data contains the classes and locations of the objects of interest. Some of the popularly utilized supervised learning algorithms are:

- Neural Networks
- Decision Trees
- Random Forest
- K-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines

Unsupervised Learning

In the unsupervised learning, the algorithm tries to learn and identify useful properties of the classes from the given annotated data, without the help or intervention of a human. A priority algorithm, K-means clustering, etc. are some of the common unsupervised learning algorithms.

Reinforcement Learning

In this type of learning, the machine is allowed to train itself continually using trial and error. As a result, the machine learns from past experience and attempts to capture the best knowledge possible to predict accurately. Markov Decision Process, Q-learning, Temporal difference, etc. are some of the examples of reinforcement learning.

2.5.3 Data Mining Techniques

The final goal is to extract semantics as of video to be used in higher level action analysis jobs. Categorizing the kind of a noticed video obstacle is an essential step in accomplishing this goal. With the advantage of obstacle type evidence, supplementary specific besides precise methods can be established towards differentiate higher level plans of video obstacles. Hence, the outcomes of research established a novel video obstacle classification using data mining techniques created on obstacle shape correspondence as part of our visual surveillance system. Typical video scenes might contain a variability of obstacles such as people, vehicles, animals, natural phenomenon (e.g. rain, snow), plants and clutter. Though, main aim of interest in surveillance applications is usually humans and vehicles. Therefore, the below paper used Obstacle classification using data mining techniques.

(Mishra et al 2013) develop an algorithm proposed for detection and also classification of vehicle in heterogeneous traffic. The complete procedure is distributed in four phases i.e. camera calibration, vehicle detection, speed estimation, and also for classification. Vehicle detection is conceded with background subtraction and also with blob tracking techniques. Speed of the vehicle is predictable through via start as well as stop lane sign then calibration constraint. Classification of vehicles depends upon the several features of the detected vehicles. Thus these features provide the input to SVM on behalf of

classification. A non-linear kernel used such as the classifier.

(Daigavane et al 2011) develop an application created on neural network aimed at vehicle detection and classification. This system detects then classifies the vehicles through success rate 90%. Vehicle are traced through blob tracking method then neural networks classify the vehicles on source of length and also height .Thus the cases where the system not capable to perform classification appropriately. When several vehicles passage together, by approximately the similar velocity, they tend to get gathered together as single vehicle. Similarly, the existence of shadow can reason the system to classify vehicles wrongly.

(Patel et al 2015) offered a novel Number Plate, segmentation method for offline images of the vehicle is used to process by algorithm. The system will not work attaching the camera with it and enchanting the real time image of the vehicle. Also, the typical processing time of 2s can be upgraded by code optimization as current code is not improved. The main restriction of the system is the determined speed at which the system can work it is somewhat difficult to differentiate moving and stationary obstacles. The First vehicle image is given as input to the in-lapping window-based method which converts the image into binary image based on the algorithm presented. And finally, the NP is segmented by using region-based row and column clustering methods. The two limitations of the system that it does not work: distance of more than 5m and also non-uniform NP. The system is similarly dependent on a high-resolution camera so if the image is taken through high resolution camera of more than 5 mega pixels at that time the algorithm might produce better accuracy.

(Shahare et al 2014) Authors discuss the various methods for observing moving obstacles methods in static also dynamic background in video. Authors also presented a technique for formulating low rank model aimed for detected obstacle. Authors also discussed a variety of techniques to detect moving obstacle in video frames and between the methods reviewed, the background subtraction method; the subtraction of colour and edge channels are done separately in advance for finding out the result. It is not robust in contrast to changes in illumination. It cannot detect non stationary background obstacle

such as swinging leaves, rain snow and shadow cast by moving obstacle. Additional, in this Authors suggested a single procedure of optimization which adds the obstacle detection and background learning which can be used to identify the moving obstacle correctly, the only improvement required time and accuracy attributes.

(Vaddi et al 2015) Authors proposed a prototype system for vehicle recognition. Thus the proposed method useful for several applications especially appropriate for Indian conditions. Obstacles of interest are recognized from images by the sift also with Harris corner features; they are further converted in the form of Histogram of words so termed as container of features. Classification is done by means of Support vector machine. By Considering the Indian conditions the recognition rates so far assured its usability within the context of a state-of-the-art. Thus the Experimental results shown for vehicle recognition process is correct up to restrained level as associated to base line result. They also done Analysis for the failure cases in the system also their solutions also offered. Thus the limitation on the optimization of the run time of the system and also development of recognition toughness for the case of infraclass variation and also be extended by improving database size and number of vehicle classes.

(Jin-Hua et al 2010) present a system that detects the types of vehicles to achieve the identification of vehicles. The extracted image heaving some noise due to that gives the misjudgement of the vehicle so considering this situation a Naive Bayesian classification is introduced. While characteristic parameter errors occurred, the probability values of different types of vehicles are calculated through Naive Bayesian classification, also compare the possibility standards.

(Toda et al 2016) Authors proposes algorithms for detecting moving obstacles using a moving stereo camera. The camera motion parameters be present predictable by with optical flow with a stereo camera. The optical flow taking place in the background is detached. The authors proposed technique has been calculated in experiments with two pedestrians in an indoor environment. The experiment was successful for detection two pedestrians.

(Xiang et al 2014) Proposed algorithms to eliminate the effect of illumination change besides shadow associated through the moving obstacle is a local intensity ratio model (LIRM) which remains robust toward illumination change. Authors discussed the distribution of local intensity ratio based on the illumination and shadow model. Thus, the moving obstacles be segmented without shadow with normalized local intensity ratio via Gaussian mixture model and then erosion was used to get the moving obstacles contours besides erase the scatter shadow patches and noises. After that, authors acquire the improved moving obstacles contours via a new contour enhancement technique, in which foreground ratio also spatial relation was considered. At last stage, a new method is used to fill foreground by holes. The Experimental outcomes prove that the proposed method acquire moving obstacles without cast shadow besides demonstrations excellent performance with existence many illuminations change conditions. Thus the limitation in particular scenes that foreground is like to background or else foreground is like to shadow, foreground may be easily detected as background. If illumination changes rigorously, such as changes in light of indoor scene then background might easily detect as foreground then the performance got drops.

(Xuehua et al (2015) offers an algorithm which can resolve the problems effectively by the enhanced Gaussian mixture model also Support vector machine. First of entirely, the paper presents an enhanced Gaussian mixture model which can effectively identify the moving obstacles also determine the problems of Gaussian mixture model sensitive toward light variations. Then the paper plans certain classifiers to distinguish the pedestrians also vehicles through the idea of the improved SVM. The experimental outcomes demonstrate that the technique has a high recognition rate also can satisfy the real-time intelligent transportation surveillance.

Iwaski et al (2015) Authors considered road traffic flow surveillance under various environmental conditions that reason for poor perceptibility of vehicles going on road. Authors considered thermal images taken through infrared thermal cameras on the way to recognition vehicle. Two techniques have been suggested. The first technique usages

pattern recognition used for windshields then their surroundings identify vehicles. The second technique usages tires' thermal energy reflection areas on road by means the detection targets.

Kuo et al (2016) Authors usages single camera with a cascade detection technique toward to implement the pedestrian collision warning system. Successful Recognition on the image processing technologies, Collisions can be expected since by means of the evidence of motion vector then the upcoming driving path. The situation can prompt the driver to attentiveness pedestrians also decrease the proportion of traffic accidents. The consequences of experiments demonstrate that this system can concurrently grip many pedestrians also appropriately compute the chance of collision.

Wang et al (2014) Authors presented novel moving obstacle detection method and they solely utilize the temporal information to generate the motion saliency map, in which the saliency of the moving obstacles as vehicle is also understandable moving characteristics is enhanced while the saliency of the background is reserved. The maximum entropy sum algorithm and the modified fuzzy growing method are applied to adaptively extract the ground truth of the moving obstacles. Author proposed method it does not require user interaction or parameter tuning as most of the preceding works ensure. Thus, the experimental results shown that the proposed schemes can detect moving obstacles with high accuracy and robustness. The proposed algorithm is based on the frame difference. While there are consecutive serried moving obstacles coming through the scene, they will exert negative interactions on each other. Drawback dealing with the interactions between the consecutive moving obstacles.

Chen et al (2011) examine the efficiency of state-of-the-art classification algorithms to classify road vehicles on behalf of an urban traffic monitoring system using a multi-shape descriptor. Thus the study is useful to monocular video acquired after stationary pole-mounted street side CCTV camera on a busy street. They are used toward classify the obstacles into four main vehicle classifications such as car, van, bus and also motorcycle. Image analysis intended for vehicle classification can exist usually categorised into three principal methods: Model-based classification, Feature based classification and Measurement based classification. Through the number of experiment the outcomes is

related toward the compare support vector machines (SVM) then random forests (RF) classifiers. 10- Fold cross validation used to estimate the performance of classification methods. The outcomes conclude that all methods accomplish a recognition rate above 95% on the dataset, with SVM constantly outperforming RF. A grouping of MBF also IPHOG features offer the unmatched performance of 99.78%.

Zhang et al (2013) Authors has presented algorithm will identify the moving target constructed on dynamic scenes more successfully. But still there are also limitations and shortcomings, in presented algorithm cannot eliminate such as leaves flutter noise and cannot identify multiple moving targets.

Zhuang et al (2016) Author has presents an extremely effective also accurate technique is aimed at predetermining obstacles now both in crowded and also in simple scenes. An adaptive technique for contextual detection remained firstly employed and the foreground sections of the scene more in, several sequences were obtained from a path with obstacles placed in the path of the moving cart mounted with a camera. The results indicate that targeted optical flow excludes background outliers thus making it more suitable for obstacle detection.

Czapla et al(2016) the optical flow taking place in the background is detached. The authors proposed technique has been calculated in experiments with two pedestrians in an indoor environment. The experiment was successful for detection two pedestrians. The only problem with false detection. Author has proposed method of vehicle detection applies conversion of traffic images into the binary form. Vehicle detection is carried out on the basis of analysis of the detection field state. It is determined by calculation of sums of the edge points within the detection field and comparison the obtained values to the present threshold values.it requires low computational cost. Vehicles of various classes are correct detected with the use of this method. The small number of required operations causes that the proposed method is fast and non-complex hence it can be useful in real-time processing and for hardware applications. The proposed method of vehicle detection is intended for measurements and control in road traffic systems. This method of vehicle detection also can be suitable for mobile systems.

2.6 Gap Analysis from the Literature Review

Thus, the researchers really did the great job, however the upcoming researchers are working as per standard but however the latest and future upcoming traffic scenario is different and is not complied thus the researcher issues is not equipped with existing system which has been identified as gap analysis as mentioned below:

1. Some papers had encountered is dataset problem. It's heavily biased towards one of the class and makes difficult to analyze on the classification.
2. Some paper had the highest number of misclassifications occurs between the car and van categories, where both size and shape features exhibit significant similarity.
3. Some paper had drawbacks of their work are the performance is delayed by the accuracy of the vehicle mask and the estimated bumper to road surface height.
4. Some paper fails to gives high accuracy for night time or bad weather conditions in their work.
5. Some paper fails to detect Vehicle if the distance between vehicle and the camera is less otherwise it faces lots of problems.
6. Some techniques had not covered Separation of Overlapped vehicles.
7. Some method is to detect road not vehicle so it does not handle overlapped vehicles.
8. Some method does not detect vehicles accurately in traffic jams and all-weather conditions.
9. As a major function of smart transportation in smart cities, vehicle model recognition plays an important role in intelligent transportation. Due to the difference among different vehicle models recognition datasets, the accuracy is less.
10. The current model for the vehicle detection absences in examining to challenges like low illumination, occlusions, partial and non-frontal views.
11. Shadow's basis the motion detection approaches fail in segmenting only the moving obstacles and make the upper levels such as obstacle classification using data mining techniques to achieve erroneous.

Chapter 3: Research Methodology

In this chapter, a computer vision approach for automated vehicle analytics and a machine learning/data mining approach are proposed and is the best one due to minimal infrastructure changes. No changes inside the vehicles are needed in computer vision methods. Automatic obstacle as vehicle analysis involves the detection and tracking of vehicle attributes such as color, model, etc. automatic car analysis automatic. Many applications can be developed through vehicle analysis, such Smart Parking Lots, Advanced Driver Assistance System, Visual Monitoring, Smart Towns, And Smart Transportation Systems,

3.1 Introduction Research Methodology

We intend to use computer vision algorithms with this dissertation to identify obstacles as a vehicle detection system, traffic management systems with smart functionalities. Some video cameras on highways and city will be utilized in computer vision and machine learning/data mining technologies to demonstrate concepts and test the classification algorithms. The overall aims of our research are to improve the collective understanding of the potential and limitations of the methods employed in our work by the research community. For this thesis, the objectives are as follows:

3.1.1 Objective of the Research Study

- For high-precision detection of obstacle as vehicles.
- Exploring various extraction methods for features and concentrating on GIST image features because it considers the image to be a single entity and calculates features based on the entire image. Other techniques of extraction of features attempt to find prominent interest points in the image and use these interest points to compute features.
- Transforms and speeds up two existing classification schemes and compares their performance with model recognition to create optimized dictionaries for scaled invariant feature.

- To develop a solid understanding of the characteristics of comparative models and the rationale behind performance differences, in particular, relevant to accuracy and speed.

3.1.2 Significance of the Research Study

The number of cars has increased every day since the industrial revolution. Traffic is one of the world's new challenges. People in cities waste most of their time in city traffic. It is important for all countries all around the globe to have a digital traffic system that works 24/7 and makes the tasks simple and efficient. The computerized system cannot therefore be operated without a precise vehicle detection system. The system clearly affects the economy, citizens' lives, the industry and so on. However, it is necessary to contribute to the topic until the algorithms are able to detect obstacles accurately, whether they are found in pipelines in pictures or videos.

3.1.3 Need of the Research Work

1.Many residential society administrations, tolls, business complexes and parking spaces in India lack an automated system for car parking and vehicle monitoring. Most of the commercial and residential places face an impending problem of illegal car/vehicle parking inside their premises.

2.The monitoring of crime prevention and the fight against terrorism are another useful adaptation of these techniques.

3. For both public safety and national security, this vision-based Automatic Vehicle Detection and Classification category is important.

4. CCTV installed on roads can detect different traffic accidents, such as speed traffic, misdirected route, collision and accidents.

5. The Maharashtra Accident Rate data show that the monsoon leads to visibility problems are more frequent.

6. The research proposed will assist police and security authorities in the rapid identification, monitoring and interception of suspect cars, allowing crime prevention and increasing civil security.

3.1.4 Benefit of Research Work

1. Computer vision and machine learning technology offers many advantages for many important ITS applications and components such as ADAS, AVS, Traffic Monitoring and

Activity Monitoring, Traffic Behavioral Analytics and Traffic Management. In view of increased safety concerns in ITS, vehicle identification and classification are of major interest in these applications.

2. In highly vulnerable areas such as public parking, security is a major concern (e.g., malls, stadiums, airports, etc.). The images of surveillance cameras may be of great assistance in identifying cars in certain colors, types, marks or models in these critical scenarios.

3. If the police look for a target vehicle of a given model, obstacle detection model system would save a significant amount of time, resources and work. Video/picture sensors could be shared by mobile police vehicles.

3.2 Research Approaches

Research approaches are research plans and procedures that extend from broad assumptions to detailed data collection, analysis and interpretation methods

3.2.1 Mixed Approach

First, we introduce the approach to mixed methods that support both quantitative and qualitative data. In particular, the integrated strategy was used as a subset of mixed methods to broaden the perspective on the subject. The main source of information for the study is quantitative online surveys. A survey to understand the 361-participant issue as a common driver by using non-probability judgement techniques of sampling. A total of 361 replies had received from frequent travelers.

3.2.1.1 Qualitative Research

Quantity, amount, intensity or frequency are meaning procedures and significances that are not thoroughly examined or measured (if measured). The social nature of a situation, the relationship between the researcher and what has been studied and the situation constraints of the research are stressed by qualitative researchers. Rather than processes, quantitative studies highlight the measurement and analysis of causal relationships between variables. The research should be free of value.

Main Characteristics of Quantitative Research

Control: This is the key element because it allows the scientist to identify the causes of his observations. In an attempt to answer certain questions, experiments are conducted. They try to figure out why something happens, what causes an event, or under what

conditions it happens. In order to provide clear answers to these questions, control is necessary. We have to eliminate the simultaneous influence of many variables to isolate the cause of an effect to answer questions in education and the social science. Controlled investigation is essential because the cause of an effect cannot be isolated without it.

Definition of operations: This means that the terms have to be defined through the measurement steps or operations. Any confusion about the meaning and communication must be eliminated by such a procedure. Consider the statement `Anxiety causes students to score poorly in tests. You could ask, "What is anxiety meant?" It only adds to the confusion to say that anxiety is tense or some other term. However, if anxiety refers to a score of an anxiety level, other people can realize what you mean by anxiety. The definition of an operation forces one to identify empirical terms or references. Ambiguity is therefore reduced to a minimum. Once again introversion may be defined as a score on a specific scale, hunger, and the social class as defined by occupation, which is so many hours since last fed.

Replication: The data obtained in an experiment must be reliable to be reproducible; that is, if the study is repeated the same result must be found. If observations cannot be repeated, our descriptions and explains are considered untrustworthy.

Hypothesis Testing: The systematic creation and empirical testing of a hypothesis.

TABLE 3.1 Qualitative obstacle comparison with approaches to vehicle detection.

Method	Growth to fine and boosted classifications	Dictionary based	Models based on deformable parts	Deep learning	Training architectures for image processing
Accuracy	++	+=	++	++	+=
Generality	==	++	+=	++	+=
Speed	++	+=	==	+=	+=
Advantages	It can work on small	Representation can be divided	Deformations and occlusions can be handled	Representation can	Multiple modules of a system are the

Drawbacks/ requirements	Features are defined in advance	All object instances cannot be detected	Small items cannot be detected	Extensive training sets for efficiency specialized hardware (GPU)	Too specialized for a specific environment may be the system obtained
Typical applications	Security, robotics	Backup, search.	pedestrian detection Transportation	Backup, search.	Health, robotics and human-computer interaction

Accuracy: ++, High; +=, Good; ==,Low.

Speed: ++, real-time (15 fps or more); +=, online (10–5 fps); ==, offline (5 fps or more).

Generality: ++ (+=), for many (some) classes of objects; ==, depend on characteristics designed for particular classes

TABLE 3.2 Summary of current directions and open issues

Solved problems	**Single-class**	**Single-view**	**Small distortions**	**Multi-scale**
Current directions	Multi-class (scalability and efficiency)	Multi-view/pose Multi-resolution	Deformable occlusions Object and background interlaced	Contextual information Temporal features
Open	Incremental learning	Relationship of objects	Detection of pixel level Objects of the background	Multi-modal

Song et al. (2012), Verschae and Ruiz-del-Solar (2012), and Erhan et al. (2014). Efficiency was addressed by using the same display for various object classes, for example, and by developing multi-class classifiers specifically designed for multiple class

detections. Dean et al. (2013) Present one of the few existing multi-class object detection works on a very large scale that considered 100,000 object classes.

3.2.1.2 Survey Questionaries for need of the Research study

Structured online questionnaires have been created to reach drivers in a short time and receive a broad overview of their issue faced in night driving on highway, under low - light. In addition to two open-ended questions, the survey included yes/no reply options. This technique has been used as it helps the respondent to examine the detailed view of a specific problem. It is very useful when we want to explore new issues and ideas in depth. These surveys helped me a lot in topic finalization. A survey to understand the 361-participant issue as a common driver by using non-probability judgement techniques of sampling. A total of 361 replies had received from frequent travelers.

A sample question sequence was:

Surveys: were conducted using Google online forms. Some 361 respondents filled these forms.

Time was from oct 2016 to Jan 2017.

Figure 3.1 Survey Form for Preparedness for Obstacle Detection Issues under heterogenous Climate.

Preparedness for Obstacle Detection Issues under heterogenous Climate.

* Required

Email Address *

Your answer

Name

Your answer

Phone *

Your answer

The exact distance between the vehicles cannot be judge due to low light in Mumbai – Pune Expressway

- ◯ yes
- ◯ no

Is the ghat section is really risky to drive, especially at night it ghat section of the expressway?

- ◯ yes
- ◯ no

Is there always a visibility issue during the monsoon which leads to accidents?

- ◯ During Day
- ◯ During Night
- ◯ Always

Is there always a visibility issue in tunnel which leads to accidents due some Street light are not working?

- ◯ yes
- ◯ no

Does the Foggy windows can make things difficult while driving it impairs the driver's visibility.

- ◯ Yes
- ◯ No

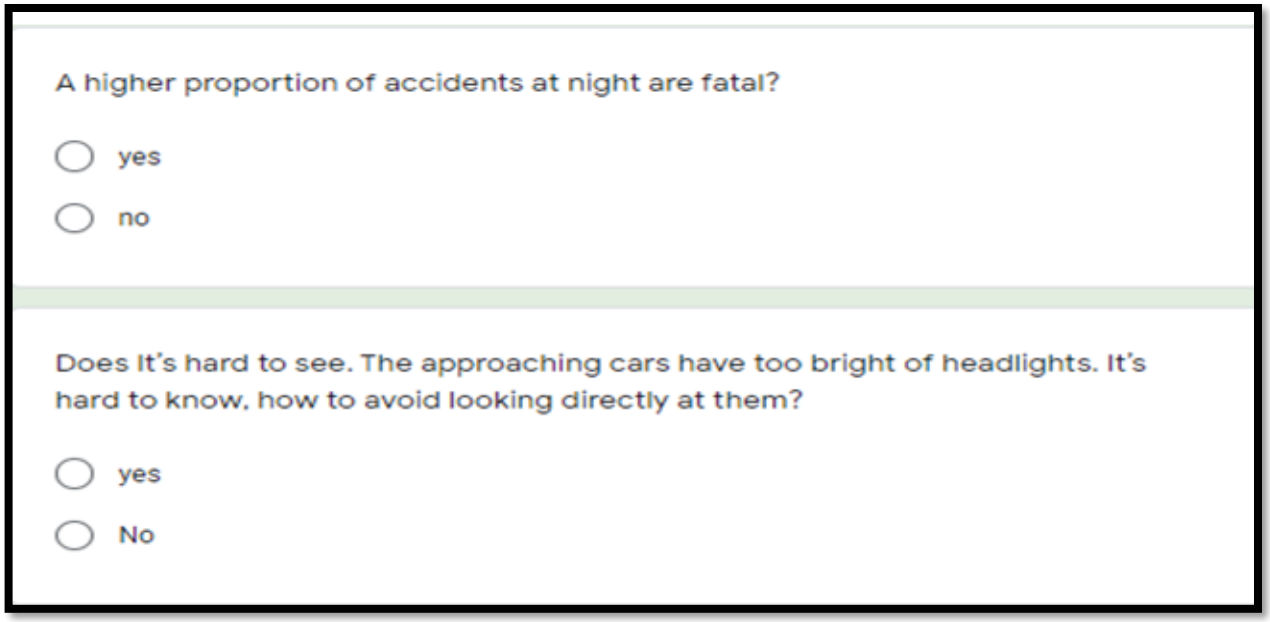
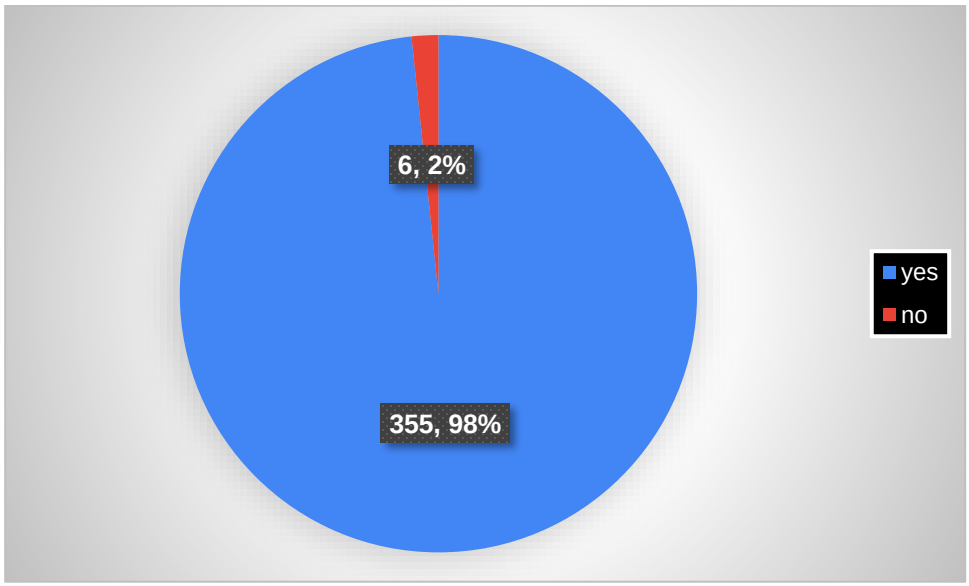


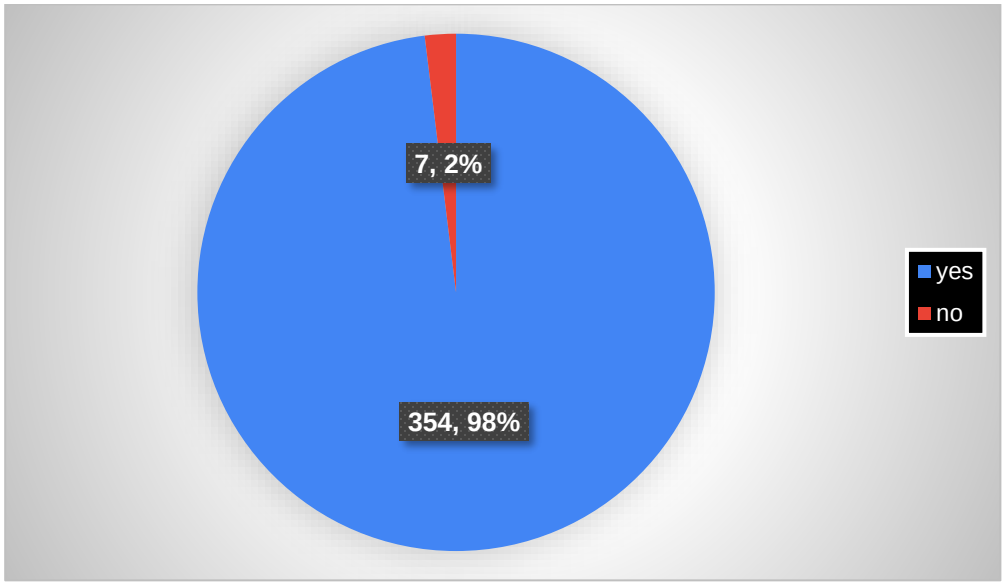
Figure. 3.2 Represents Road Weather, Visibility Issues in Express Highways causes for fatal accident.

Q1. The exact distance between the vehicles cannot be judge due to low light in Mumbai – Pune Expressway?



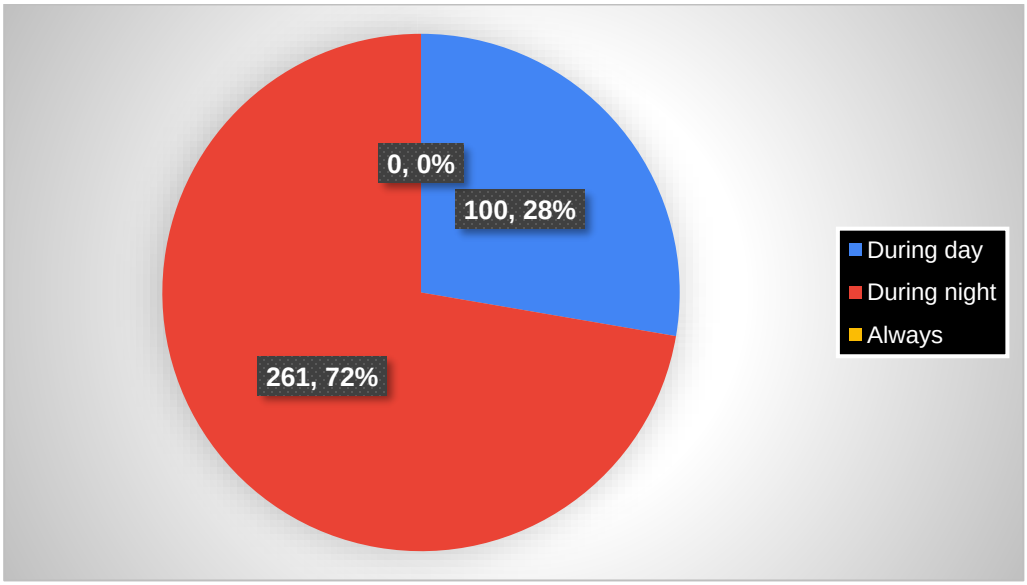
Total 98% agree, the vehicles cannot be judge due to low light in Mumbai – Pune Expressway and 2% disagree.

Q2. Is the ghat section is really risky to drive, especially at night it ghat section of the expressway?



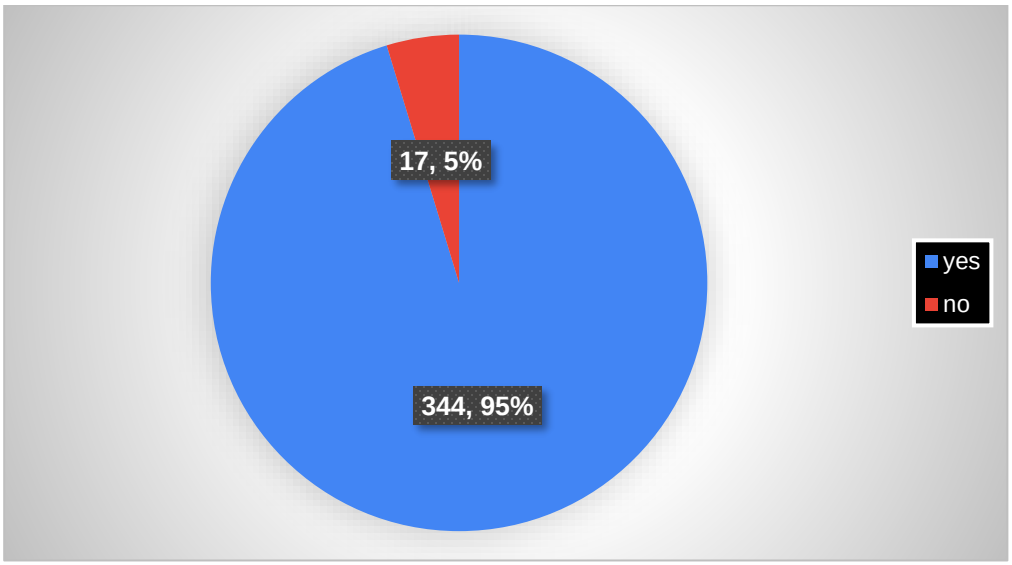
Total 98% agree ghat section is really risky to drive, especially at night it ghat section of the expressway.

Q3. Is there always a visibility issue during the monsoon which leads to accidents?



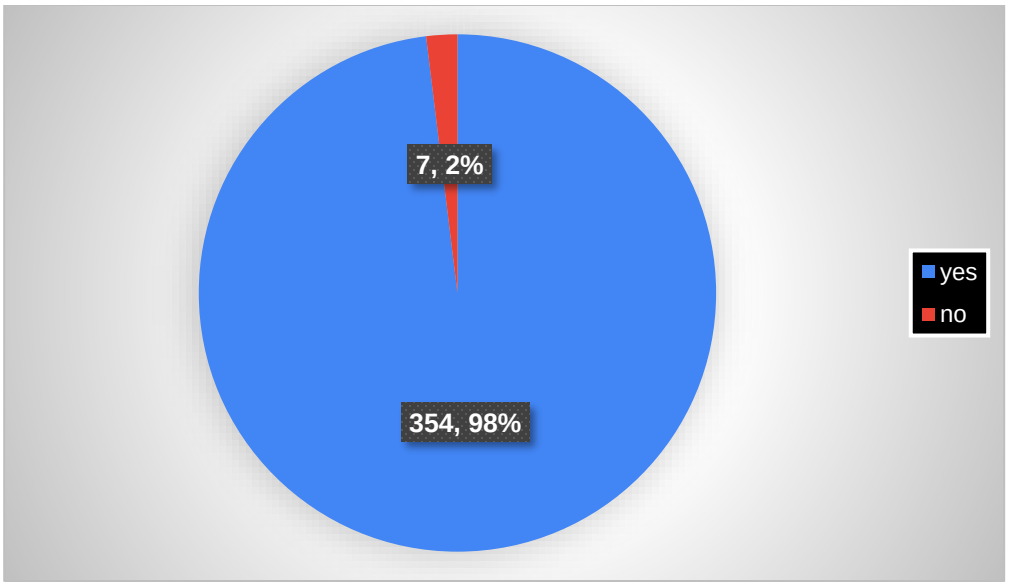
72% agree During day a visibility issue during the monsoon which leads to accidents.28% agree During Night a visibility issue during the monsoon which leads to accidents.

Q4. Is there always a visibility issue in tunnel which leads to accidents due some Street light are not working?



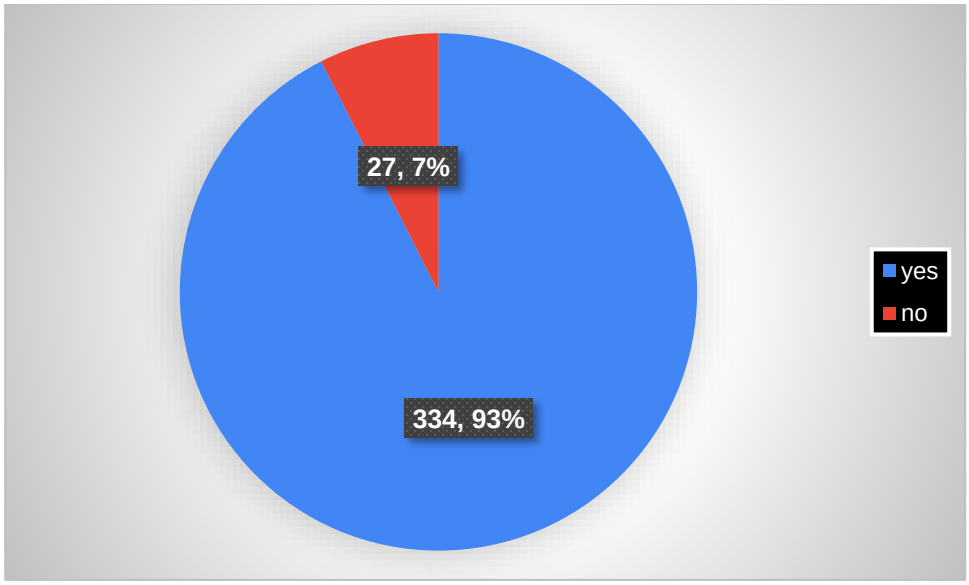
95% say yes about the visibility issue in tunnel which leads to accidents due some Street light are not working

Q5. Do the Foggy windows can make things difficult while driving it impairs the driver's visibility?



98% say yes, the Foggy windows can make things difficult while driving it impairs the driver's visibility.

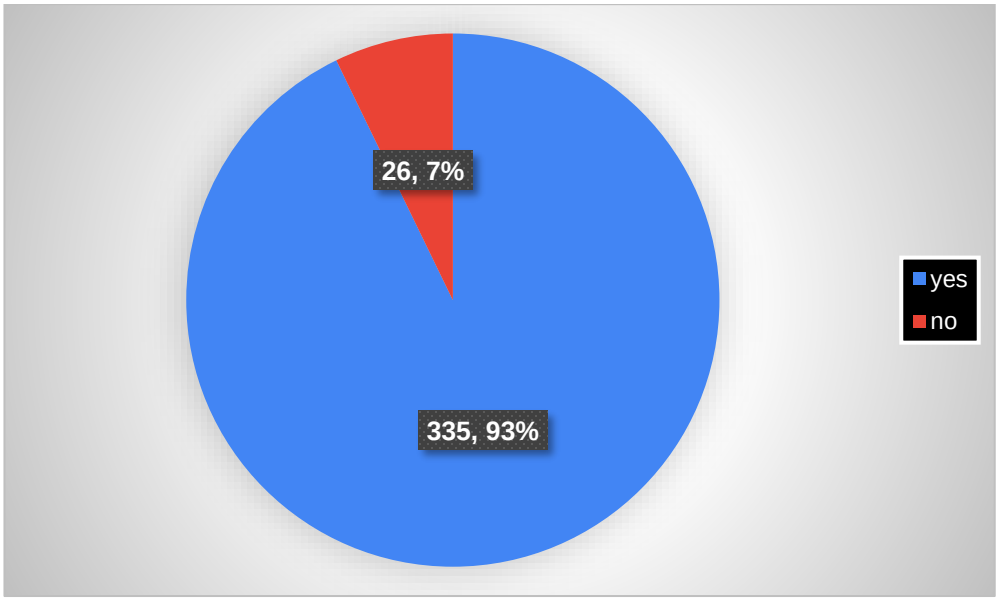
Q6.A higher proportion of accidents at night are fatal?



93% agree that a higher proportion of accidents at night are fatal.

Q7. Does It’s hard to see. The approaching cars have too bright of headlights. It’s hard to know,

how to avoid looking directly at them?



93% agree It’s hard to see the approaching cars have too bright of headlights.

3.2.1.3 A Survey on research work of Object as Obstacle Detection in 20 Years

Object detection aims to develop computer models and techniques that provide one of the most basic information required for applications of computer vision: Where are these objects? As an essential computer-vision problem, object detection is the foundation for numerous other computer-related tasks, such as the segmentation of the pictures (B. Hariharan et al. 2014, J. Dai et al. 2017) (A. Karpathy et al. 2015, Q. Wu et al. 2018), object-tracking (K. Kang et al. 2018), etc. The application view can be taken as an object detection into two research topics, "general object detection" and "detection applications," whereby the one seeks to explore the methods of detecting different types of objects within a unified framework for simulating human vision and cognition. The rapid development of deep learning techniques (Y. LeCun et al 2015) has led to notable

breakthroughs and the development of new blood to an unprecedented point of attention In recent years. Object detection has now been widely employed in many applications in the real world, including automotive drive, robot vision, video monitoring, etc.

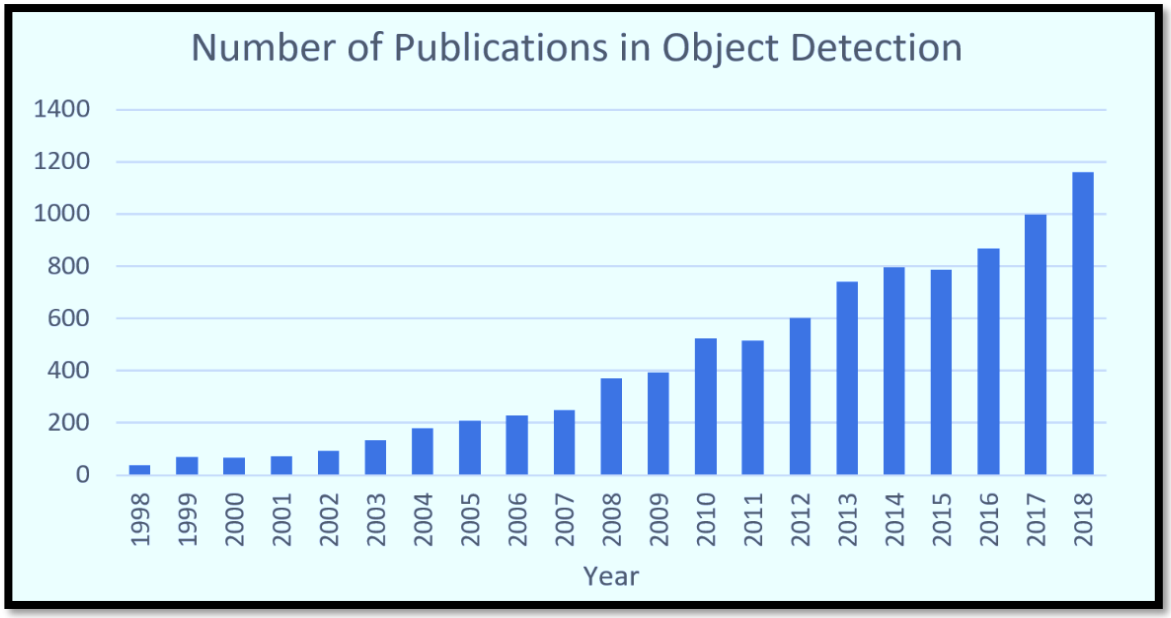


Figure. 3.3. The growing number of object detection publications between 1998 and 2018. (Google Advanced Search Scientist Data: "object detection" AND "object detection.") Difference from other associated assessments In recent years, several reviews have been published on general object detection. The principal difference between current issues and the previous reviews is as follows:

A thorough review in the light of technical developments:

1. This reviews more than 400 papers covering over one quarter of a century in the evolutionary history of object detection (from the 1990s to 2019). The majority of previous reviews focus only on a short period of history or on a number of particular detection tasks without regard for the technical developments throughout their lives. Not only can readers build the entire hierarchy of knowledge by standing on the highway of history but they can also find future directions in this fast-developing field.

2. A comprehensive study of key technologies and of the state of the art of recent years: The state-of-the-art object detection systems were integrated after a number of years of development, using a number of techniques including 'multiscale detection,' 'hard, negative mining,' and 'boundary box regression.' However there has been a lack of a fundamental assessment of past reviews that allow readers to gain a better understanding of the nature of these sophisticated techniques. "What are the advantages and disadvantages of each group?" This article analyses these concerns in detail for readers.

3. A complete study of detection speed up methods: The detection of objects has been an extremely important but difficult task for many years. An extensive review of acceleration techniques at several levels, including a "detection pipeline" (e.g., cascaded detection, shared computing map feature), a "backbone detection" (e.g., network compression, lightweight network design) and a "numeric computation," over a period of twenty years (e.g., integral image, vector quantization). In previous reviews this topic is seldom covered.

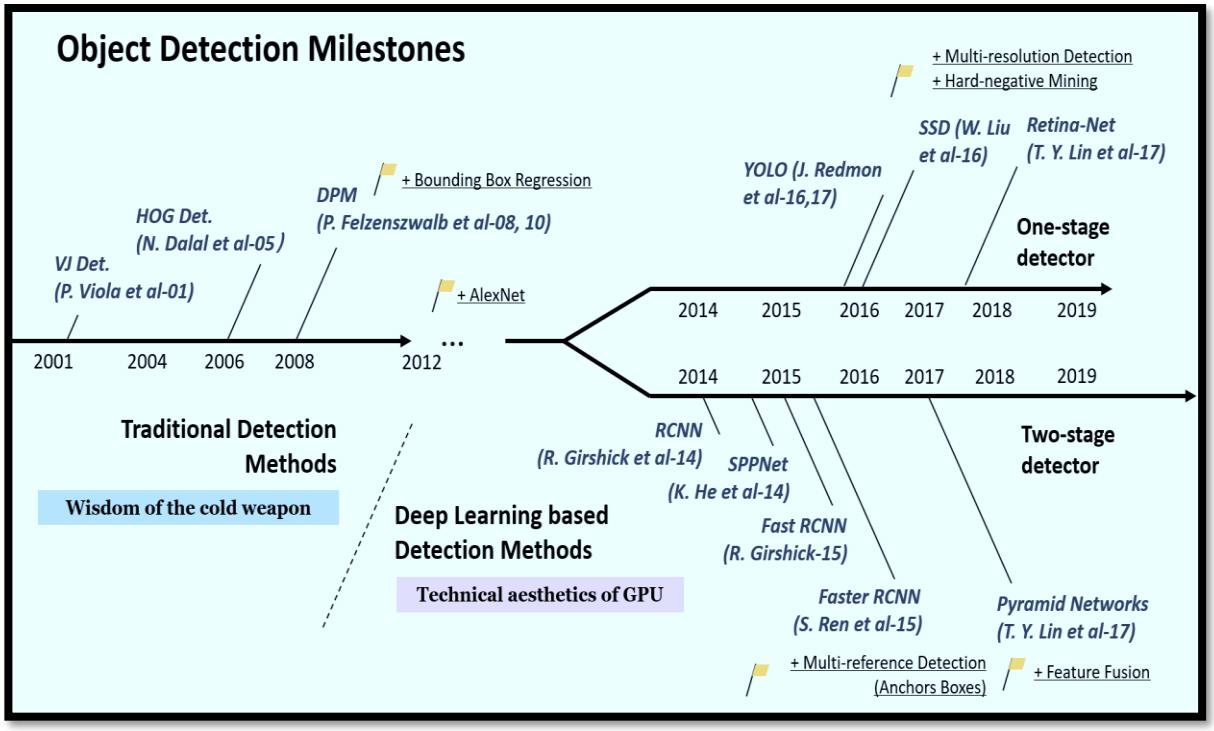
Object Detection Problems and Challenges

In fact, this question is not easy to answer, even if people always ask, "What are the difficulties and challenges in object detection?" and may even be over-generalized. Due to completely different targets and constraints in the different detection tasks, their difficulties can vary. Besides some common problems for 3 other visual tasks such as objects in different perspectives, lighting and intra-class variations, the challenges in detection of objects are but not limited to the following: rotation of objects and the change in their scale (e.g., small objects) and precise location of objects, dense and occult detection of objects and acceleration of detection.

3.2.1.3.1 Object Detection Route Map

It has been widely accepted in the past two decades for the advancement in object detection to be seen in two historical periods: the 'traditional object detection period before 2014' (before 2014) and the 'deep learning detection period after 2014' (see Figure. 3.4).

Figure.3.4. Demonstrates the Object Detection Route Map & Milestone detectors



3.2.1.3.2 Background points: Traditional detectors.

If we think of the detection of objects today as a technical aesthetics under the power of deep learning, then we would see "the wisdom of a cold weapon when we turned around 20 years ago." Most early object detection algorithms have been manufactured by hand. Given the failure to represent images effectively at that time, people had no choice but to design sophisticated representations of features and to speed up the use of limited computing resources.

• Detectors of Viola Jones

18 years ago, for the first time P. Viola and M. Jones were able to detect human faces in real time without restrictions [P. Viola and M. Jones et al 2001, P. Viola and M. Jones et al 2004]. When running on a 700MHz Pentium III CPU, the sensor was in comparable detection accuracy tens or even hundreds of times faster than other Algorithms in its day. In memory of their important contributions, in this context, a detection algorithm referred to as the "Viola-Jones (VJ) detector" was given by the authors' names.

The VJ detector follows a very simple way of detection, namely sliding glasses: it is necessary to go through all possible scales in an image to see if there is a human face to

any window. Even though the calculation behind it appears to be very easy, it was far beyond the power of your computer. By using three major techniques: 'integral image,' 'feature selection,' and 'detection cascades,' the VJ detector has dramatically improved its detection speed.

1) Integral image: The integrated image is a computer method for accelerating the filtering process or convolution. The Haar Wavelet is used as the representation feature of a picture in the VJ detector, just as other object detection algorithms in its time (C. P. Papageorgiou et al 1988, C. Papageorgiou et al 2000, A. Mohan et al 2001). The integral picture makes every window's computer complexity independent of its window size in the VJ detector.

2) Feature selection: The authors used Adaboost algorithm [Y, Freund et al 1999] to select smaller functions, mostly useful to detect face effects from a large number of random features in the pile of characteristics instead of the manually selected hair basis filters (about 180k-dimensional). 3) Cascade detection: In the VJ detector, a multi-stage detection paradigm was introduced to decrease the overhead by spending less on background windows and more on facial targets.

• Detector HOG

In 2005, N. Dalal and B. Triggs originally proposed the Histogram of Oriented Gradients descriptor (HOG) (N. Dalal and B. Triggs et al 2005). The S. Belongie et al 2002 Scope and form contexts (D. G. Lowe et al, 1999) can be considered as an important improvement in the scale-invariant feature of transform. The descriptor is designed to be computed on a dense grid of uniformly spaced cell and to use superimposed local contrast (on blocks) to balance the feature invariance (including translation, scale, illumination) with nonlinearity (on discrimination between different categories of objects). HOG can be used for the detection of a certain variety .The problem of pedestrian detection was primarily motivated by the object classes. The HOG detector rescues input images for multiple times to detect objects of different sizes while maintaining the size of the screen. A large number of object detectors have been built over time in the detector of HOG (P. Felzenszwalb et al 2008, P.Felzenszwalb et al 2010, T. Malisiewicz, A. Gupta, and A. A. Efros et al 2011) And over many years a wide range of computer vision apps.

• Models based on deformable parts(MDP)

Models based on deformable parts The peaks of traditional object detection methods were the VOC-07, -08, and -09 winners. In 2008, P. elzenszwalb proposed MDP as an extension to the HOG detector and then R. Girshick [R.B. Girshick, P.F. Felzenszwalb and D.A. Mcallester et al 2011, R. B. Girshick et al 2008] made a variety of improvements. The MDP follows the philosophy that detects "divide and conquer" in which education can simply be regarded as learning a correct way to decompose an object and the inference can be viewed as the detection ensemble of different object parts.

For example, the problem of detecting a “car” can be considered as the detection of its window, body, and wheels. This part of the work, a.k.a. “star-model”, was completed by P. Felzenszwalb et al. Later on, R. Girshick has further extended the star-model to the “mixture models” [R. B. Girshick, P. F. Felzenszwalb, and D. A. Mcallester et al 2011, R. B. Girshick et al 2008] to deal with the objects in the real world under more significant variations. A typical MDP detector consists of a root-filter and a number of part-filters. In MDP weakly monitored learning methods are developed, where all part filter configurations can automatically be learnt as latency variable rather than manually specifying the parts filter's configurations (e.g., size and position). This process has also been formulated by R. Girshick as a special case for multinational learning [S. Andrews et al 2003] and several other important techniques are also employed for better detection accuracy such as "hard negative mining", "bounding box regression" and "context priming." In order to speed up detection, Girshick has developed a "compile" detection model technique to a faster one, implementing a waterfall architecture that has accelerated 10 times without any loss of accuracy (P. F. Felzenszwalb et al 2010, R. B. Girshick et al 2012). While the accuracy of the detection by the object detectors of the present day has far surpassed MDP, many still depend on their valuable insights such as mixture models, harsh negative mining, binding box regression and more.

3.2.1.2.3 Milestones: CNN based Two-stage Detectors

After 2010, the object detection reaches a plateau with hand-crafted characteristics saturated. "During 2010-2012, progress was slow and small gains were accomplished by setup systems and a small number of successful methods" (R. Girshick et al 2012). In 2012, the world has revived neural networks (A. Krizhevsky et al 2012). As a deep network can learn robust and high-level imaging properties, the question naturally arises

as to if we can detect objects. R. Girshick et al. led the blockage to break in 2014, with the proposal to detect objects by CNN (RCNN) regions (R. Girshick et al 2014, R. Girshick et al 2016). Since that time, object detection has developed at a speed without parallel. Object detection can be divided into two deep learning types: two stage detection and one stage detection whereas the previous frames detection can be divided into a "fine-turning process."

• RCNN

The idea behind RCNN is simple: selective searching begins with a selection of object proposals (K. E. Van de Sande et al 2011). Each proposal is then resized to a 5th size image and fed into an ImageNet trained CNN model. Finally, linear SVM classifiers are used in the prediction of the presence of an object in each region and the recognition of object categories. RCNN has significantly improved performance of VOC07 from 33.7% (MDP-v5[43]) to 58.5% on average. Although the RCNN is progressing significantly, its drawbacks are clear: the redundant calculation of functions on many overlaps (over two thousand boxes from an image) achieves a very slow detection speed (14s per image with GPU). SPPNet was proposed and tackled later that year (K. He, X. Zhang, S. Ren, J. Sunet et al 2014).

• SPPNet

In 2014, K. He et al. proposed Space Pyramid Pooling Network (SPPNet) (K. He, X. Zhang, S. Ren, and J. Sunet et al 2014). A 224x224 image for AlexNet is needed for previous CNN models, such as (A. Krizhevsky et al 2012). SPPNet's main contribution is the deployment of an SPP layer that allows a CNN to generate a fixed image size/region of interest representation regardless of the image size. When you can calculate the function maps of the whole image by using SPPNet to detect objects, arbitrary length displays may be generated to train detectors in order not to calculate the convergence features repeatedly. SPPNet is more than 20 times faster than R-CNN, without sacrificing detection accuracy (VOC07 mAP=59.2%). Even if SPPNet has improved its detection speed efficiently, there are still certain disadvantages: firstly, the training takes several steps, and secondly, SPPNet can only finalise its fully connected layers without any prior layers. In the following year, Fast RCNN(R. Girshick et al. 2015) was later proposed and decided.

• Faster RCNN

S. Ren et al. suggested Faster RCNN detector in 2015, shortly following Fast RCNN (Ren et al 2015). Fast RCNN (COCO mAP@.5 = 42.7%), COCO mAP@[.5,.95] (= 21.7%), VOC07 mAP =73.2%), VOC12 =70.4% (17 fps), and ZFNet (Very close to real time) (M. D. Zeiler and R. Fergus et al 2015]). Faster-main RCNN's contribution is the creation of a regional proposal network (RPN). RCNN and Faster RCNN are gradually integrated in a single end-to-end learning environment in the most different blocks in a system of object detection like propositional detection, function extraction, bounding box regression, etc. While Faster RCNN interrupts the RCNN bottleneck, the following detection step continues with computational redundancies. A number of improvements were then proposed, such as the RFCN (J. Dai, Y. Li, K. He, and J. Sun et al. 2016) and the RCNN Light Head (Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun et al 2017)

• Feature Pyramid Networks

In T.-Y. Lin et al, based on Faster RCNN, 2017, featured Fired Pyramid Networks were proposed in 2017 (T.-Y. Lin, R.-B. Girshick, K. He, B. Hariharan, 'and S.J. Belongie et al. 2017). Deepest FPN-based learning Only top-level network detectors perform detection. While the functionalities in deeper CNN layers benefit from category recognition, they do not allow for the identification of objects. To that purpose, FPN has developed a top-down architecture with side connections for the construction of high-level semantism on all scales. As a CNN naturally constitutes an advanced pyramid, the FPN has taken significant steps to detect objects on a wide variety of scales. The FPN is a modern single model detection system, which produces an MSCOCO data set with Faster R-CNN (COCO mAP@.5=59.1 percent, COCO mAP@[.5,.96]=36.2 percent) that is bell-free, state-of-The-art. FPN is now an integral part of many recent detectors.

3.2.1.2.4 Milestones: CNN based One-stage Detectors

• You Only Look Once (YOLO)

In 2015, R. Joseph et al. proposed YOLO. It was the first detector of a one-stage profound education. A fast version of YOLO with VOC07 mAP=52.7% runs at 155fps, while the upgrade is VOC07 mAP=63.4% and VOC12 mAP= 57.9 percent at a speeding of 45fps. YOLO is "You just look once" abbreviation. The authors have completely abandoned

their previous "proposal detection + verification" detection paradigm. Rather, the whole philosophy is entirely different: to apply a single neural network in the whole picture. This network divides the picture into regions and simultaneously predicts border boxes and probabilities for each area. Subsequent improvements were made by R. Joseph on the basis of YOLO and the v2 and v3 editions proposed (J. Redmon

A Farhadi et al 2017 & 2018), which improves detection precision while maintaining a high speed of detection. YOLO suffered a drop in localization accuracy compared with two-stage detectors, mainly for certain small objects, despite its great improvement in detection speed. More attention has also been given to the future of YOLO (J. Redmon and A. Farhadi et al 2017 & 2018) and SSD (W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y.Fu, and A. C. Berg et al 2016).

• Single Shot MultiBox Detector (SSD)

SSD (W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y.Fu, and A. C. Berg et al 2016) was proposed by W. Liu et al. in 2016. This detector was the second phase of deep learning. The main contribution of SSD is the introduction of multi-reference and multi-resolution detection techniques, which improves detection accuracy significantly for a single-stage sensor, particularly for smaller objects. SSD is both advantageous for speed and accuracy (VOC07 mAP = 76.8%, VOC12 mAP = 74.9%, COCO mAP = 46.5%, MAP [.5.95] = 26.8%, quick version 59 fps). The main difference between SSD and all previous detectors is that the first scales objects on various layers of the network are detected, while the latter scales only run detection on their top layers.

• RetinaNet

The one-stage detectors track the accuracy of two-stage detectors for years, despite their high speed and simplicity. T.-Y. Lin et al. found why and suggested RetinaNet in 2017. (T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar 2018). It was argued that the central cause is the extreme imbalance in the foreground class during training of dense detectors. In order to achieve this, RetinaNet has introduced a new loss function called "focal loss" by reshaping the standard cross-entropy loss so that during the training the detector will be more focused on hard Evaluations of the errors. One phase detectors can achieve focal losses, while keeping the high detection speed, which is comparable to two-stage detectors. (COCO mAP@.5=59.1%, mAP@[.5,.95]=39.1%).

3.2.1.2.5. Object Detection Datasets and Metrics

The development of advanced computer vision algorithms requires greater data setting with less bias. A number of renowned datasets, including PASCALVOC Challenges datasets (Everingham, L. Van Gool, C. K. Williams, J.Winn, A. Zisserman et al 2010, M. Everingham, S. A. eslami, L. Van Gool, C. K. Williams, J.Winn and A. Zisserman et al. 2015, VOC 2007, VOC2012) have been published for object recognition in the last 10 years (for example, VOC2007, VOC2012, VOC2012, ImageNET, VOC2014).

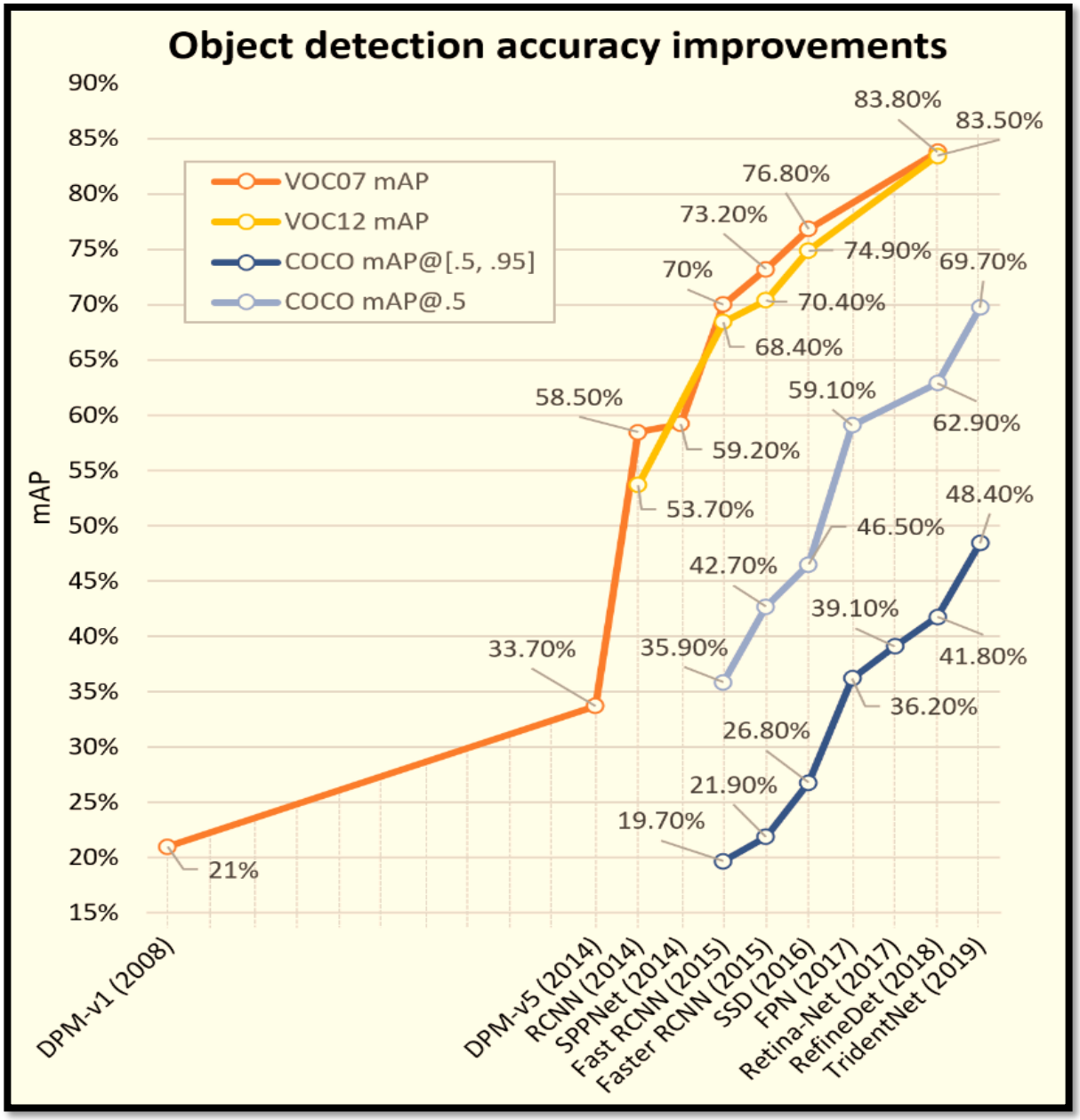
• Pascal VOC

One of the most important competitions in the early computer vision community was the PASCAL Visual Object Classes (VOC) Challenges1 (2005-2012) (M. Everingham et al 2010). PASCAL VOC has many tasks, including classification of the image, detection of objects, semantic segmentation and action detection. Two Pascal-VOC versions are mainly used when detecting objects: the VOC07 and the VOC12, which consist of 5k tr. images + 12k objects annotated and the latter consists of 11k tr. images, + 27k objects annotated. Twenty classes of life-common objects (Person, Cat, Bird, Cow, Horse or Sheep), Vehicle: aircraft, bicycle, boat, bus, motorbike, railway; Indoor: bottle, chair, table, couch/water-tops, tv, monitor) are annotated in these two datasets. In recent years with the release of some larger datasets such as ILSVRC and MS-COCO, the VOC has gradually become the test bench for the newest detectors.

• ILSVRC

The Visual Recognition Challenge IMGENet (ILSVRC)2 has pushed the state of the art in the detection of generic objects forward. Every year from 2010 to 2017, ILSVRC is organized. It includes a challenge to detect images via ImageNet (J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei et al 2009). The ILSVRC dataset includes 200 visual object classes. Two orders greater than VOC are the number of image/object instances. ILSVRC-14 includes, for example, 517k and 534k objects.

Figure.3.5. Demonstrate the accuracy improvements of object detection on VOC07, VOC12 and MS-COCO datasets. Detectors.



• MS-COCO

The most difficult object detection dataset available today is, of course, the MS-COCO3(T.–Y. Lin et al 2014). Since 2015, the annual MS-COCO dataset competition has taken place. It has less categories of objects than ILSVRC but more instances of objects. MS-COCO-17, for example, contains 164K images, 897K of 80 categories of annotated objects. Compared to MS-VOC/ILSVRC, COCO's the major advances are that each item is further marked with a per-instance segmentation to support accurate

localization, in addition to the bordering box annotations. MS-COCO also contains more small objects (whose area is smaller than 1% of the image) than VOC and ILSVRC, as well as more densely positioned objects. The distribution of objects in MSCO is more similar to the real world with all these features. MS-COCO has become the standard defect for object sensing, just like ImageNet in its days.

• Open Images

In 2018, the open image detection (OID) challenge4 (I. Krasin et al, 2017) was introduced, but on an unprecedented scale, following MS-COCO. Open images have two tasks: (1) the standard object detection and (2) the detection of visual connections, which specifically detects paired objects. The data set includes 1.910k images and 15.440k bounding boxes on 600 category objects for the object detection work. for object detection

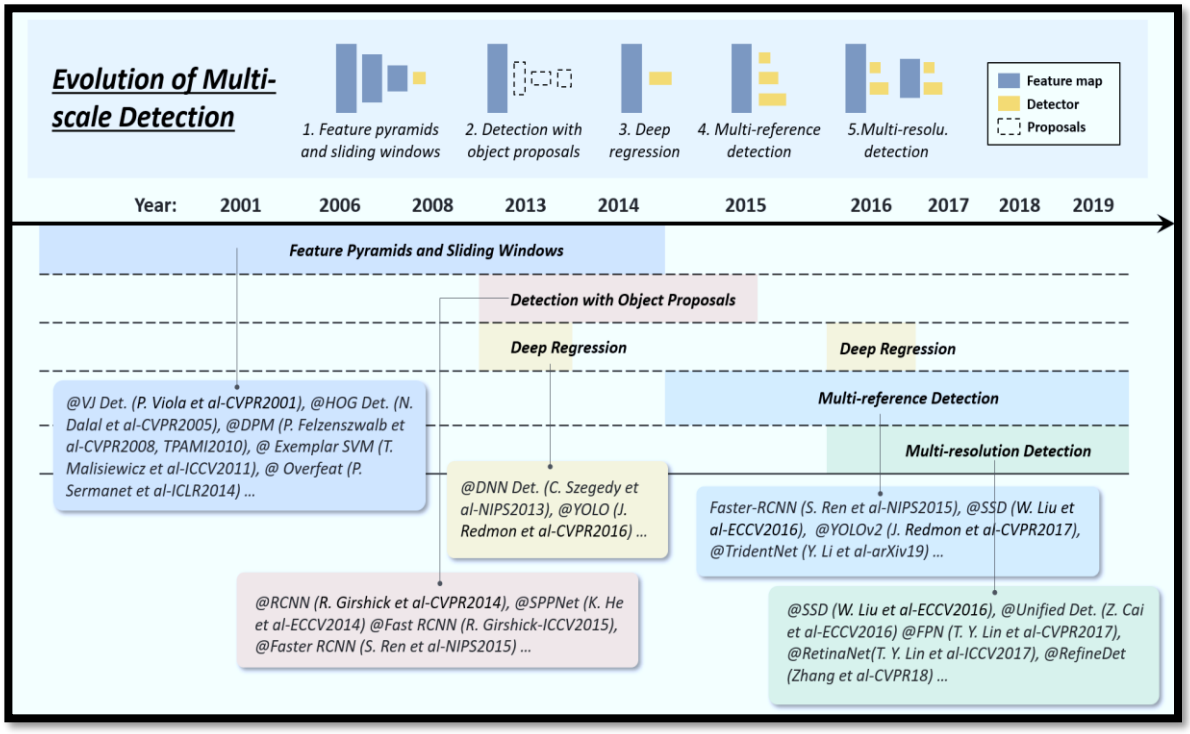


Figure.3.6. Detector of objects from 2001 to 2019 in multi-scale evolution:1) feature pyramids and sliding windows, 2) detection with object proposals, 3) deep regression, 4) multi-reference detection, and 5) multi-resolution detection.

3.2.1.3.5.1. Procedural Evolution of Multi-Scale Detection

One of the most important technical challenges to object detection is the multi-scale detection of objects with "different sizes" and "varying aspect ratites." Multi-scale detection of psychological

and sliding windows in the past 20 years has been through several periods: "Function-reference pyramids (Before2014);" "Article Detection (After2016)"" (Figure3.6) and "Multi-resolution Detection (After2016). The detection process is now undergoing multi-scale regression (2013-2016).

• Feature pyramids + sliding windows (before 2014)

As the VJ detector gained computer power, researchers began to pay more attention to an intuitive detective method with the creation of "feature pyramid + sliding windows." A series of milestone detectors based on this detection pattern were built between 2004 and 2014.including the HOG detector, MDP, and even the Overfeat detector (P. Sermanet et al 2013) of the deep learning era (winner of ILSVRC- 13 localization task). Models for early detection, such as the VJ and the HOG detector, were designed specifically to detect "fixed aspect ratio" objects (e.g., faces and upright pathways) with a simple construction of a feature pyramid and slides, fastened size detection window. At that time, no consideration was given to detecting "different aspect ratios. " R. Girshick et al. began to find better solutions outside the feature pyramid in order to detect objects with a more complex appearance, like those in PASCAL VOC. The "mixture model" (P. F. Felzenszwalb et al.2010) was at that time one of the best solutions for detecting objects with different aspect ratios by training several models. Besides, the exemplary detection provided a new solution for each object instance of the training set, training individual models (exemplary). Since objects are increasingly diversified in modern datasets (e.g. MS-COCO), the mixture model or exemplary based methods will necessarily lead to diverse detection models. A question then arises of course: is the approach to detect objects of different aspect ratios unified in a multi-scale? This question was answered by the introduction of "object proposals" to be submitted.

• Detection with object proposals (2010-2015)

Object proposals refer to the class-agnostic box group of applicants that may contain any objects. It was used for the first time in 2010. (B. Alexe et al 2010). Object detection helps to avoid a full search of an image through the sliding window. The following three requirements should be met by an object proposal detection algorithm:1(2) high localization accuracy, and (3) improving precision and reducing processing time based on

the first two requirements. Modern methods can be classified into three categories for the detection of proposals: 1) segmenting grouping approaches (K.E. Van de Sande et al 2011; J. R. Uijlings et al 2013; J. Carreira et al 2010, p. Arbel et al 2014), 2) window scoring approaches [B. Alexe et al 2010.] M.-M. Cheng et al 2014. For a complete review of these methods please refer to the following papers The early proposal methods for detecting the visual leakage followed a philosophy for bottom up detection and were profoundly affected. Subsequently, researchers moved to low-level vision and more sophisticated craft to enhance the location of candidate boxes. Follower, learning methods began to show more benefits in this problem after 2014 with the popularity of deep CNN in visual recognition. From the bottom-up vision to the "overfit to a specific set of object classes," object proposal detection has evolved and the distinction between detectors and proposal generators has been blurred (N. Chavali et al 2016). Since the "object proposals" evolved the detection of sliding windows and quickly dominated deep-sea learning detectors, several researchers began asking questions in 2014-2015: what is the primary role of object detection proposals? Is it designed to improve accuracy or just to speed up detection? To answer this question, some researchers (K. Lenc and A. Vedaldiet al 2015) have attempted to weaken the role of the proposals or simply to detect CNN-features through sliding windows, but nbody of them has got satisfactory results. The proposal detection quickly disappeared following the rise in one-stage detectors and techniques of "deep regression" (to be introduced).

• Deep regression (2013-2016)

In recent years, the way people handle multi-scale detection is increasingly straightforward and brute, as computing power for GPU has been increased. It is very simple to apply profound regression to solve multi-level problems, which means that the co-ordinates of a bounding box can be predicted directly based on the deep study features. The advantage is that it is simple and easy to implement, while the disadvantage is that the location may not be sufficiently accurate especially for certain small objects. The latter problem has been solved by "multi-reference detection" (to be implemented).

• Multi-reference/-resolution detection (after 2015)

The most popular framework for multi-scale object detection is multi-reference detection. The main objective is to pre-define a set of reference boxes (anchor boxes) with different sizes and aspect ratios at different locations in a picture. Multisolution detection, i.e., by identifying objects of various scales at various network layers is another popular technique in the past two years. As a CNN naturally forms a pyramid during its future spread, Larger objects in depths and smaller objects in shallow layers are easier to detect. Multi-reference and multi-resolution detection in state-of-the-art object detection systems have now become two main building blocks.

3.2.1.3.5.2. Procedural Evolution of Context Priming

Visual objects are usually integrated with the surrounding environments in a typical context. In order to promote visual perception and cognition, our brain takes advantage of the links between objects and environments [SK Divvala et al 2010]. Context priming was used to improve detection for a long time. In its evolutionary history, three common approaches exist: 1) local context detection, 2) global context detection, and 3) interactive context.

Figure 3.7. Demonstrates the Evolution of bounding box regression techniques in object detection from 2001 to 2019

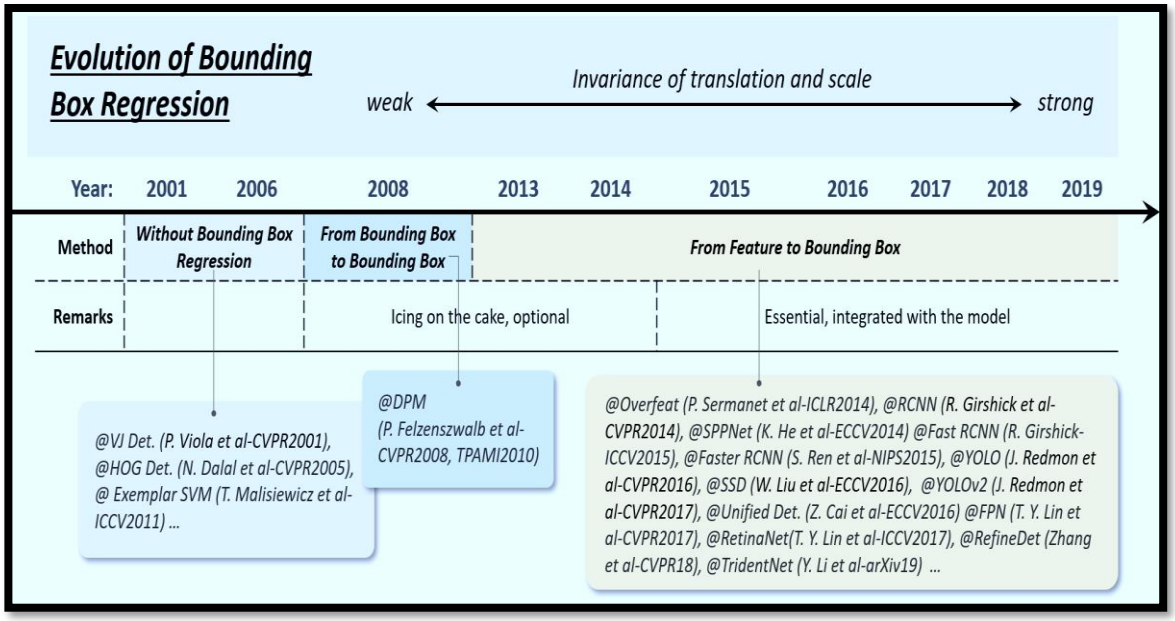
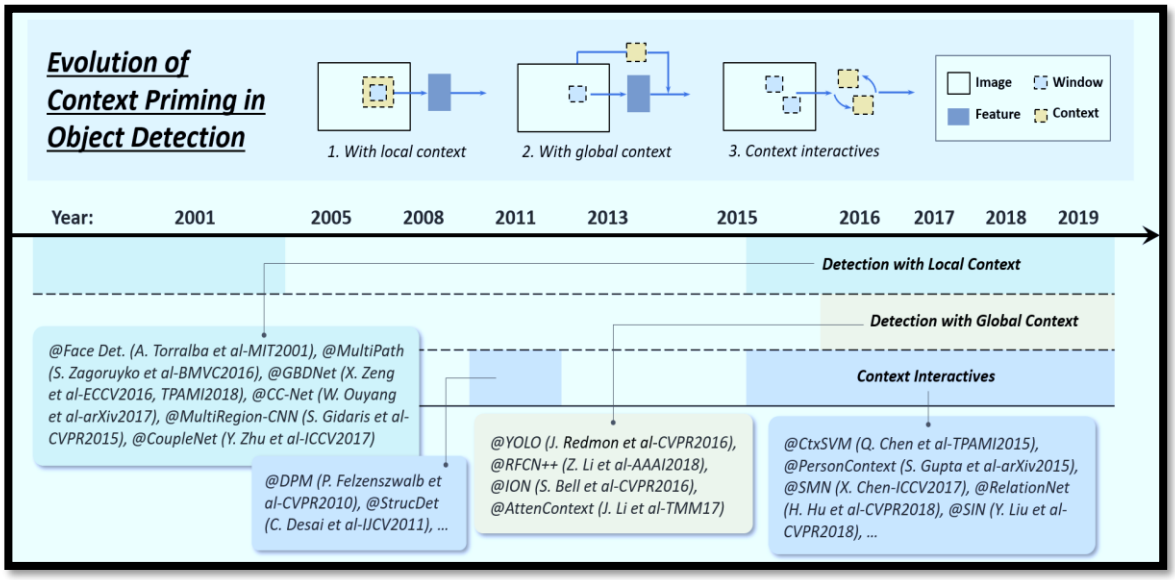


Figure.3.8. Demonstrates the Evolution of context priming in object detection from 2001 to 2019: 1) detection with local context, 2) detection with global context, 3) detection with context interactives.



The visual information in the area surrounding the object to be detected is the local context. Local contexts help improve object detection have long been acknowledged. In the early 2000s, Torralba (A. Torralba et al.2011) found it considerably enhances the performance of facial detection by including local contextual areas such as the facial border contour. Dalal and Triggs also found it is better to accurately detect pedestrians if a small amount of background information is incorporated. Recent detectors based on learning can also be improved with local contexts by simply broadening the receptive field of the networks or the size of subject proposals.

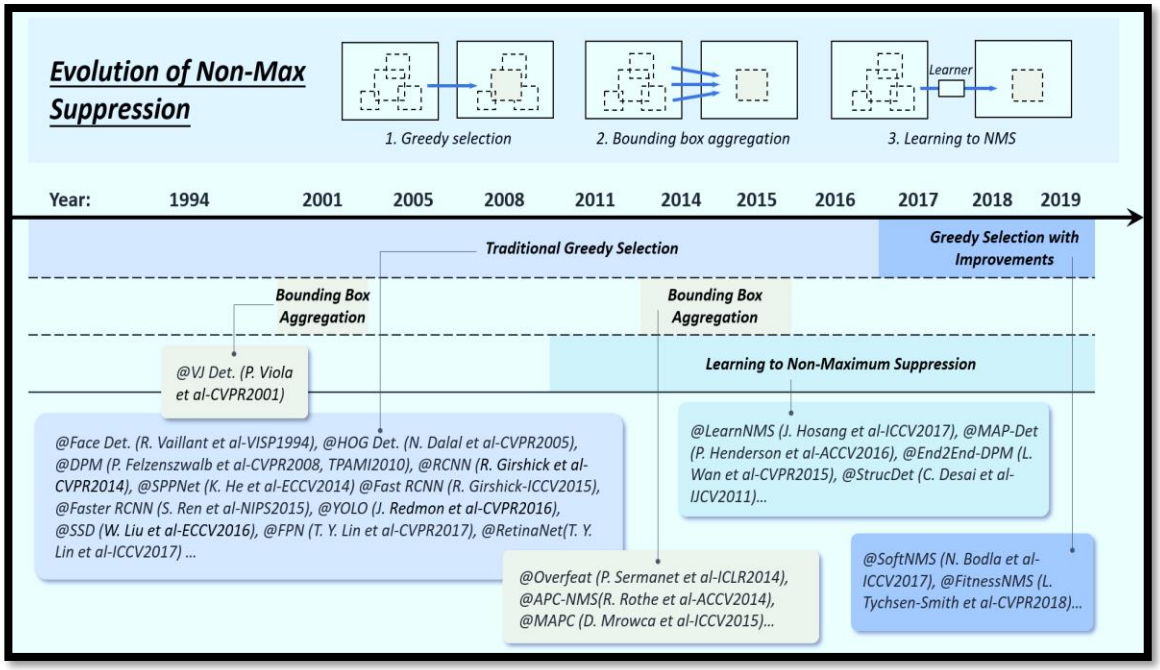
• Detection with global context

As an additional input source for object-detection, global context exploits scene configuration. For early-time object detectors, it is common to integrate a statistical overview of the elements comprising the scene, such as Gist (S. K. Divvala et al 2009) There are two ways to integrate global contexts for modern deep learning detectors. The first way is to make the most of the broad reception area [or the global pooling of a CNN feature] (even larger than the input image). The second way of learning with the repetitive neural networks is to consider the global context as a sequence of information.

• Context interactive

Interactive context refers to the data that transmits interactions, such as constraints and dependencies of visual elements. Object instances are detected and recognized individually for most object detectors, without using their relationships. Recent research has shown that modern object detectors can be improved with an interactive context. Some recent changes can be combined Two categories, in which the first explores the connection between objects and the second explores modelling the dependence between objects and scenes.

Figure.3.9. Demonstrate the Evolution of non-max suppression (NMS) techniques in object detection from 1994 to 2019: 1) Greedy selection, 2) Bounding box aggregation, and 3) Learn to NMS.



3.2.1.3.5.3 Procedural Evolution of Context Priming Technical Evolution of Non-Maximum Suppression

A major group of techniques for object detection includes non-maximum removal (NMS). Since neighboring windows usually have similar detection scores, non-maximum deletion is used to delete replicated bounding boxes and get the final detection result. NMS was not always integrated in early times of object detection. Because at that time the desired output of an object detection system was not completely clear. NMS has gradually been developed over the last 20 years into the following three method groups: 1) greedy choice, 2) the aggregation of bounding boxes and 3) NMS training.

• Greedy selection

Greedy is an old-fashioned but most popular way to detect objects by NMS. For a set of overlapping's, the border box is chosen with the maximum detection score while its neighboring boxes are removed according to a preset overlap threshold The idea behind this process is straightforward and intuitive (say, 0.5). The above processing is carried out in an iterative way. While greedy selection is now the NMS de-factor method, it still has some room to improve. The top box may not be the best one, first of all. Selection is still the strongest basis for object detection today. Second, nearby objects can be suppressed.

Lastly, it does not eliminate miscarriages. Despite recent changes to manuals, the greedy selection remains the strongest basis for today's object detection, despite the fact that its performance has been improved to our best knowledge.

• BB aggregation

The aggregation of BBs is another group of NMS techniques that combine or cluster multiple bounding boxes overlapped into one final detection. The advantage of this method is that object relations and their spatial layout are fully considered. Some famous detectors, such as the VJ detector and the Overfeat, are using this technique.

• Learning to NMS

A recently received group of NMS improvements is the learning of NMS. The main purpose is to think of NMS as a filter to re-score all raw detections and train NMS in an end-to-end way as part of a network. These methods have demonstrated promising results for improving occlusion and dense detection of objects over traditional hand-made NMS methods.

3.3 Collection of Datasets

A dataset is a collection of related observations. The dataset is a collection of associated observation in machine learning which is divided into training and testing data groups. Training data is used to train the algorithm of machine learning, while test data are used to test and evaluate a machine learning solution. The efficiency of the proposed solution is demonstrated using data sets. Obstacle as vehicle detection dataset consists of obstacle as car images, different image features, Each image in the dataset is taken to represent it. The data set contains images of the vehicle closer to real scenarios. More than one researcher has used and brought the data set closer to the real-world situation. Here are few more data sets for an overview of the data sets for vehicle detection. A dataset of 177 pictures from 21 classes was created by Pearce and Pears (M. Everingham et al. 2015), each class consists of five or more images. 85 more pictures (53 classes, each one having an image or two) are added from other unusual vehicle classes. Test data set is created with 177 images in the application of the 'leave-one-out' system. 20 cars used to create the dataset (Jang et al. 2011). They captured pictures for each toy car from 16 various viewpoints.

The training dataset (Psyllose et al.2011) has been composed of 10 classes with 5 pictures per course. The training dataset includes an unknown class with five pictures. The data set is produced with the same pattern except for 10 images in each class.

A data set for side view vehicle images was created (Jonathan et al 2015). Authors used 50% of pictures from each course for training purposes with a maximum maximum of 200 images, while the rest of pictures are used to test. (Baran et al. 2015) created the dataset by downloading internet images or capturing outdoor pictures. The training dataset consists of 17 vehicle classes and 80 pictures per class, 1360 images for the training process are therefore available. The test dataset consists of 2499 pictures and is compiled like a training data set from outdoor and online. The test data set has degraded images with a lower resolution than the training data set. The images are not obscured.

There are three major approaches to existing work using different datasets.

1. The first is based on appearance and identifies cars, including dimensions, shapes and textures, through its inherent characteristics. These methods are based on the cameras' position and position.
2. In the second approach car models are classified with local or global invariant characteristics such as edge-based characteristics (Tsai et al 2007), contour transforming and high-level features such as Speed up robust features (SURF) (Denget al 2009, Mathias et al 2004), gradient histogram (HOG), PHOG and Gabor features (Trabelsi, et al2017); (Hu et al 2015).

3.3.1. Study on Existing Datasets

Far from all research efforts on vehicle as cars, medium-sized datasets are central. There are two main reasons why the large image classification of vehicles is limited.

1. First, only a few large benchmark data sets for vehicle detection are publicly available. This is mainly due to the cost of obtaining class labels. In fact, only a few thousand (or fewer) training pictures comprise most existing fine grained picture classification data sets. Existing vehicle datasets cover a subset of brands and models and those that can be used mainly to detect and estimate vehicles.
2. Second, it is difficult to classify at large because it poses more difficulties than its intermediates. With the appropriate set of training data, designed classifiers can

improve their performance. In fact, for the large range of variations on the angles of view, lighting, and the rather wild appearance difference within the same class, a very high number of images is needed for each class. (Lin et al. 2014) FG3DCar published dataset of 300 thirty-class images. FGComp data (J. Krause et al 2013) includes 8,144 automotive images, covering just 196 models, out of which 60% are 2012 models.

Table 3.3 Demonstrates the summary of the existing datasets addressing the task

Viewpoint	**#Samples**	**#Classess**	**References**
Front	1132	77	V. S. Petrovic and T. F. Cootes (2004)
Front	180	5	D. T. Munroe and M. G. Madden (2005)
Front	830	50	X. Clady, P. Negri, M. Milgram, and R. Poulenard (2008)
3D Free	400	36	J. Prokaj and G. Medioni. (2009)
Front	90	10	A. Psyllos, C.-N. Anagnostopoulos, and E. Kayafas (2011)
Front	262	74	M. AbdelMaseeh, I. Badreldin, M. F. Abdelkader, and M. El Saban. (2011)
Rear	400	10	J. Krause, M. Stark, J. Deng, and L. Fei-Fei. (2012)
Mixed	16185	196	E. Hsiao, S. N. Sinha, K. Ramnath, S. Baker, L. Zitnick, and R. Szeliski. (2013)

3D Free	190	8	J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen. (2014)
Front	6936	29	D. Llorca, D. Colas, I. Daza, I. Parra, and M. Sotelo. (2014)
Mixed	136727	1716	L. Yang, P. Luo, C. Change Loy, and X. Tang. (2014)
3D free	63750	126	J. Sochor, A. Herout, and J. Havel. Boxcars: (2016)

3.3.2 Testing the Existing Datasets

In order to handle real-world surveillance applications, we have compared the effect of multiple network arches, including those evaluated in Diversity in Data.

We analyze the importance of data diversity in the management of real-world surveillance applications in the first experiment. We start our assessments by comparing our data set performance with the Comp Cars data set. Although we have hierarchical labels from make to model, we only use these classes in the data set (125 classes) and we use the label on the make only, following their approach. We divide the images into two halves for training and testing in the resulting 51 classes with the corresponding data sets known as Comp Cars 51 and Stanford dB 51.

Table 3.4 Demonstrates the Number of Images in Each Dataset.

Datasets	#Classes	#Train	#Test
CompCars-51	51	1527	1506
Stanford db-51	51	1986	1984

We investigated the classification accuracy of networks trained on each dataset in confronting with samples from other datasets.

1. The nature of the images in Comp Cars differ greatly from those in Stanford dB images, because they are usually taken with much higher resolution in a more controlled environment.

2. The aim of these experiments is to assess the performance of the model in more difficult scenarios.

3. Generation of a third dataset, called Stanforddb-51 from Comp Cars, by combining the data sets discussed. Table 3.5 summarizes the results of these experiments. We report the accuracy of the Top 1 and Top 3 classification for car makers, which reflects the accuracy of classification considering the top three matches, for each pair of trains and tests. In spite of its significant results in the test set with images of similar character, the model training at Comp Cars is expected to significantly degrade on the test images chosen from Stanford dB -51. However, the combined data set exceeds both previous cases and demonstrates that the use of further training data can boost classification results by increasing data diversity in training examples.

Table 3.5: Demonstrates the Classification results for the models trained on different datasets

	Test			
Train	CompCars-51	Stanford dB db-51	Comp Cars Stanford dB db-51	
CompCars-51	**96.88**	**36.10**	**62.23**	**Top-1**
	97.88	**50.05**	**70.69**	**Top-3**
Stanford dB db-51	**40.28**	**90.26**	**68.22**	**Top-1**
	52.85	**93.48**	**75.93**	**Top-3**
Comp Cars	**96.61**	**94.10**	**95.16**	**Top-1**
Stanford dB db-51	**97.48**	**96.47**	**96.91**	**Top-3**

The classes containing more than 20 images have only been considered. For trains and tests, division of the images into 70% and 30% in every class. The distribution of the various classes based upon the number of images is shown in Figure 3.10.

Figure 3.10 Demonstrates the sub-classes of a selected make the distribution of images is displayed.

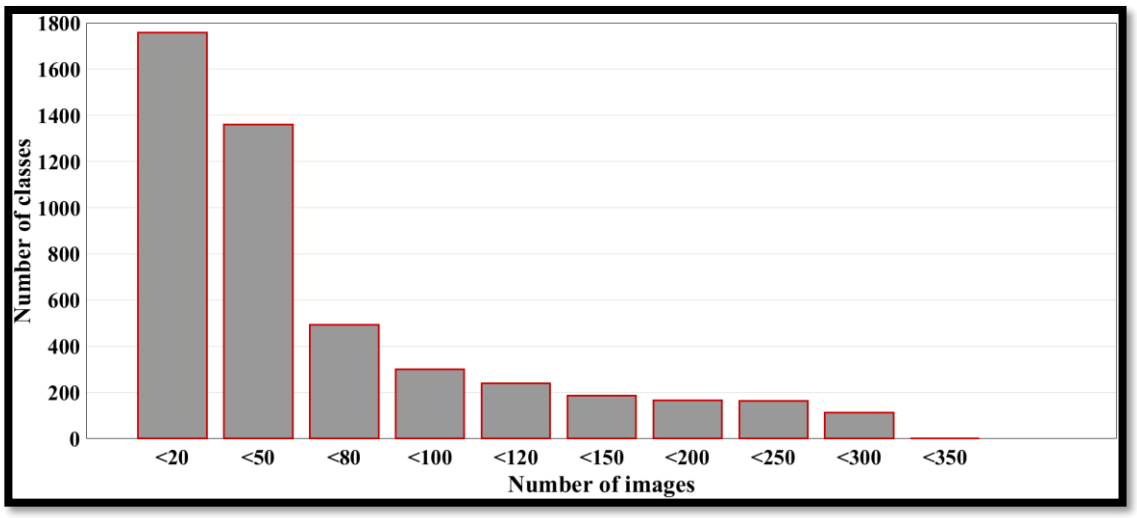
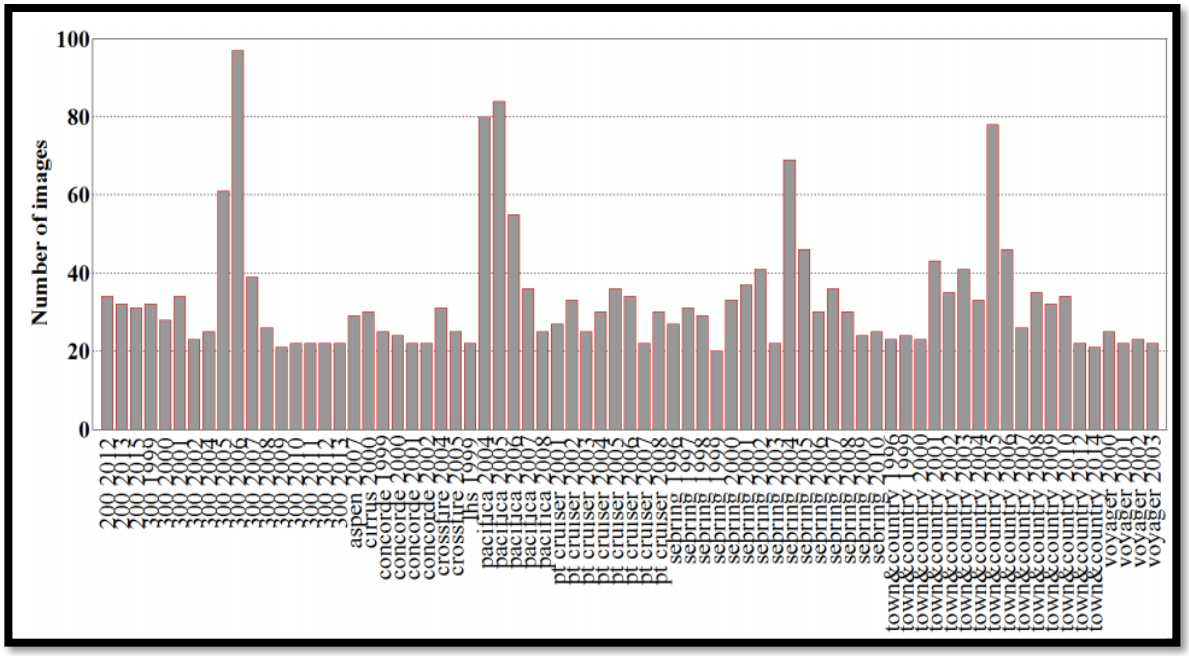


Figure 3.11 Demonstrates Distribution in Comp Cars & Stanford dB dB- 51 of the number of images per model year. Color-normalized with mean and standard deviations from ImageNet scales, aspect relationships, color increases, and horizontal flip increases were used for the "Chrysler" sample.



Test Phase for choosing dataset

All detections have been centered and color standardized by the system during the testing phase. The Top-1 and Top-5, which were trained on db-3036, respectively achieved a precision of 51,76% and 92,90%. In addition to the large increase in classes, this drop in accuracy in comparison with the results. Some predictions show that the model is responsible for differences in viewpoints and lighting conditions. The ground reality class below each image and the odds for the Top-5, with the corresponding class in the top bar. As we can see, the Top-1 match generally relies heavily on other predictions. The multiplicity problems are those samples. A picture of "Toyota Camry 2009," matched as "Toyota Camry 2010," can be found in the left example. The point which is interesting is the fact that in many of these samples, we deal with pictures which are partly occluded or have an uncommon view to observe the learned space of the models for the aforesaid challenges. T-Distributed Randomly chosen grammar images from the presented classes, Embedding (t-SNE) (L. v. d et al 2008) for projection.

3.3.3 Proposed Heterogenous Dataset

Despite ongoing research and the practical interests of car model analysis, due to the diversity and limitations of existing data sets, computer community attention is limited. The dataset used in our experiments contains images taken by various imaging devices and different viewing angles that ensure that different scenarios can be found in a real-life

scenario with a wide variety of variations. The cars are not well aligned and some pictures are irrelevant. The information was collected by crawling websites relating to sales of vehicles. The titles and descriptions provided by the sellers for each post are automatically annotated. In order to remove undesired images from the interior of vehicles and noisy markings, we have developed a semi-automated process.

Days were devoted to the data collection from the internet in order to reach goals that nobody has ever achieved (that is, nobody has collected data on different models of cars). Because internet data is uncoordinated, vehicles, such as a pedestrian, can partially be overshadowed in real world scenarios. Images from various points of view can be captured. The weather and lighting conditions are continually changing. Much time to remove irrelevant pictures from the data. In simple terms, this made the data quite pure.

In this study, 141.941, consisting of 7.232 MM, are examined in filtered data sets (i.e., classes). Honda Civic, Figure 3.11, is the most common vehicle in the dataset. Displays the number of pictures and models in the dataset of the Top 20. The data set proposed is a collection of images in dark and rainy samples with strong shadows and image's view, an important parameter when the data collection as car recognition images for obstacles is created. Thus, online data collected from the existing images database of the vehicle is sorted manually and finally only because of the obstacle stored in the proposed heterogeneous dataset. Data is only collected from Comp Cars dataset.

Figure 3.12. Demonstrations the Number of top 20 images and models in the internet dataset collected.

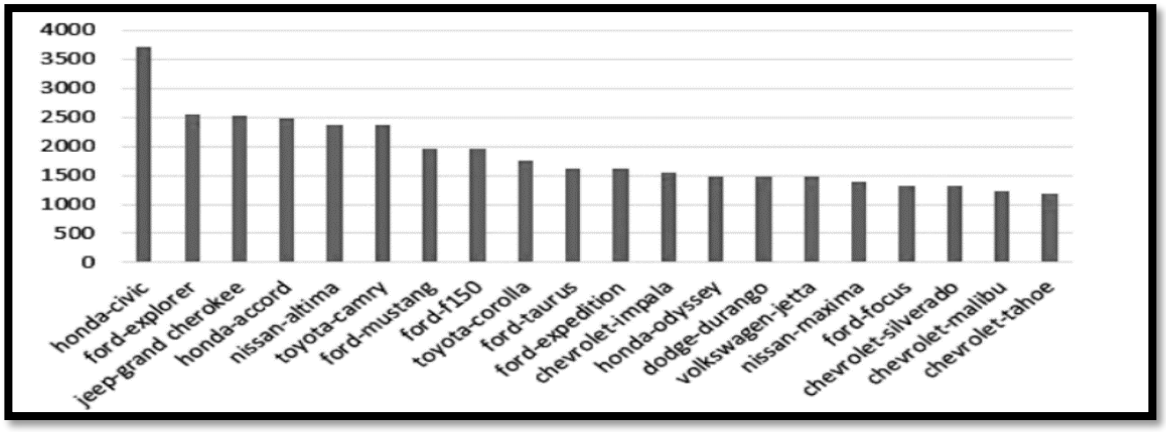


Table 3.6 Demonstrations Comparison with the existing datasets with the proposed heterogenous Dataset

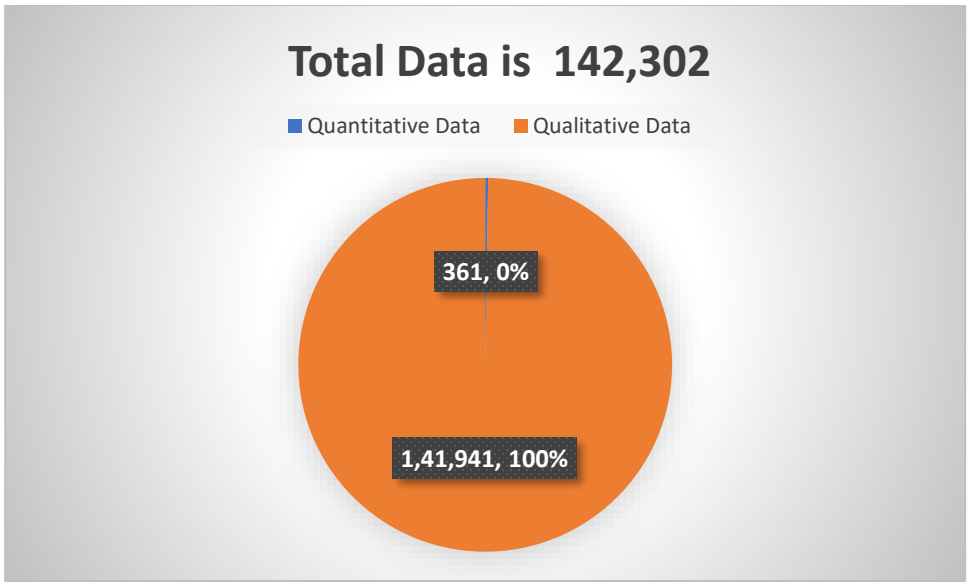
Dataset	No. Images	class
CompCars-51	8144	196
Kaggle-dB	10,622	29
Total Proposed heterogenous Dataset	141,941 images	7,232

The Existing dataset consist of some dataset of CompCars-51 dataset with images of 8144 and class 196 and Kaggle-dB with images of 10,622 and classes of 29. Thus, the Proposed Heterogenous Dataset is collected from online and images captures at heterogenous climates of car as obstacles. Finally, the proposed heterogenous contain images of 141,941 and the classes is 7232.

Table 3.7 Demonstrations of Total Data Used for the Research Study

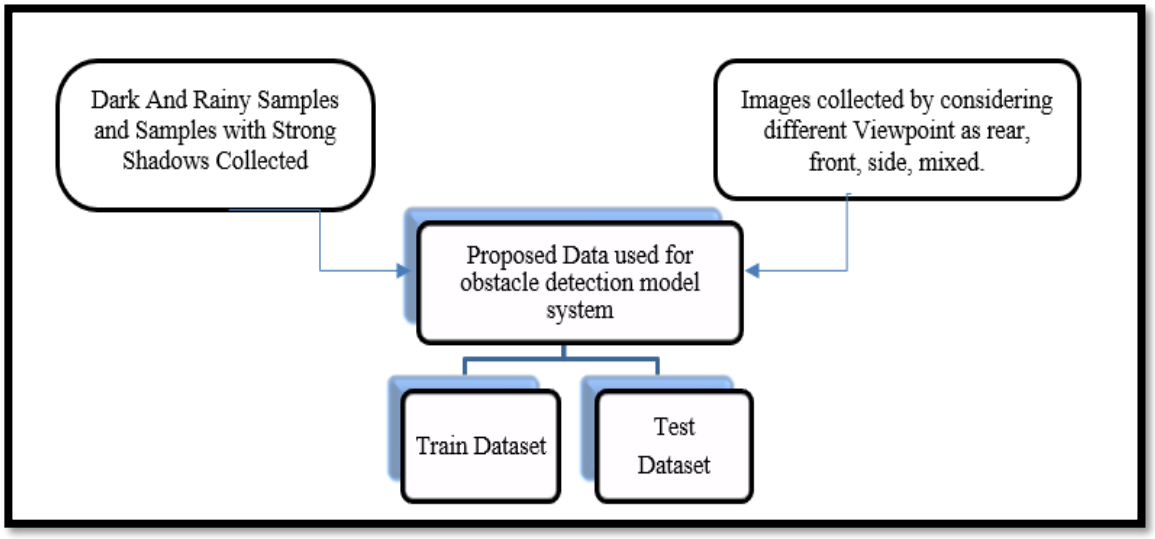
Data	Amount
Quantitative Data	361
Qualitative Data	141,941
Total Data	142,302

Figure 3.13: Demonstrates the Quantitative and Qualitative data for the Research Study.



The figure shows the data collected through the survey questionaries using and the Quantitative data represents the data used for testing and testing for the Proposed model.

Figure 3.14. Demonstrations the Processes of Proposed Datasets



The data set of Comp Cars & Kaggle Car Dataset is publicly available and has been used by the researchers in various studies. The dataset is divided into a test dataset. In this dataset, training data and models are divided into 29 different classes. The training data set contains information on actual car manufacturers and models. The test data set also contains model data, but is used only to compare the prediction for the ODMS system

with the actual model information. The proposed dataset is thus trained in 80 percent training data and 20 percent test data. An inventory of associated observations is a dataset. The data set is a collection of related observation in machine learning which is divided into datasets of training and testing. The training dataset is used for training the algorithm for machine learning, and the test dataset is used to test and evaluate a solution based on machine learning. The efficiency of the proposed solution is demonstrated using data sets. As a vehicle detection data set, the obstacle is that car pictures represent different images in each dataset. In test times, Dataset also receives training with a few unknown models of the car, which are then added to a training dataset by name unknown1 and then fed with the appropriate model information in a training system at monthly audit. unknown obstacle. The proposed heterogeneous datasets are therefore updated with new data on a timely basis.

Table 3.8. Demonstrations some samples of training and testing dataset.

Vehicle	Model	Train	Test	Vehicle	Model	Train	Test
SUZUKI	BALENO	972	194	NISSAN	MAGNITE	36	7
SUZUKI	SWIFT	556	111	NISSAN	LEAF	119	24
SUZUKI	DZIRE	546	109	NISSAN	X-TRAIL	96	19
SUZUKI	VITARA BREZZA	240	48	NISSAN	TIDA	22	4
SUZUKI	ALTO	244	49	NISSAN CEFIRO	CEFIRO	103	21
SUZUKI	WAGON	36	7	NISSAN LIVNA	LIVNA	119	24
SUZUKI	CELERIO	145	29	NISSAN MARCH	MARCH	96	19
SUZUKI	IGNIS	265	53	NISSAN SENTRA	SENTRA	22	4
FORD EXCAPE	EXCAPE	84	17	NISSAN SENTRA	SENTRA	24	5
FORD LIATA	LIATA	157	31	NISSAN TEANNA	TEANNA	66	13
FORD MONDEO	MONDEO	208	42	NISSAN TIDA	TIDA	108	22

Vehicle	Model	Train	Test	Vehicle	Model	Train	Test
SUZUKI	BALENO	972	194	NISSAN	MAGNITE	36	7
SUZUKI	SWIFT	556	111	NISSAN	LEAF	119	24
SUZUKI	DZIRE	546	109	NISSAN	X-TRAIL	96	19
SUZUKI	VITARA BREZZA	240	48	NISSAN	TIDA	22	4
SUZUKI	ALTO	244	49	NISSAN CEFIRO	CEFIRO	103	21
SUZUKI	WAGON	36	7	NISSAN LIVNA	LIVNA	119	24
SUZUKI	CELERIO	145	29	NISSAN MARCH	MARCH	96	19
SUZUKI	IGNIS	265	53	NISSAN SENTRA	SENTRA	22	4
FORD EXCAPE	EXCAPE	84	17	NISSAN SENTRA	SENTRA	24	5
FORD LIATA	LIATA	157	31	NISSAN TEANNA	TEANNA	66	13
FORD MONDEO	MONDEO	208	42	NISSAN TIDA	TIDA	108	22
FORD TIERRA	TIERRA	26	5	NISSAN X-TRAIL	X-TRAIL	37	7
HONDA	CRV	496	99	RENAULT	KWID	22	4
HONDA	CITY	157	31	RENAULT	TRIBER	24	5
HONDA	JAZZ	208	42	RENAULT	KIGER	22	4
HONDA	CIVIC	26	5	RENAULT	DUSTER	24	5
HONDA	AMAZE	496	99	SUZUKI	SOLIO	34	7
HONDA	FIT	9	2	TATA	TIAGO	9	2
HONDA	WR-V	38	8	TATA	ALTROZ	9	2
HONDA CRV	CRV	113	23	TATA	TIGOR	9	2
HONDA FIT	FIT	9	2	TATA	NEXON	9	2
HYUNDAI	SANTRO	26	5	TATA	HARRIER	9	2
HYUNDAI	GRAND I10	16	3	TATA	SAFARI	9	2

HYUNDAI	AURA	12	2	TORD TARCEL	TARCEL	110	22
HYUNDAI	VERNA	26	5	TOYOTA	WISH	20	4
HYUNDAI	CRETA	16	3	TOYOTA	INNOVA	24	5
MAHINDRA	E20PLUS	12	2	TOYOTA	ALTIS	29	6
MAHINDRA	BOLERO	26	5	TOYOTA	CAMRY	34	7
MAHINDRA	XUV300	16	3	TOYOTA	SURF	56	11
MAHINDRA	THAR	12	2	TOYOTA	GLANZA	55	11
MAHINDRA	SCORPIO	26	5	TOYOTA	YAARIS	34	7
MITS.	OUTLANDER	12	2	TOYOTA	MARCH	22	4
MITS. LANCER	LANCER	16	3	TOYOTA	ALTIS	521	104
MITS. ZINGER	ZINGER	12	2	TOYOTA CAMRY	CAMRY	247	49
NISSAN	LIVNA	22	4	TOYOTA INNOVA	INNOVA	15	3
NISSAN	TEANNA	24	5	TOYOTA RAV4	RAV4	79	16
NISSAN	SENTRA	66	13	TOYOTA SURF	SURF	40	8
NISSAN	CEFIRO	22	4	TOYOTA VIOS	VIOS	242	48
NISSAN	SAFARI	24	5	TOYOTA WISH	WISH	175	35
NISSAN	SUNNY	66	13	TOYOTA YARIS	YARIS	155	31
NISSAN	TERRA	67	13		**TOTAL**	**8436**	**1687**

Proposed the five listed characteristics bring the Proposed Heterogenous datasets closer to real-life situations compared to other datasets available:

1. Dataset includes images of moving and stationary obstacle as vehicles up to 65 km/h.
2. The dataset images contain obstacle as vehicles that are view from the front of obstacle as vehicle with several view angles ranging from +20° to −20°C in

comparison to a scene. The dataset images are captured throughout daytime and nighttime.

3. The dataset is created with sunny, rainy and cloudy conditions.
4. An irrelevant object, like a pedestrian, partially overshadowed to some of the obstacles as vehicles.

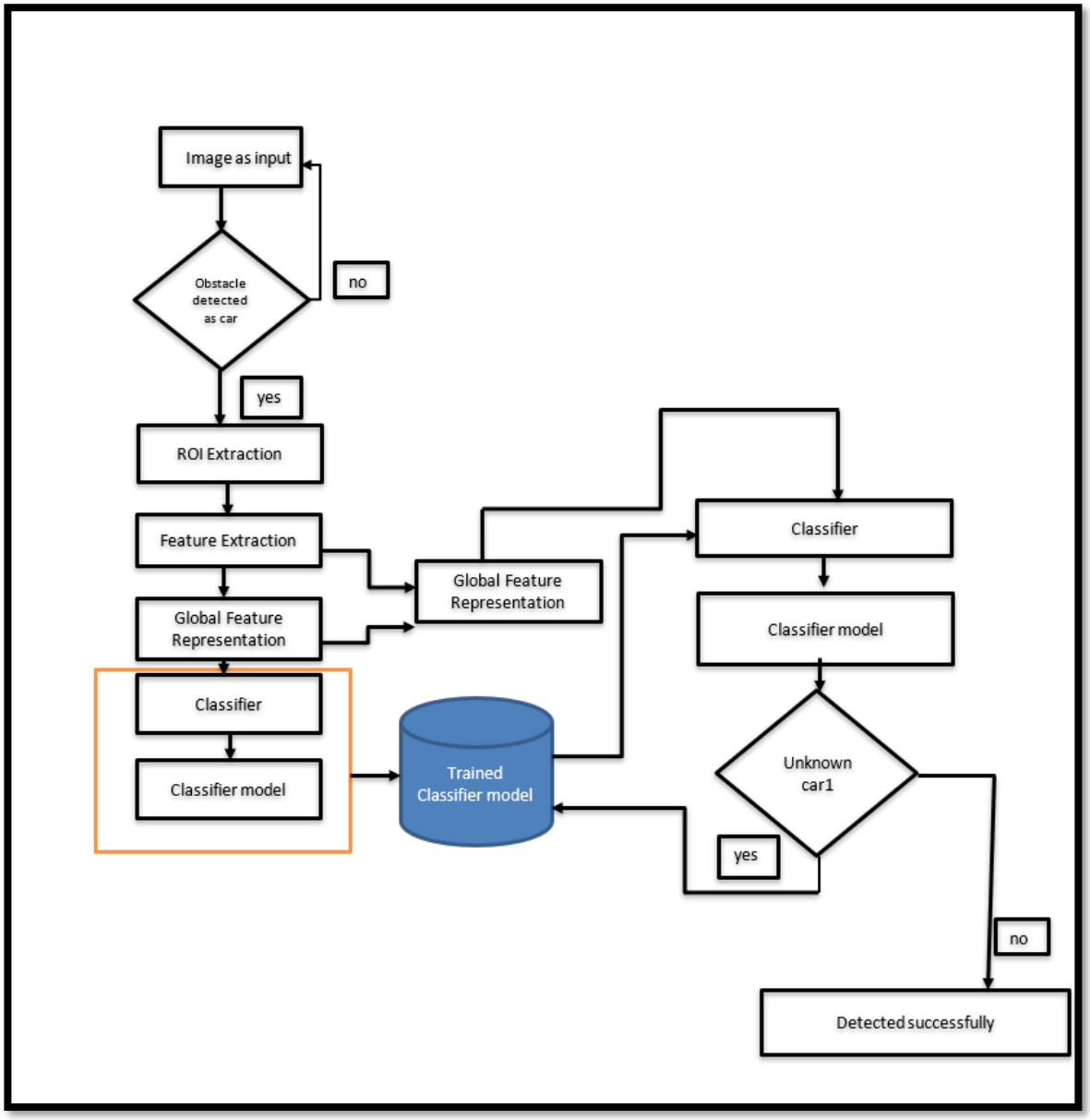
3.4 Proposed Framework for Obstacle Detection Model System

An ODMS system based on machine learning and vision techniques in real time. Computer vision techniques are used to express images in fewer vehicle attributes. Techniques of machine learning are used for vehicle classification. The ODMS system has two subsystems, the subsystem for training and the subsystem for testing. The training subsystem uses a sub-set of the available dataset to train the ODMS engine while the classification subsystem recognizes Models of obstacle as vehicles in new images never used as training. Extraction and feature extraction are common both for training and test tasks in vehicle detection, area of Interest (ROI). In the training process, global representation of functions and classification components is different from the testing process. Depending on the technique applied, the global feature representation module can create a model for encoding image features. The classifier module also produces a model as a result of the training process that the test module then uses to predict the results for the new images examined. The arrows from the training process to the testing process therefore represent the use of the testing models created during the training. Without the global feature representation component, the proposed system works. The features extracted are fed to the classifier directly. Minimizing the global feature representation enhances the processing speed of the ODMS without damaging the accuracy of the recognition. Obstacle Detection Model System (ODMS) can be used to categorize the specific instances of a vehicle based on its model. In this work, the ODMS system is proposed to classify the obstacle as used to categories the particular instances of a vehicle on the basis of its model based on vehicle as car. Smart Cities and ITS applications are actively investigating ODMS systems. In order to recognize videos of individual vehicles, or images, ODMS systems can

utilize the existing infrastructure of installed surveillance and traffic cameras to obtain photographic images and use machine vision methods as an input for ODMS systems. As ODMS system classify the vehicles based on their attributes hence we can use machine learning classification algorithms to create such system. All existing vehicle detection works do not consider as a classification parameter. Unlike existing vehicle detection systems, and classification of the obstacle as vehicles as car-based model. Video surveillance and monitoring and other applications for law enforcement agencies are very advantageous to autonomous vehicle recognition. The availability, inter-class and intra-class similarities of a wide range of vehicles between various vehicles is a challenge for identification. Similarly, the image acquisition process, image quality, weather, lighting, and illumination variation make ODMS more challenging.

The proposed model for the city is now different from the highway. It has been segregated based on the types of input and the speed of the vehicles. In case the speed of a vehicle is less than 50 km/h accordingly it has been proposed that the image be considered as an input for the city-based framework. However, if the speed of the vehicle is greater than 50km/h then it is difficult to capture the image, in such case the video is considered as an input for more accuracy in a highway-based framework.

Figure 3.15.(a) Proposed Framework for Obstacle Detection Model System for Cities.



1. The first task of ODMS is to verify the car as an obstacle in the input image under the speed of vehicles less than 50 . Further processing is not necessary if the vehicle is not present. However, the input further processed to identify the vehicle if the vehicle exists.
2.Once the existence of vehicle is confirmed, next process is to remove the background and unwanted portion of the image and isolate the vehicle in the image.
3. The Interest Region (ROI) represents the vehicle part in the image that gives its distinctive and prominent features. The discriminatory and prominent features of the

vehicle can be easily distinguished. In our work on designing ODMS and used bumper, front lights, and bonnet area as ROI red bounding block. The ROI extraction module removes both the vehicle background and part of a certain image from which the classification performance cannot be aided during classification.

4. The next step is to extract image features and generate an global image feature representation that is used by the classifier to train and predict the system's new images. Extraction and extraction of features are common in both training and testing tasks in the Interest Region (ROI).

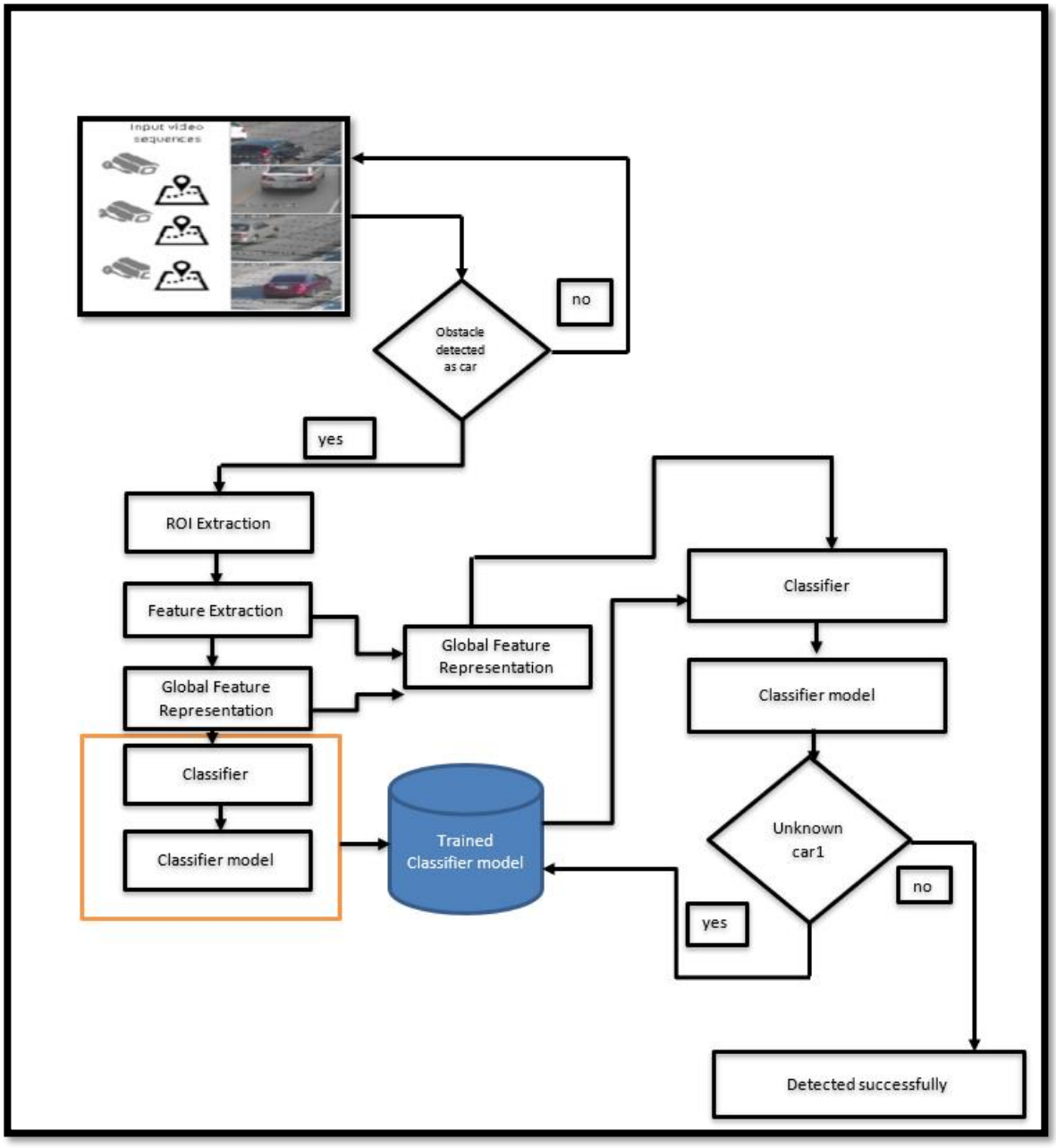
5. The Feature extraction module calculates the discriminatory characteristics of ROI extracted

6. Depending on the technique used, the global feature representation module can create an image coding model. For uniform representation, the Global Feature Representation module uses the entire Image Function Set for global feature representation technique. Extraction of the region of interest is frequently seen as part of vehicle detection.

7. As part of the training process, the classifier module produces a model that is then used by the test module to predict the results of the newly explored images. The arrows from the training process to the testing process therefore represent the usage of models in testing process created during the training process.

The system proposed classifies obstacle as vehicles as car according to type. In comparison to existing systems, the systems proposed increase ODMS efficiency in terms of recognition rate and processing speed.

Figure 3.15.(b) Proposed Framework for Obstacle Detection Model System for Highways.



1. The first task of ODMS is to verify the car as an obstacle in the video frame under the speed of vehicles greater than 50. Further processing is not necessary if the vehicle is not present. However, video frame is further processed to identify the vehicle if the vehicle exists.

2.Once the existence of vehicle is confirmed, next process is to remove the background and unwanted portion of the image and isolate the vehicle in the image.

3. The Interest Region (ROI) represents the vehicle part in the image that gives its distinctive and prominent features. The discriminatory and prominent features of the vehicle can be easily distinguished. In our work on designing ODMS and used bumper, front lights, and bonnet area as ROI red bounding block. The ROI extraction module removes both the vehicle background and part of a certain image from which the classification performance cannot be aided during classification.

4. The next step is to extract image features and generate an global image feature representation that is used by the classifier to train and predict the system's new images. Extraction and extraction of features are common in both training and testing tasks in the Interest Region (ROI).

5. The Feature extraction module calculates the discriminatory characteristics of ROI extracted

6. Depending on the technique used, the global feature representation module can create an image coding model. For uniform representation, the Global Feature Representation module uses the entire Image Function Set for global feature representation technique. Extraction of the region of interest is frequently seen as part of vehicle detection.

7. As part of the training process, the classifier module produces a model that is then used by the test module to predict the results of the newly explored descriptions. The arrows from the training process to the testing process therefore represent the usage of models in testing process created during the training process.

The system proposed classifies obstacle as vehicles as car according to type. In comparison to existing systems, the systems proposed increase ODMS efficiency in terms of recognition rate and processing speed.

3.4.1 Obstacle as Vehicle Detection

The proposed algorithm to identify obstacles in front of the vehicle during movement. In order to calculate the obstacle's depth, the collision status of the approaching obstacles is then estimated. The novelty and the key to this algorithm is to estimate the size ratios of the obstacles coming from the successive frames. The change in the size of the detected

points (diameter) and in the size of the convex hull (area) built from these points is calculated. This is achieved by evaluating. When certain empirical values are greater than the size ratios, this means that the obstacle is identified and can cause a hazard as shown in Algorithm 3.1.

Figure 3.16. Demonstrates the Concept of approaching obstacle as vehicle detection.

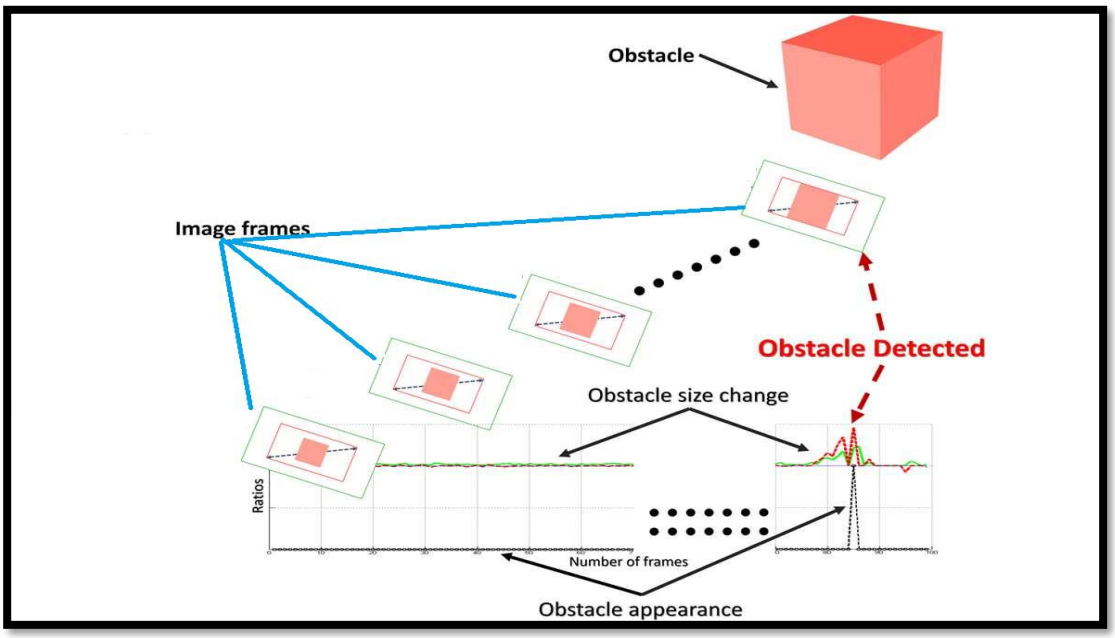
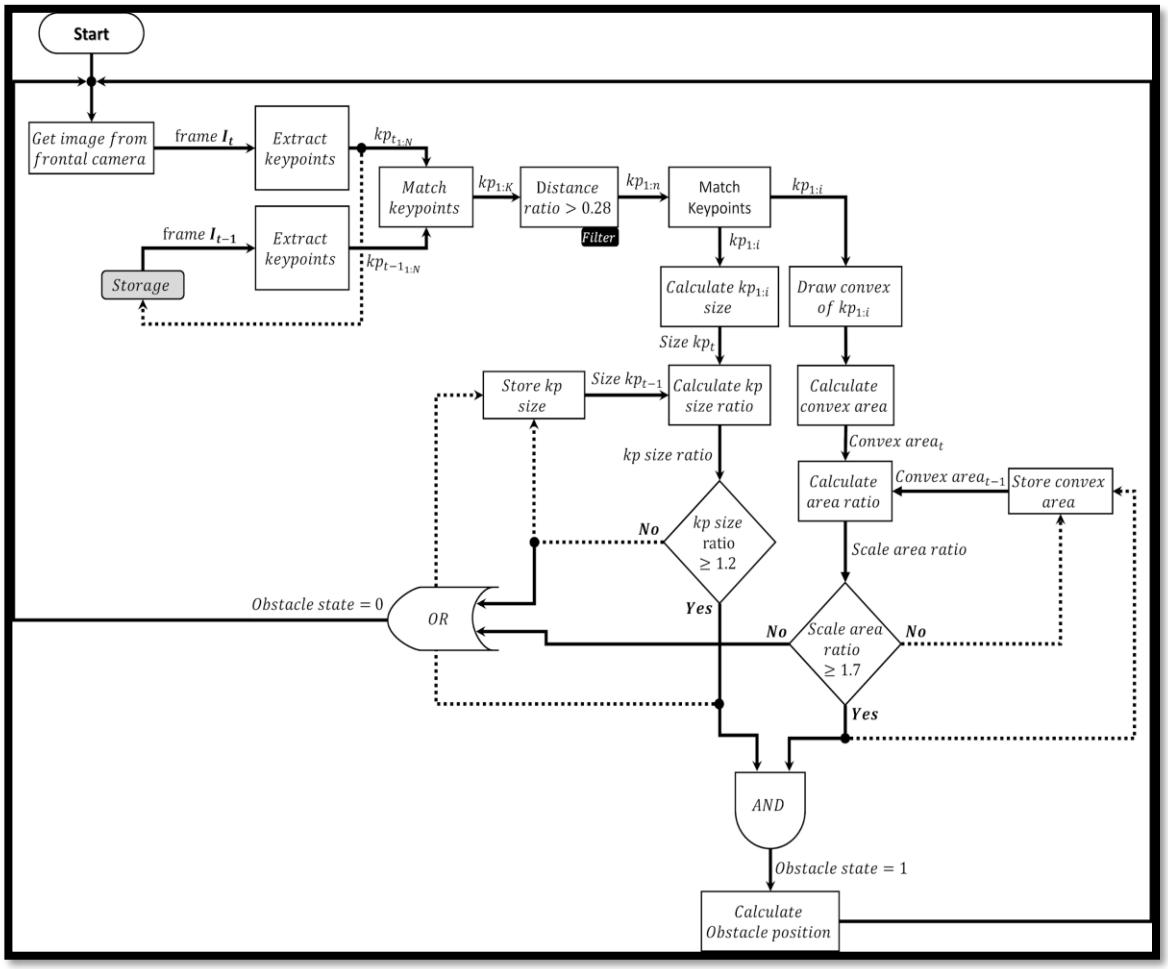


Figure 3.17 Demonstrates the Proposed approach for Obstacle as vehicle detection



A region of interest (ROI) image to be treated rather than the entire image in this step. The results derived from the experiments are based on the selection of the ROI. It has been discovered that no danger is caused by an object detected from the ROI area and it can only be regarded as an obstacle to objects detected in this ROI. In addition, the processing of the ROI rather than the whole picture reduces the calculation time significantly. The feasibility of this approach was demonstrated by the test conducted and the results are discussed. Different conditions such as lighting variation that can lead to noise and error affect the captured frames. However, even under these conditions, the key points must be accurately extracted. The SIFT algorithm is therefore used because of The ability to precisely identify and localize the feature points even with the specific scale and

rotation properties of different image conditions. According to Algorithm 3.1, All key points are detected and the two consecutive frames of their descriptors are extracted, whereby a position vector (x, y) and a size (s) of each key point are obtained. The Brute-Force algorithm is applied after detection of key points to match the key points of the two pictures, and returns only points found in both pictures. Algorithm 3.2 demonstrates the Brute-Force approach concept to match and locate a pair of points at the smallest distance.

The matched key points are filtered to increase accuracy, eliminating those with an over-empirical threshold value with a minimum distance ratio (0.28). Let mkp is the filtered matching key point that is as follows calculated:

$$mkp(n) = \begin{cases} (x, y, s), & \text{distratio} \leq 0.28 \\ 0, & otherwise \end{cases} \qquad \forall n \in K \tag{1}$$

When the key point size (diameter) is s, distratio is the minimum distance of the keypoints matched and K is the number of key points matched. Subsequently, the keypoints obtained by equation (1) are compared to a second or the first frame, and only if its size increases, will the algorithm return the matched key points.

$$mkp(i) = \begin{cases} (x, y, s), & Size(mkp_2(i) > mkp_1(i)) \\ 0, & otherwise \end{cases} \qquad \forall i \in n \tag{2}$$

The next step is to determine whether a frontal obstacle can be detected. Hence, an object of interest (OOI) is created in both frames by creating a convex hull of the corresponding points, from the extracted and filtered key points by Equation (2), as shown in Figure 7: The following:

$$C = \sum_{i=1}^{N} \lambda_i mpk_i | (\forall i : \lambda_i \geq 0) \tag{3}$$

Where C defines the convex hull, and where a μi is a nonnegative assignment of mpki key point weight oscillate between N and oscillate = 1 Next, it is considered that each convex hull is regarded as irregular in order to assess changes in the area of the detected obstacles polygon. Therefore, the area of C can be calculated as follows for a given C as a convex hull:

$$C_{area} = \frac{1}{2}\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \\ x_1 & y_1 \end{vmatrix} = \frac{1}{2}\begin{matrix}[(x_1y_2 + x_2y_3 + x_3y_4 + \cdots + x_ny_1) \\ -(y_1x_2 + y_2x_3 + y_3x_4 + \cdots + y_nx_1)]\end{matrix} \tag{4}$$

Where vertices are x(1:n) and y(1:n), and where n is the polygon sides. In the end, the size ratio between the matched keypoints and a convex hull area are calculated as follows from the second to the first frame:

$$ratio(mkp) = \frac{1}{N}\sum_{i=1}^{N}\frac{Size(mkp_2(i))}{Size(mkp_1(i))} \tag{5}$$

$$ratio(C) = \frac{Size(C_2)}{Size(C_1)} \tag{6}$$

Then the algorithm calculates the collision condition whether or not the coming obstruction is a collision.

$$State = \begin{cases} 1, & ratio(mkp) \geq 1.2 \wedge ratio(C) \geq 1.7 \\ 0, & otherwise \end{cases} \tag{7}$$

Next, an empirical study has been developed concerning the relationship between the keyboard sizes, the area of the obstacle and the distance between the nearest obstacle. This relationship was assessed by various indoor and outdoor experiments. The best ratios are [1.2–1.5] and [1.7-2.0] respectively for the key points size and obstacle area if the object has a constant movement speed. The collision condition of the obstructions detected by the Monocular Camera is 1 when an obstacle exists or 0 when no obstacle is identified.

After the obstacles have been detected, Pr estimates the position of the extreme outer points which construct an obstacle in image (Pl, Pr), where Pl is the point of a position

which has the minimum value of x, Pr is a maximum of x, and Pu and Pd are the minimum and maximum values of y, respectively. Finally, left, right zones are calculated as collision-free zones

$$\tau = \left(\tau_l, \quad \tau_r\right) = \left(Zone_{L_{width}}, \quad Zone_{R_{width}}\right) \tag{8}$$

The width and high that the points (Pl, Pr) create are the following ZoneLwidth, ZoneRwidth:

$$Zone_L = Rectangle\left[(0, P_{u_y}), \quad (P_{l_x}, P_{u_y}), \quad (0, P_{d_y}), \quad (P_{l_x}, P_{d_y})\right] \tag{9}$$

$$Zone_R = Rectangle\left[(P_{r_x}, P_{u_y}), \quad (ROI_w, P_{u_y}), \quad (P_{r_x}, P_{d_y}), \quad (ROI_w, P_{d_y})\right]$$

Where, w and h respectively are the ROI width and height.

3.4.1.1. Highway Obstacle as Vehicle Detection

The detection of road obstacles is a difficult problem. Highways provide a dynamic, unknown environment with real-time limitations. The high travel speeds also force a system to detect objects in long distances. Most of the work focused on detecting large objects, particularly other vehicles, although a number of competing methods were proposed for detecting obstacles on the road. There are several methods which can detect moving vehicles successfully; but the more difficult problem remains unresolved in finding small, static road debris like tyres or craters. Let us first define a couple of terms. The minimum distance required for the vehicle to stop completely from braking start is the stop distance. Although we can often switch lanes when we see an obstacle on our present lane, our basic operating mode must enable the vehicle to stop fully before hitting the object. Our first rule says that the distance required to look for a block detection system is when the car travels to the direction, processing and detecting of the object and braking, plus stopping distance where v0 is the initial speed of the vehicle, and then braking deceleration. The latencies related to sensor acquisition, computer processing and braking reaction are (a <0.) and tsense, tprocess and tbrake.

$$lookahead = v_0 t_{delay} + d_{stopping}$$

$$\Rightarrow lookahead = v_0 (t_{sense} + t_{process} + t_{brake}) - \frac{v_0^2}{2a} \tag{10}$$

In the second rule, the rate of flow, µcyc, must not exceed unity (to ensure road overage), where the flow ratio is:

$$\rho_{cyc} = \frac{vT_{cyc}}{\Delta R} \tag{11}$$

Where Tcyc is the time of cycle and¨äR¨is the pixel projection examined in one round on the ground plane by the algorithm. The terrain over-sampling factor is equivalently 1/Ścyc. More than one over-sampling factor can allow the system to track a potential impediment to the reliability of the system over several frames. § R is restricted by the sensor's vertical field of view although only part of the available field of view can be used for an algorithm. This rule can be translated into a vertical field of view limitation. The VFOV for the performance requirements is necessary in os. The following equations can be resolved for the following:

$$\tan(\alpha + \theta) = \frac{\tan\alpha + \tan\theta}{1 - \tan\alpha\tan\theta} = \frac{L + \Delta R}{h} \tag{12}$$

$$VFOV = \theta = \operatorname{atan}\frac{h\Delta R}{h^2 + L^2 + L\Delta R} \tag{13}$$

To determine the acuity of the sensor, Kelly uses two rules. The first is to calculate the vehicle's pitch when sailing rough terrain, when significant elevation differences may be present between the two vehicle axles. The first is the minimum acuity rule. This is important for highway applications where we plan to avoid anything that contravents the assumption of a flat ground. This is not considered here. The second is based on the Nyquist theorem, which says that a vertical obstacle must be identified with at least two pixels.

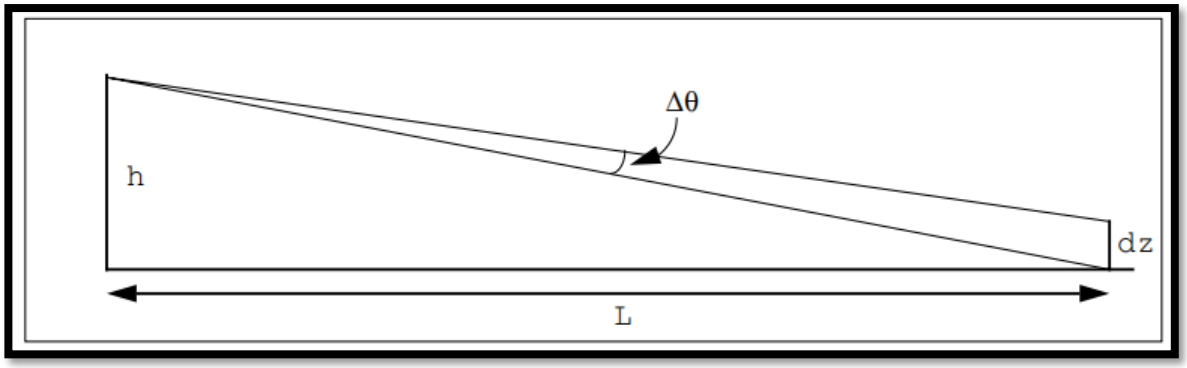
$$dz \leq \frac{p}{2} \tag{14}$$

Where dz is a vertical surface projection of a single pixel. Dz can be linked to the sensor's vertical angular resolution with the next equation:

$$\Delta\theta = \operatorname{atan}\frac{h}{L} - \operatorname{atan}\frac{h-dz}{L} \qquad h,dz \ll L \quad \Rightarrow \Delta\theta \approx \frac{dz}{L} \approx \frac{p}{2L} \tag{15}$$

However, the necessary vertical angular resolution is generally greatly underestimated because the super-sample is desirable in order to improve the detection system reliability. However the required vertical angular resolution is generally greatly underestimated because over-sampling is desirable for enhanced detection system reliability.

Figure 3.18. Demonstrates vertical height a single measurement subtends, dz, is connected to the vertical angular resolution Δθ.



The last of Kelly's requirements, on system fidelity, concern the vehicle's ability to fit between two detected obstacles. We expect a very low density of obstacles on the highway (besides other vehicles), however, so we will ignore these rules here. In a highway system, we expect that the vehicle will either stop or move out of a lane if a static obstacle is detected in it. It is instructive to calculate values for a typical highway scenario.

Table 3.9: Demonstrations the Typical Values1 for Highway Obstacle as vehicle Detection

Variable	Meaning	Value	Given (G) / Calculated (C)
v	vehicle velocity	26.7 m/s (60 mph)	G
a_{dec}	max. deceleration	-6.9 m/s2 (0.7 g)	G
t_{delay}	processing/braking delays	0.5 s	G
L	Lookahead distance	65 m	C
p	Obstacle height	0.2 m	G

Table 3.10: Demonstrations the Typical Values2 for Highway Obstacle as vehicle Detection

Variable	Meaning	Value	Given (G) / Calculated (C)
T_{cyc}	Cycle time	0.3 s	G
ΔR	ground range	8.01 m	C
h	height	1 m	G
VFOV	vertical field of view	0.0017 rad (~0.1o)	C
$\Delta\theta$	vertical angular resolution	0.0015 rad (~0.1o)	C

3.4.1.1.1 On-Road Compared to Off-Road.

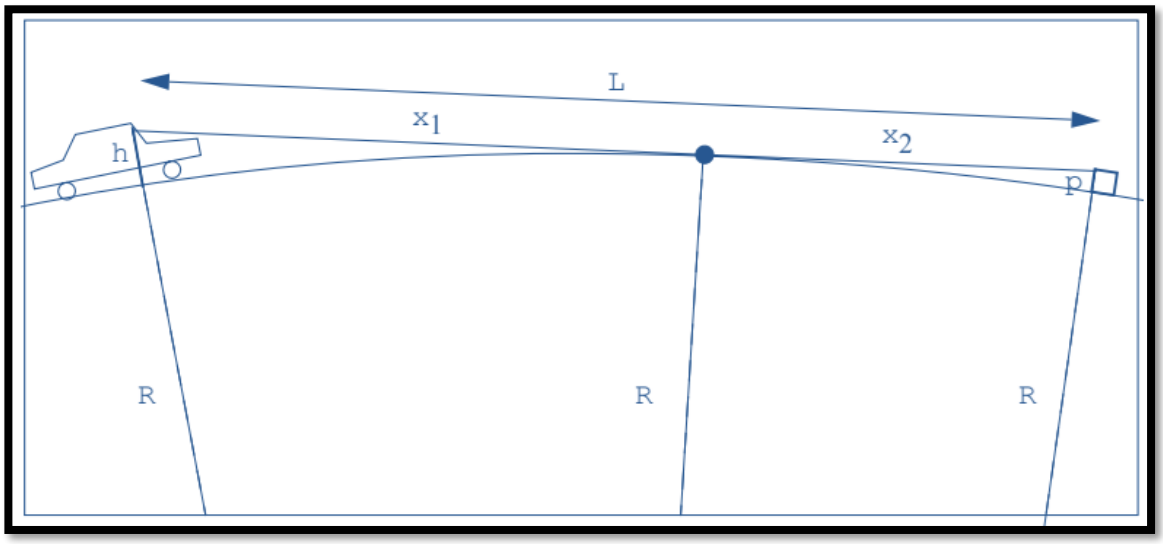
Typical speeds of 5 to 10 mph are available for cross-country applications. However, many times this is the typical highway speeds. As the stop distance is commensurate with the speed square, the stop-distance is much greater, probably by 2 orders of magnitude, on-road applications. Moreover, a trans-country vehicle moving relatively slowly can climb bigger objects without damaging the vehicle than the typical highway passenger car. To detect objects larger than a few inches in the road scenario, we must be able. When we combine the data together, we see, if standard cross-country techniques with similar sensor field-of-view techniques are used, that the required sensor resolution for road obstruct detection can be 2 to 3 orders in magnitude. We also need to look at a greater amount of ground given higher speeds. While it is generally adversely affected by the impact of latencies on lookout distance for highway applications, the processing system still needs to have small latencies. High latency systems can lead to vehicle control problems. These problems may make the problem on the road almost insecure, since the

problem of the detection of obstacles off the road is still far from solved. Happily, however, roads are fairly flat (locally, at least). In light of our earlier analysis, the need to use that fact to win in this scenario should be clear. While processing methods in cross-country territory must produce explicit maps (due to high obstruction density), estimate object heights and check the conditions for rolling over, etc., these steps are unnecessary for road navigation. We just have to find and avoid areas that violate the limit of flatness.

3.4.1.1.2. Considerations of Road Geometry

We must look at how close a road can be to planning, before we start to use the planarity constraint to solve our obstacle detection problems. The vertical crest stopping the viewing distance is one metric for highway design. The road construction manual allows maximum curvature of the road surface (for a crest) by specifying that drivers in a vehicle have a 1/2-foot object to see at least L on the road. The eyes of the driver are supposed to be 3.50 feet above the road, and L depends on the speed. The specified L shall be 580 feet for 60 mph. With the following diagrams and equations, we can calculate the minimum vertical curvature radius of the road.

Figure 3.19. Demonstrates the design specification for the highway states that the driver must be able to see an object height p = 0.5 feet on the road at distance L at a height h = 3.5 feet over the road surface. L depends on the speed of the road. With L, the radius of the vertical curvature R can be calculated.



From the diagram above, we can write the following equations:

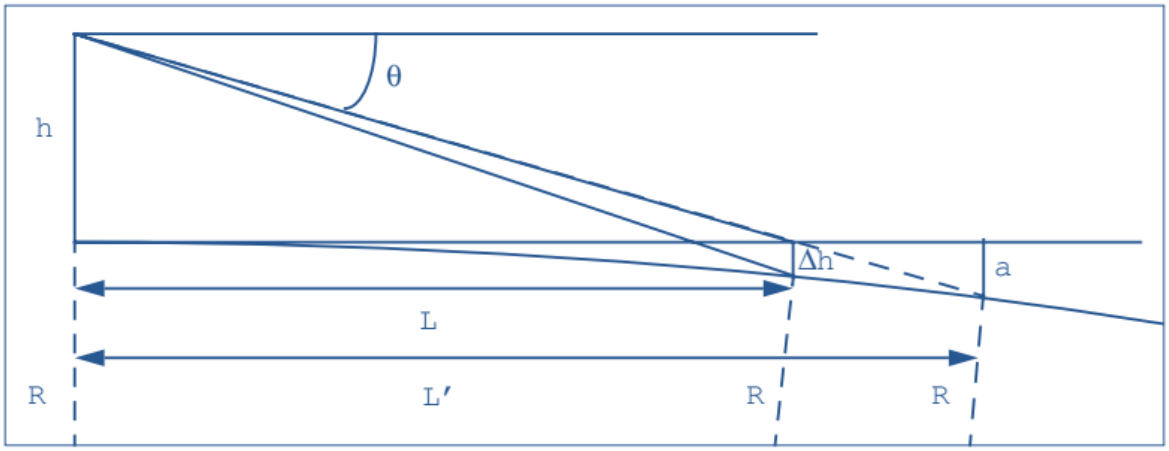
$$x_1^2 + R^2 = (R+h)^2 \qquad x_2^2 + R^2 = (R+p)^2 \qquad x_1 + x_2 = L \tag{16}$$

We have the trivial assumptions and the resolution of the above equations for R:

$$R = \frac{L^2}{2(h+p+2\sqrt{hp})} = 25309ft = 7714.3m \tag{17}$$

Now we calculate how much the road differs from the planar, L. If we have a sensor so that it hits the road at our view L in the planar case, when the road has a maximum curvature, it strikes the paving (with respect to highway specifications).

Figure 3.20. Demonstrates The camera is normally inclined at an angle — a distance L, to cross a planar road. The assumed lookahead distance is nominal surface L.



is the depth of the road at nominal lookahead distance L below the assumed surface. Let

L = 60m

$$\Delta h = R - \sqrt{R^2 - L^2} = 0.234\text{m} \tag{18}$$

$$\tan\theta = \frac{h+a}{L'} = \frac{h}{L}$$

$$\left((R-a)^2 + L'^2 = R^2 \right) \Rightarrow a = \frac{L'^2}{2R} \tag{19}$$

since $a \ll R, a \ll L'$

Substituting in for a, we have:

$$\left(L' = \frac{h + L'^2/(2R)}{\tan\theta}\right) \Rightarrow L' = \frac{Rh}{L} - \sqrt{\left(\frac{Rh}{L}\right)^2 - 2Rh} = 95.74\text{m} \tag{20}$$

We can see, we cannot know a priori at what depth the sensor ray should intersect the road, coupling vehicle and sensor pitch variations with this curvature effect. Clearly, it is unlikely that techniques that try to detect obstacles based on absolute depth will work. Pitch changes in the vehicle and sensor could cause linear sensor methods to have cycling rates faster Faster than the rate of pitch variations, depending on the pitch amplitude, to guarantee coverage. In our calculation for Šenh, we see that the flat world differences can be as big as the obstacles to be found. Allowable alterations in road grades may be almost twice as much for vertical crest curves for "sag" curves. So, for small obstacle detection on many highways, flat world models cannot be relied on simply. On the other hand, a locally flat model leads to far smaller model errors. A maximum residual error x results in the best fit length line L to a circular radius arc R where:

$$x = \frac{L^2}{16R} \tag{21}$$

The first step in the algorithm for obstacle detection is to create a map of disparity. First, we will take into account the flat world assumption, then an incremental polyhedral model. In either case the disagreement is a linear function of the form image line over the area of interest, wherein y is the image row and a and b are the offset and slope in the hederal function model. In both cases, the disparity is a linear function of an image row where y is the image row and b is the offset of and slope of the function over the region of interest. We begin the fundamental stereo equation:

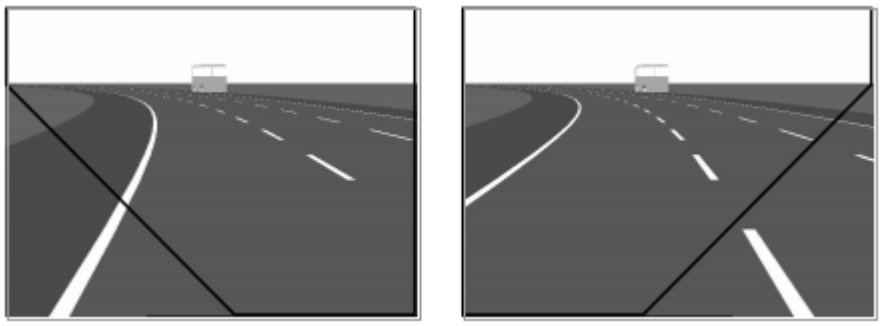
$$z = \frac{bf}{x2 - x1} = \frac{bf}{d} \tag{22}$$

When z is the distance between the world point, b is the distance between the image center's or baseline (measured in units of distance).

$$\sin(\alpha+\theta) = \frac{h}{L/(\cos\theta)} = \frac{h\cos\theta}{L} \qquad d = \frac{bf}{L} \qquad \tan\theta = \frac{y-y_0}{f} \tag{23}$$

To solve differentiation, we have image rows below the horizon in terms of the distance between the image line and the image center (disparity is zero for rows above the horizon). The simple linear difference relationship with the image line allows us to tighten the right image to match the left image with points on the road. If a polyhedral earth is assumed, the correct slope and offset parameters for the difference function must be searched during the operation (every time). The normalized correlation (or other) is calculated. metric such as SSD) over all pixels in the left image that have a match in the right image (as determined For a range of k' and yph values, by disparity function). As the correct function is chosen the disparity function which produces the maximum correlation among the entire images.

Figure 3.21. Demonstrates the Left-right pair of stereos simulated. The pixel locations in both images can be linked to the x2-x1 = k' equation for all pixels on the road (y- yph).



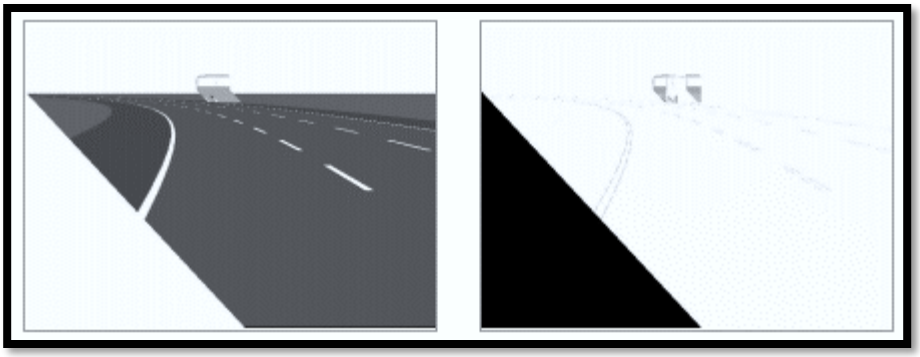
After image rectification, the first step of our algorithm performs a normalized relationship for a variety of k and yph values between the two images. The black areas contain the pixels that are included for a certain k' and yph in the correlation (in this case, the correct values). This method offers several advantages over the traditional stereo method for determining the map of disparities. First, the computer complexity is smaller than O(M*N*Y) in which M and N are the image dimension and Y, in the horizon of flat world cases, is the number of searched values and O(M*N*D), in which D the disparity level is the number that must be matched to each image row (depending on the slope variability and offset parameters). More important, the whole image can be used to match

each pixel, while maintaining the proper depth relation between adjacent pixels (for pixels which actually lie on the road). It uses the known road structure to generate a consistent form map. This work is closely associated with that of Koller, Luong and Malik, which basically uses the same model roads. But they formulate the problem a little differently so that only one degree of freedom can be achieved with the effective sensor angle αeff. Their model does not allow heff changes caused by the curvature of the road. They also differ in their approach to the differences between obstacles and the road, our next topic. They detect obstacles using an image map based on a model while comparing the expected with the actual intensities of the image.

3.4.1.1.3. Detecting Obstacles as vehicles in the Difference Image

In view of the disparity map from the first step, the right image is distracted from the left image by a warped version. In this difference image the points lying on the ground plane should ideally "disappear," so that barriers can be easily detected. This does not happen in practice, of course. A number of reasons may lead to problems. First, discrete images prevent us from sticking one image to match the other and since we once stitch each picture to correct the picture and stitch the right image to match the left image, A number of re-sampling artefacts may be introduced. Second, in general, the intensities of the points correctly matched are not the same as camera gains are possible and the images are noisy. Finally, the discretion of our parameter search space, noises or major barriers that violate our assumptions may not be exact for our model parameters. In view of this problem, we must develop a method to determine if an obstacle or other object causes a non-zero area in the difference image.

Figure 3.22. Demonstrates the When the right image twist through the correct disparity function, result as the left image of the pair that looks like the original stereo pair's left one (the white area at the bottom-left indicates pixels where no match was possible).



If we remove this image from the original left picture, we get a difference picture (we have inverted the image for clarity). Note that with some edge borders the majority of the ground plane disappears. The car in front of us is easy to show One approach to determine if the image of difference causes a non-zero area is to try to estimate and compare the difference in the non-zero region. Discrepancy expected. However, this has the issue of the unreliability and occasional radically incorrect individual disparity computations for a finite window. On the contrary, our approach modelling the intensity of a non-zero region is a barrier. By the disparity function we tweak the right image. For non-integral disparities, linear interpolation is carried out between adjacent pixel values. We compare a rectangular window around the rectangular image for each pixel in the pixel on the left with a rectangular window in the distorted image. The window's size will compensate for noise immunity, sensitivity and computer costs. Large errors of matching show potential barriers. An approach taken from Matthies to determine a 'large-coincident error 'and model the signals from stereo images as offsets with noise added to each image of the same intensity signal:

$$I_l(x) = I(x) + n_l(x)$$
$$I_r(x) = I(x + d(x)) + n_r(x) \quad (24)$$

Where I(x) is a signal of intensity and where d is the true offset or disparity between Il and Ir, and nl and nr is noise (in pixel brightness). Noise of zero average and equal variance, in each image is assumed to be Gaussian, Ś2. The following can be described by a single row in the warped version:

$$I_w(x) = I(x + \Delta x) + n_r(x) \tag{25}$$

Where is the amount of time we are off from the right match? (This should be small). In the case of a single signal (one single image line that does not have any obstruction), and with a Sum of Squared Differences (SSD), we have: We have the following:

$$SSD = \sum_{x_i} (I_l(x_0 + x_i) - I_w(x_0 + x_i))^2 \tag{26}$$

$$\Rightarrow SSD = \sum_{x_i} (I(x_0 + x_i) + n_l(x_0 + x_i) - I(x_0 + x_i + \Delta x) + n_r(x_0 + x_i))^2$$

Where x0 is the point we match, and where xi is the offset of window pixels. For small businesses: For small businesses:

$$I(x + \Delta x) = I(x) + \dot{I}(x)\Delta x \tag{27}$$

Where is the picture measurable intensity gradient. We obtain the following by combining the noise terms, simplifying and linearizing:

$$SSD = \sum_{x_i} \left(\dot{I}(x_0 + x_i)\Delta x + n(x_0 + x_i) \right)^2 \tag{28}$$

We should be near to the correct disparity value of ground plane pixels with the best disparity model for an image. If only sampling problems are responsible for errors. Based on a noise model and gradient information for the original pictures, this error model can be used to determine if the error for the area of a given pixel is too large to be due to differences of sample and noise or to an obstacle. Similar methods can be used based on total differences (SAD). To time, the effect of various camera gains in the model has been ignored. A Laplacian of Gaussian filter is used to reduce the effect of various gains. The Gaussian removes some of the high frequency data, some due to noise, and the Laplacian removes the low frequency data caused by various gains. Unfortunately, the information content in the picture is also considerably reduced. Perhaps a better measure of error not requiring pre-filtered correlation is normalized (as we already used to find our optimal

disparity function). One important element of the thesis work will be the choice of a suitable error measurement to identify the best model of difference and to find barriers in the difference image.

3.4.1.1.4. Obstacle as Vehicle Detection under Fog

Fog is the water droplets which can be seen in the earth surface. It is mainly influenced by the near-by bodies of water, based on the features of earth surface and climatic conditions. The term “FOG” and “CLOUD” can be differentiated according to the moisture content. The visibility of fog is more compared to mist, snow, dew etc. because of the moisture content being more in fog.

occurred are more due to dense fog which affects the visibility of driver. Further it reveals that the snowfall also leads to more

Algorithm 3. 1. Obstacle Detection as vehicle detection

Input: Image frames F

Output: Collision state Obs, Obstacle position τ

1 Define: Current frame Ft , Previous frame Ft -1, Current workspace ROIt , Previous workspace

ROIt -1, Current keypoints KPt [ ], Previous keypoints KPt -1[ ], Number of keypoints (N, M),

n ∈ N, m ∈ M, n = m, x ∈ n, Matched points pts, Distance ratio thresh, Object of interest Convex

2 begin

3 while ismoving() do

4 Ft 1 ← getNewFrame()

5 if Obstacle Detection isActivated( ) then

6 Ft ← getNewFrame( )

7 (ROIt 1, ROIt) ← DefWrkspc(Ft 1,Ft)

8 (KPt 1(N), KPt(M)) ← DetectKeypoints(Ft 1, Ft)

9 (ptst 1(n), ptst(m)) ← MatchSymKeypoints(KPt 1(N), KPt(M), thresh)

10 (ptst 1(x), ptst(x)) ⇒ size(ptst(m)) > size(ptst 1(n))

11 (Convex1, Convex2) ← Create (ptst 1(x), ptst(x))

12 if AreaScale (Convex2: Convex1) ≥ 1.7 and SizeScale(ptst(x) : ptst 1(x) ≥ 1.2) then

13 Obs ← true

14 τ ← CalcPos(Obs)

15 goto Algorithm 3

16 end

17 end

18 Ft 1 ← Ft

19 end

20 end

In addition, processing the ROI instead of the entire image will reduce the computational time significantly. The viability of this approach was demonstrated by the test performed and the results are discussed. Different conditions, such as variations in lighting, which may cause noise and error, affect the captured frames. However, even under these conditions, the most important points must be accurately extracted. The SIFT detector algorithms are thus used because they are capable of accurately identifying and locating feature points, especially with different image status scale and rotation characteristics. All key items are identified and their descriptors removed from the two consecutive frames, whereby each key point is positioned (x, y) and the size(s) are achieved in accordance with algorithm 1. The Brute-Force algorithm is applied after detection of key points to match the key points of the two pictures, and returns only points found in both pictures. Algorithm 3.2 illustrates the Brute-Force approach concept and identifies the smallest distance between a couple of points.

Algorithm 3.2: Brute-Force Matcher

Input: Array of key points in first image kpt1[N],

Array of key points in second image kpt[M]

Output: Indices of the matched points index1, index2

1 Define: Keypoints kp1 = (x1, y1), ... , kpn = (xn, yn), Minimum distance dmin, Distance between keypoints d

2 begin

3 dmin ? 8

4 for i ? 1, N do

5 for j ? i + 1, N do

6 d ? $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

7 if d < dmin then

8 dmin ? d

9 index1 ? i

10index2 ? j

11 end

12 end

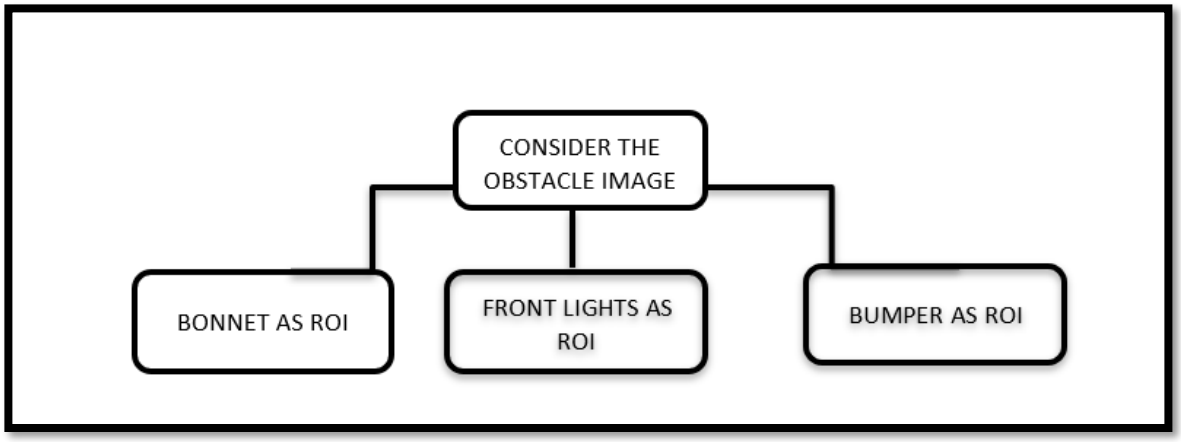
13 end

14 end

3.4.2 Extraction from the region of interest (ROI)

Region of interest ROI to represent the vehicle's part which offers the discriminatory and prominent characteristics in the image. The discriminatory and prominent characteristics of the vehicle can be easily distinguished. In our work to design the ODMS and used bumper, front lights and cap area as ROI, the frontal vehicle pictures.

Figure 3.23: Demonstrates the Region of Interest Extraction of Obstacle as Car for ODMS



The ROI remove the background or the part of the vehicle from an image not helpful for classification and can degrade performance. The ROI extraction module to remove the unwanted part of the image will be removed and the Region of Interest (ROI) extracted once a vehicle is detected. The ROI offers the visual appearance of discrimination and is used to distinguish a wide range of potential designs from models. In the image you will identify the region of interest (ROI) R. The letter R = (lR, rR, uR, bR) defines the roi, and lR, rR, uR, bR is the upper and lower limits. The shadow below the vehicle used to establish the bottom limit (bR); the bottom limit lies on the top of the shadow. By applying edge detector in the horizontal direction, the bottom of the vehicle can be identified easily. The vertical edges of the image is calculated by the horizontal directional edge.

The boundary is then calculated left and right. The left and right frontiers are calculated by means of the width and central lines given in since the central line was determined during vehicle detection using histograms. The

$$l_R = L_{vehicle} - w_v(b_R)/_2 \tag{28}$$

$$r_R = L_{vehicle} w_v(b_R)/_2 \tag{29}$$

Where Lvehicle refers to a central line location and where wv(bR) refers to the width of the car; the width of the car is calculated using the following equation.

$$wv(bR)=(Wv/h)*(f\sin\emptyset - bR\cos\emptyset) \quad (30)$$

Where Wv in the real world denotes the width of the vehicle. H, f, and os are the parameters of the camera. The focal length of the camera is given by f and is positioned at height h with the angle of tilting down to f. Most cameras are fixed in one position to capture vehicle images. The parameters are therefore fixed and can easily be detected by measuring the height and angle of inclination of the camera. The camera parameters can also be included as a common practice in the image metadata.

Lastly, to define the upper boundary uR the boundary between the hood and the windshield. An histogram is developed with the horizontal edge forces along the X-axis for a predefined search window. The histogram top is used to define the top limit uR. The algorithm for vehicle detection works with 98,46 percent precision. Likewise, the process of ROI removal can incorrectly calculate ROI and can lead to wrong acknowledgement. A region of interest (ROI) which, as shown in the example, is a rectangle with the grill part of a car and its head and indicator lights.

Figure 3.24: Demonstrates the Incorrectly Computed ROI

Data Stored in Trained Dataset



Similar ROIs are also detected and extracted from all the data set images of the training. These ROIs as RIs are then used to create a pattern database that saves its vectors. As a real video stream is a sequence of frameworks, the ODMS approaches of the proposed framework are ready to process each frame from the input video sequence, especially the real-time approach. In the first instance, as described at the beginning of this section, the aforementioned ROI detector processed all images from the test data set (including the training dataset). This ROI detector is also used at the beginning of the entire process under real-time conditions. Only if the detector returns the area of interest will the next steps of the process, described in detail under the next sections, be done. Various local characteristics are extracted from ROIs with or without incorporating them to describe the vehicle detections models into Global Features Representations (GFRs).

3.4.3 Techniques for Feature Extraction

Include those feature extractors and descriptors which take into account only key point patches of the calculators, and not the entire image and the ROI, under the main points-based techniques. SIFT and SURF are the two most popular features.

3.4.3.1 Feature Extraction using Scale-Invariant Feature Transform

A bundled function detector and descriptor is the Scale Invariant Function Transformation (SIFT). The detector extracts key points and patches from an image so that the scale, lighting, turning and viewpoints differ accordingly. A compact and robust signature is then created by the descriptor for each patch that recognizes its appearance. Initially, the method was proposed (Lowe et al 2001). SIFT has been popular for the detection and recognition of objects, image recovery and image classification since then. It was also used to try and resolve ODMS issues. The extraction of SIFT features is described as follows in four steps:

1. Space Scale Extreme detection: Invariant key points are identified by the use of Gaussian (DoG) images across different image scales and locations. As illustrated in Figure 3.25, the scale space of a given image is several octaves. The image size is sampled by a factor (for instance, 1/4) for each successive octave. Within each octave, a Gaussian

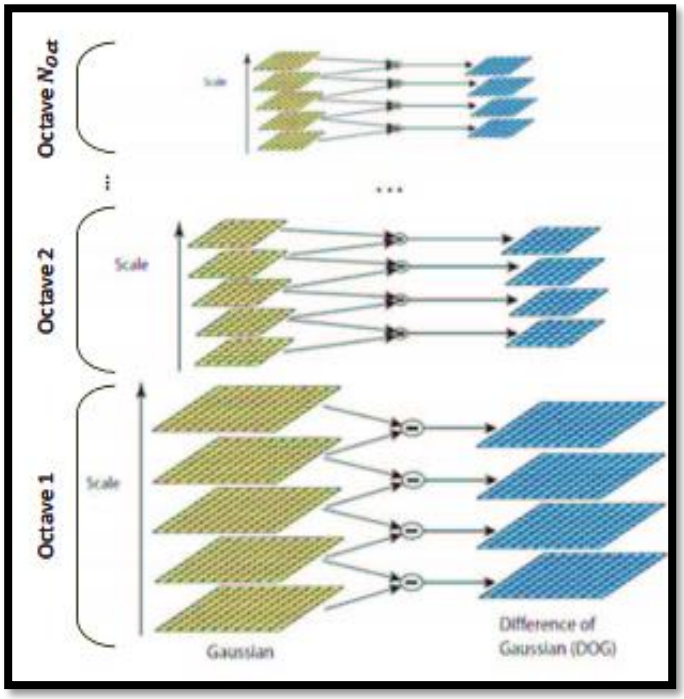
Blur operator with a growing scale gradually blurs the image. Difference-of-Gaussian (DoG) pictures are formed out of every successive pair of Gaussian blurred images. In the final analysis the key point candidates are detected from these DoG images, using maximum or minimal technology (Sivic et a l 2003).

2. Key point Localization: unstable candidates for key items are filtered out in this step. The key unstable points are the ones that have little or no contrast (i.e., the edges and flat areas).

3. Orientation Assignment: In order to achieve rotational invariance, one or more dominant orientations are assigned to each key point position based on the gradients of its local region and, with respect to those dominant orientations, the relevant key item descriptions are displayed.

4. Description of the key point: A 16-16 neighborhood is selected for each key point under the standard SIFT technique, divided into 16 windows of 4 pixels or sub-pixels each. The dominant position of the key point (computed in the previous stage) is removed from the gradient orientation of each neighborhood point.

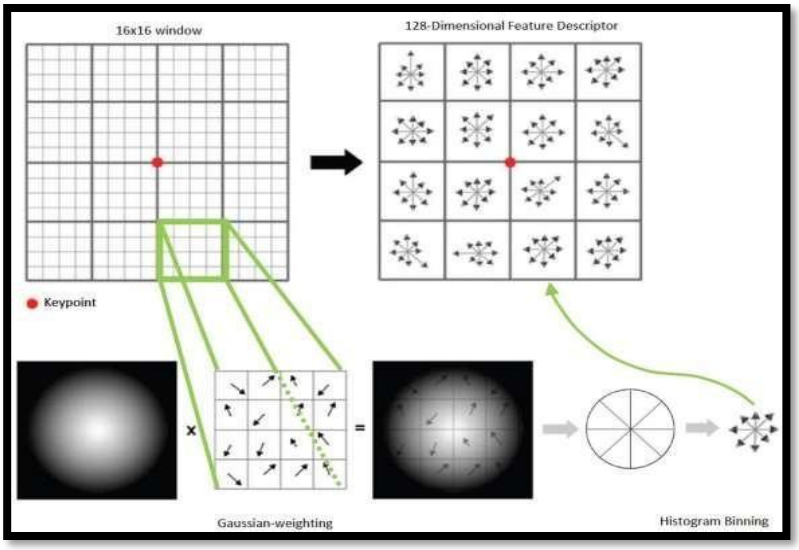
Figure 3.25: Demonstrates the generation step Scale-Space to get DoG images for an input image.



The following gradient sizes and directions are calculated and collected with an 8-bin histogram at each point of a four-to-four window. For example, the 0 pour - 44 pour bin of the 8-bin histogram will include a point with a gradient direction to say 15 pour. The amount added to the +-respective bin of the histogram depends on the gradient size of the point multiplied by a Gaussian weight based on the distance between the two. The further the matter is, the less weight it would be. The Gaussian The 2D Gaussian function is the window weighting with the Ţ of the key point neighborhood in both dimensions equal to half its width (i.e., 16/2 = 8 in SIFT standard). The gaussian weighting window is used to avoid sudden descriptor changes in window positions. The impact of points further away from the descriptor center is also reduced (as they cause most mis-registration errors,)The $4 \cdot 4 \cdot 8 = 128$ values for each key-point descriptor is then obtained. SIFT's invariance is mainly because the key points are extracted from various dimensions and with different Gaussian blur scales. SIFT features are also invariant. The invariance of rotation is achieved by looking at the relative gradient guidelines (regarding the dominant orientation of a key point) while calculating a key points descriptor. Invariance of illumination is achieved by normalization of the vector.

Figure 3.26: Demonstrates the SIFT descriptor computation around a key point.

(Top-Left).



The red key point 16x16 quarter. (Lower left) Gaussian weighting is applied to gradient guidelines calculated in a window at every point. (Lower right-hand) The circle divides the orientations into eight containers (i.e., the Histogram binning step). (Top right). (Top right). The collection from the 4x4 windows comprises a total of 16 8 bin orientation histograms. The arrow lengths are gradient magnitudes while orientation is gradient direction.

3.4.3.2 Feature Extraction using Speed Up Robust Feature (SURF)

Another type of invariant and invariant scale and rotation detectors and descriptor are the Speed Up Robust feature (SURF). In comparison to other feature designers such as SIFT, it has been proposed by (Bay et al. 2008), which has shown its processing speed and precision efficiency. In various applications such as ADAS, SURF has shown encouraging performance (A. Mammeri et al2014). The procedure for building SURF Descriptors is described below.

- Interest points: the points of interest are located on an integrated picture with the Hessian matrix approximation. The part derivatives of the second order of a picture are derived from the hessian array to describe their local curvatures. The rapid processing speed of SURF is the application of integral pictures with blob-like detectors in an approximation of Gaussian second-order derivatives.

- Computer Descriptor: A neighborhood of 4 to 4 blocks is removed around each point of interest obtained in the previous step. Each block has five to five sample points that are regularly spaced. A 4-D vector f descriptor f consists of the combination of the HAIR Wavelet (dx and dy) responses for each block b (where b = 1,, 16) of each specimen point.

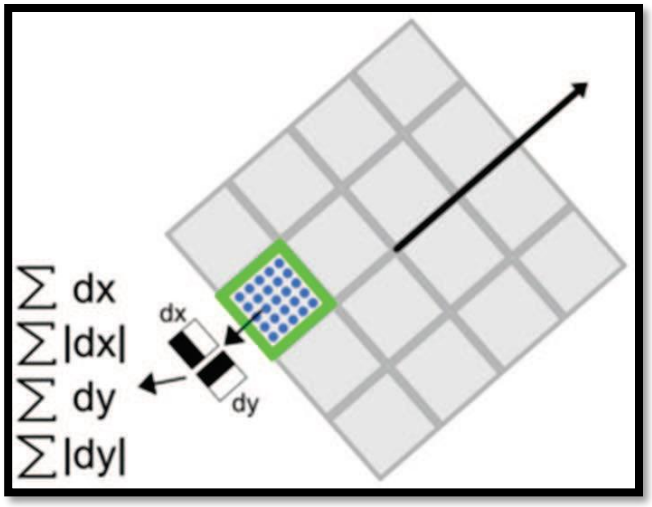
$$fb=(\Sigma dx,\ \Sigma d|x|,\Sigma dy, \Sigma|dy|) \tag{31}$$

Figure 3.27. Demonstrations the one-block SURF descriptor computing method in the area centered around the key. The 4-D vectors of each block as represented by: The SURF descriptor for the square area around an interesting point forms the following:

$$f = [f1, f2, ..., f16] \tag{32}$$

- Orientation Assignment: Each interest point and descriptor is assigned a dominant orientation in order to provide for rotation invariance. First of all, Gaussian x and y weighted hair wavelet responses are calculated for each region. Then the orientation of the largest sum of reactions is estimated as the dominant orientation of the point of interest and descriptor with a rotating orientation window. If the rotation invariance isn't needed to produce a faster SURF, called Upright SURF, this step can be taken (U-SURF).

  Figure 3.27. Demonstrations the one-block SURF descriptor computing method in the area centered around the key.



3.4.3.3 Feature Extraction using Histograms of Oriented Gradients

Dalal and Triggs (N. Dalal et al 2005) for robust detection of human beings introduce Histograms of Oriented Gradients (HOG), which are widely employed in several object recognition works. Vectors for the front of the vehicle and the HOG representations. A sliding window approach is used to calculate the HOG image descriptor or ROI. A certain window, each block with p cells, is divided into a n overlapping block grid. In each cell, weighted pixel gradients are stored in an h-bins orientation histogram, in the same way as SIFT. The histogram of the cells in a block is combined to give the histogram vector at the block level. A robust normalization process is performed on each block to provide a strong lighting invariance. Then, normalized histograms are combined with the window-level histogram vector of all blocks overlapping in a window.

The Histograms Oriented Gradient algorithm can be summarized in four steps:

1. Gradient computation: The gradient gives an image derivative in both directions (x-gradient and y-gradient). As a first step of HOG calculations, the magnitude and the direction of the gradient are calculated. The x-gradient emphasizes the vertical edges (a 90° angle with x-axis), y-gradients show the horizontal edges (a parallel edge to x-axis) and all the others are emphasized with x-gradients and y-gradients. The magnitude of the gradient generates borders and all the flat areas become black. The color image (RGB channels) shall be computed with three gradient sizes and the corresponding direction, and the maximum gradient size and direction shall be chosen for additional calculation on

HOG. The angles of gradients are from 0 to 180° The absolute gradients are called as non-signed gradients when absolute values of x-gradient and y-gradient are used. If the x and y-gradients are used without the absolute procedure,

2.Orientation binning: A histogram for the gradient directions is the next step in the creation of the HOG descriptor. In our experiment, different file sizes were used as opposed to standard 8x8 files and results (for example, 8x8 files). The original image is divided into windows. There are two values for each pixel in the window; i.e., gradient magnitude and direction of gradient. The histogram includes 9 bins, each of which has 20 degrees, to fit the gradient of 0 to 180 degrees (bin 0 degrees, 20 degrees, 40 degrees, etc.). In(N. Dalal et al. 2005) different bin sizes have been compared and it has been concluded that the bin size of 20 grades is better than other sizes. In the case of the signed gradient with the same bin size rather than the announced gradient, the histogram size has been doubled. The orientation-based histogram channel has a weighted value in each pixel in the window depending on the gradient level. HOG does not add the size of the gradient to only one bin, but rather distributes the size of the gradient proportionally into two containers, based on the direction of its gradient. For example, the magnitude of a pixel is 5 and the direction of a gradient is 25 degrees. Bin 20 and bin 40 have a distance ratio of the above-mentioned pixel based on the gradient direction. Therefore magnitude 1.25 is added to bin 40 and magnitude 3.75 to bin 20. For pixels with a gradient of more than 160 degrees, In bin 160 and bin 0, the gradient magnitude is added. The histogram representation reduces the effect of noise; individual pixels may have noise, but they have little influence on the histogram.

3. Block Descriptor and Normalization: To make the larger block, HOG combines windows. There are 4 windows in each block (2 x 2). The blocks overlap in both directions, such as sliding windows. The windows 1, 2, 6 and 7 are combined in one block, while the next block is also made up of windows 2 and 7. With the exception of corner windows (included in one block), each window is part of four blocks and edges (included in two blocks).

Windows histograms are linked to a larger vector called block descriptor. As there are four windows for each block and the histogram for every window has 9 bins (or a 9 x1 vector), the descriptor is therefore of block height 36 x 1 (4 histograms x 9 bins each

histogram). Block descriptors are individually normalized to account for lighting and contrast changes.

The size of the gradient is sensitive to light changes. The intensity of each pixel is reduced in darker images, and the gradient size of the histogram is decreased. We use standardization to separate the descriptor from the lighting changes. Since gradient strengths have to be standardized locally, we group cells into larger space-connected blocks.

4.HOG Feature Descriptor: Finally, for each image, all block descriptors have been linked into a single HOG function descriptor. For instance, we have a 128x64 image. The total number of 16 x 8 windows and 15 x 7 blocks will be set if we split the image into 8 x 8 windows. Therefore, the total HOG descriptor size in this case is 3780-dimensional (15 x 7 x 36).

However, in a dense, grid-like and non-orientated mode, the hog blocks are calculated on a single scale, while SIFT-descriptors are calculated for scale invariable key points and rotated to align the orientation. The HOG block descriptors are used to encode spatial form information in conjunction with the SIFT descriptors.

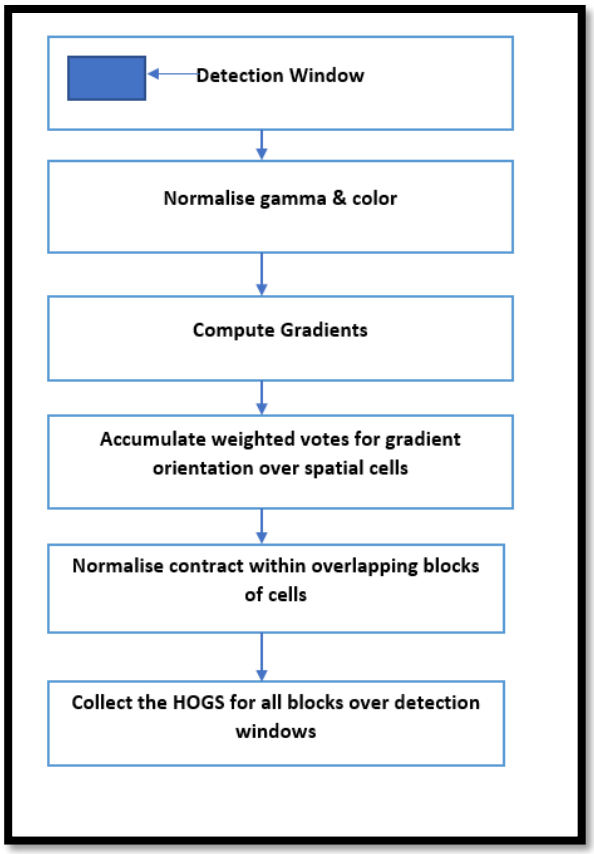
In order to create HOG functional descriptor in our work we used various window sizes. We didn't have several windows to build a larger overlap block. Consequently, in comparative to the standard HOG descriptor, we created HOG descriptors of smaller sizes and improved processing speed. The standard HOG creates a feature descriptor in the previously mentioned format with a size of 3780 elements to a 128x64 picture. of HOG Overlap less descriptor is 1152 elements and the same configuration for the same image.

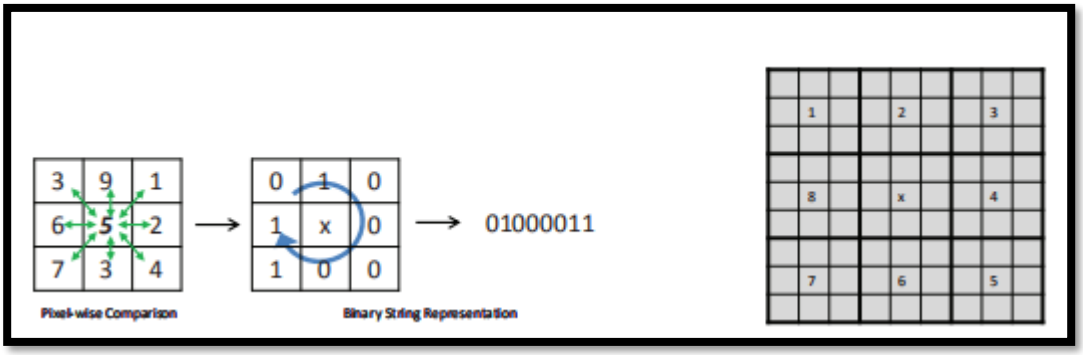
3.4.3.4 Feature Extraction using Multi - Scale Block Local Binary Patterns

The original lbp operator proposed by T. Ojalaet al 2002 labels pixels based on a comparison of each pixel with the neighborhood of 3/3 and on the binary or decimal numbers for each comparison of a pixel, if the middle pixel is higher, "0" shall be inserted into the position of the binary strand, otherwise "1" shall be inserted. In the original ltdp operator proposed by T. Ojalaet al. A histogram of all such pixel labels is then the overall texture descriptor. The improvement proposed in MB-LBP takes sub-regions into account, rather than pixels, in comparison. A square quarter of pixels is considered around

each pixel. The district is divided in uncut blocks (sub-regions) of s/3 /3 dimensions. The (s1) sub-regions' average grey values are compared to the central sub-region.

Figure 3.28: Demonstrates the Computing the HOG descriptor over an input image or ROI for one of the sliding windows.





a) b)

Figure 3.29: Demonstrates the (a) The LBP binary line for a pixel is computed, taking into account its 3x3 neighborhood: Pixel-based comparisons are made; (b) A 9 / 9 operator MB-LBP, with a total of 8 sub-regions in a central area: comparisons between average sub-regional grey value.

3.4.4 Global Representation Techniques

Local characteristics extracted from images often serve as an overview of the object of interest in the image, such as a vehicle from some Make-Model (MM) class. In this section, the most representative of these global representation systems are described and discussed in detail: (1) Representation for grid-based MM vehicles, (2) Mid-Level Representation for Lexicons and (3) Concatenation for the local features. When the limits of these plans are discussed,

3.4.4.1 Grid-based Representation

The grid-based vehicle representation, proposed by (J.-W. Hsieh et al 2013) and (L.-C. Chen et al 2015), uses vehicle local features (such as HOG, SIFT or SURF) to build up a global image while collecting spatial information.

The ROI (Rv) of vehicles obtained by means of the Vehicle Detection module is divided into two division schemes, (1) Direct Division and (2) Inside Division, which are equally sized unoverlapping blocks. The former division system splits Rv into a m/n grid with capots, but Rv is divided into a m/n grid, with the exception of the caproid area. In various MM classes the region of the hood was found to be less informative and discriminatory. Two methods can be used to determine the upper Rv limit for the Direct Division scheme: (a) on the cap limit, or (b) on the basis of Rv width (using a learned width-height ratio). This first is known as Wide Direct Division, and this last is known as Narrow Direct Division (NDD). The two Direct Division methods are illustrated in Figures 3.8a and 3.8b. It was demonstrated in (J.-W. Hsieh et al 2013) that WDD and NDD methods are less effective than Inside Division-based methods.

Figure 3.30: Demonstrates the Grid-based Representation of obstacle as vehicle models: (a) Direct Division scheme; (b) Indirect Division scheme.



Probably because of the inclusion of hood region which has negligible or no discriminatory information, the WDD method results in a low performance. On the other hand, because the NDD depends on a fixed width-height-to-height ratio, some MMs may not be applicable to that ratio. Thus, informative regions of the front (parts or headlights of the air grille) could be ruled out, thereby reducing their performance. Two ways to implement the internal division scheme: (a) Symmetrical Inside Division (SID) (b) Full Inside Division (FID). Only half of grid blocks, e.g., the left half, are chosen in the former). All grid blocks in the latter are taken into account. The Two strategies are employed to build a global representation: (1) block-wise concatenation, and (2) block-wise grouping, according to the type of local features used. Block-wise concatenation is used for window-based local features like HOG in which the block descriptors are linked to several combination patterns. The two examples of block combination patterns For a grid with 3 blocks, 6 blocks (m = 3, n = 6). (1×1), two (1×2), four (2×2), six (2×3), nine (3×3), and eighteen (18×1) A total of 51 block combination patterns result in blocks.

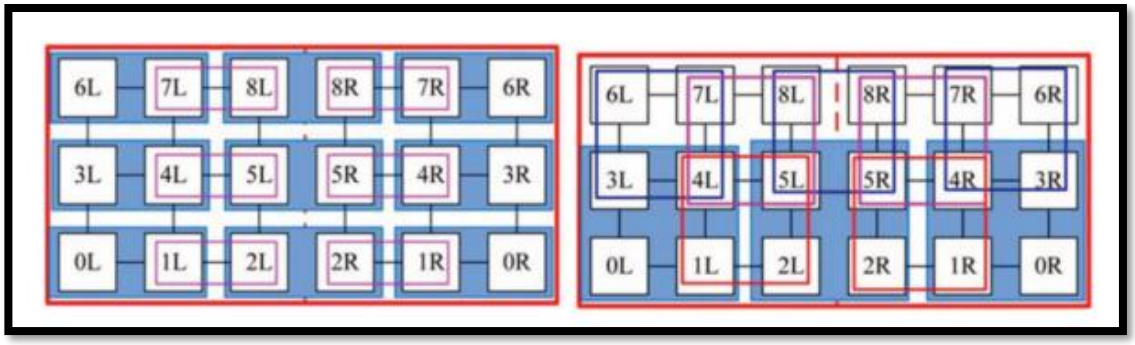
Figure 3.31: : Demonstrates the two Direct Division methods: (a) Wide Direct Division (WDD); (b) Narrow Direct Division (NDD).



Figure 332: Demonstrates the Two Inside Division method (a) Symmetrical Inside Division (SID); (b) Full Inside Division (FID).



Figure 3.33: Demonstrates Different patterns of block combination: (a) Combining every two (1 × 2) adjacent blocks; (b) Combining every four (2 × 2) adjacent blocks .



The global representation's size. Where Nbins in the block descriptor are the number of HOG bins. On the other hand, Block-wise grouping, in which descriptors are simply grouped together block by block, is used for key-point - local features such as SURF and SIFT. Thus, m · n groups of descriptors of the local area are collected and a classifier for each block is subsequently used for training (i.e., the m · n group is achieved).

Grid-based representations restrictions

The proposed representations of vehicle detection based on the grid, resumed as follows, have several limitations:

• The WDD approach takes into account the hood region where vehicle detection is subject to little or no discrimination.

• A fixed width-height ratio of ground truth ROIs is employed for the NDD method to establish the top limits of a rv.

• The established ratio may not be applicable to all MMs as there may be different ROIs for some models .

• The SID method only takes the left (or right) half of any Rv, according to which it estimates the other half. If the chosen half is in the occlusion position, the estimated half represents the occlusive features of the object and thus the MMR system fails.

In the case of occlusions, the FID method is also very susceptible to failures. The local characteristics of a block that is above the occlusive object could confuse the classification of the block.

• If the pitch angle of an incoming car is different from the one in the training period, the resulting face-views, Rv's and m's are different and may degrade accuracy of the obstacle as vehicle mode recognition.

3.4.4.2 Mid-Level Representations

The densely extracted local features of an image or ROI are coded into one vector representation using the Mid-Level Representation (MLR technique). For example, Locality Sensitive Linear Coding (LLC) vector, Vector of Locally Added Descriptors (VLAD), Super Vector Coding (SVC), and Fisher Vector (FV) include such encoding methods for MLRs. Here we describe its MLR method and discuss various inconveniences and limitations.

3.4.4.2.1 Description

The FV-based image representation is made using the approach for image classification problems developed by (Perronnin et al. 2010). Local descriptors are collected from several training pictures and a Gaussian Mixture Model is constructed using a pre-specified NG. The key points for a given picture on the basis of the Gaussian Detector Difference (DoG) are extracted (D. G. Lowe et al 2004). Then, densely sampled function descriptors are extracted from a patch around each key point (such as SIFT or PCA-SIFT (J. Sivic et al 2003). The average differences between the densely sampled descriptors and GMM centers of the first and second order are calculated in order to create a key point-based patch FV representation. Let f1, ..., fK be a series of local feature descriptors, which are densely sampled, and extracted from an image-based patch. The difference between the function descriptors and GMM centers in the first and second order is then:

$$\alpha_g = \frac{1}{K\sqrt{w_g}} \sum_{j=1}^{K} s_{gj} \Sigma_g^{-1/2} (f_j - \mu_g) \tag{31}$$

$$\beta_g = \frac{1}{K\sqrt{2w_g}} \sum_{j=1}^{K} s_{gj} \left[(f_j - \mu_g) \Sigma_g^{-1} (f_j - \mu_g) - 1 \right] \tag{32}$$

In the sgj weight of the soft assignment jth functional descriptor (j = 1, ..., K), the diagonal covariant matrices of the GMM are Gaussian (g = 1, ..., NG, etc.). All αs and β's computed are then concatenated for the key point-base patch to build the Fish vector F:

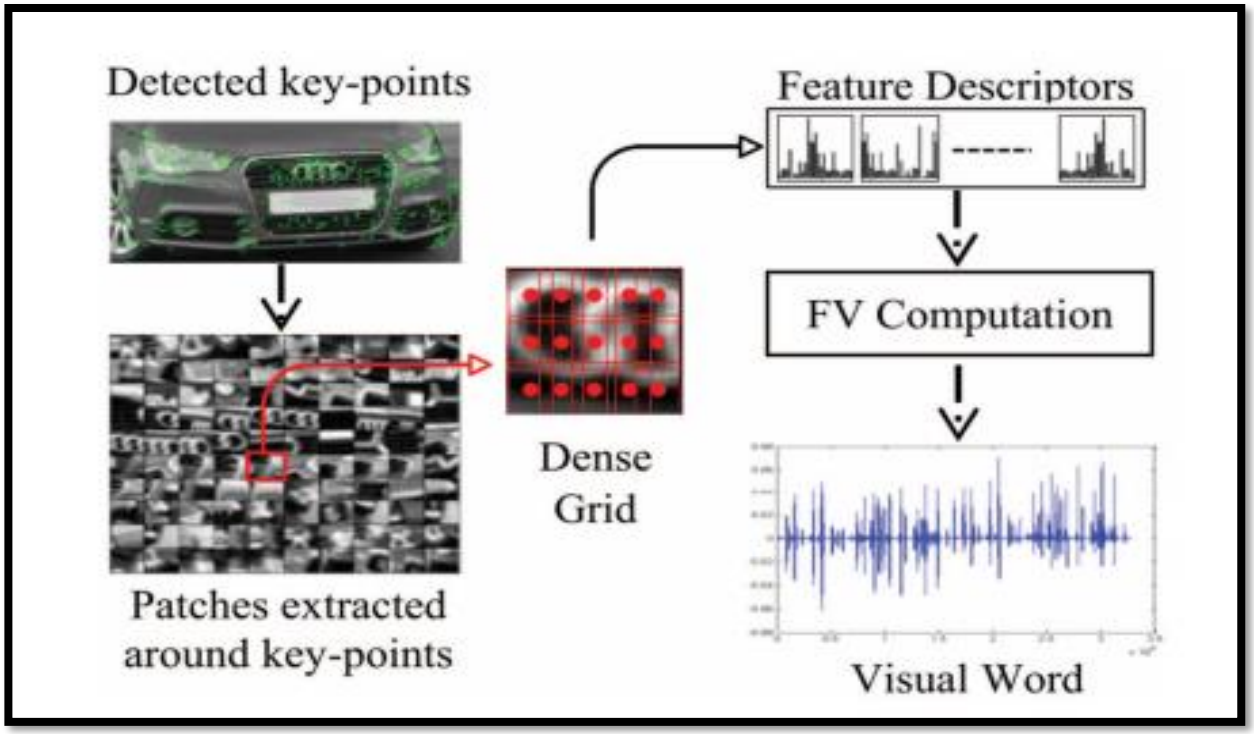
Each key point - based patch therefore gives an FV descriptor known as a visual word size 2 · NG · d, where d is the fj dimension. Figure 3.34 shows the flowchart by means of an FV descriptor computation to produce a visual word for an important point - based

patch. Then all of the FV-based visual words of the image are simply the global representation of an image. If a picture I is made of Npi key points, its overall representation is size 2 · NG · d · Npi. Up to 300 eyepoints - patches based on 128 or 256 images in (M. Fraz and al 2014) were obtained. That means that the overall picture would be sized 2 · 256 · 64 300 = 9830400 with the GMM of NG = 256 and the PCA-SIFT of d = 64 (very high dimensional). All training pictures of a class c where (c = 1, ..., Nc) are combined in the visual word into a sublexicon Sc. Each Sc is of dimensions (PMc I N pi) up to Mc, where Mc is the number of training c pictures, and Pi N pi is the total number in those Mc pictures of key points. In a complete L lexicon, which is an over-complete description area of all classes, the sub-lexicons of all Nc classes are collected.

$$L = [S\ 1, S\ 2, ..., S\ Nc] \quad (33)$$

The visual words for the key — based patches from the ROI vehicle in the sample image are calculated during the test phase. On the basis of visual word matching, the score for a probe image is calculated with each class sub lexicon. A distance-based match is made between the visual words and the visual words of each sub lexicon S c in Euclidean. The parallel score of a visual word and the closest S c word is the parallel score of the visual word with S c. The sum of the similarity values of all visual words to S c is taken from the similarity of the image of the sample in class c. The sample picture is given the highest similarity score class.

Figure 3.34: Demonstrates the Fisher Vector A key Point-based patch descriptor computing pipeline.



a)

3.4.4.2.3 Restrictions of this FV-based MLR technique

The major restrictions in the FV-based MLR technique are as follows:

• The visual words are high dimensionally, due to the dense samples that, in turn, lead to very large, global representation. The description of each patch is very complex and decreases its processing speed. As a result, computational complication complexity and memory requirements are increased and machine-based classifiers such as SVM are difficult to training. All key point-based patches are considered when forming sub lexicons, irrespective of their discriminating and representative abilities, which may cause confusion in the classifier. Occlusions could degrade the performance, as the key point - based patches from an occluding object will also be considered in matching

3.4.4.3 Local Features Concatenation

Many works use a simple method of concatenation of local features to create global image representations. These include mostly works using local features extracted from edges or gradients. In order to construct its global representations a concatenation pixel-wise is made of the ROI vehicle.

Figure 3.35 illustrates, for example, the various image characteristics and global representations used in vehicle making and vehicle models (Petrovic et al 2004). From a top line to the right, the first picture consists of the input ROI, followed by the corner reactions from Harris, the spectrum phase, Sobel vertical edge reactions and the horizontal edge reactions from the Sobel. At the bottom left there is a vector of weight for horizontal gradient responses in Square-Mapped gradients, vertical and horizontal components and vertical and horizontal components of Square-Mapped gradients. These numerical feature vectors, based on simple pixel concatenation, are considered as fixed length vectors.

The images from left-to-left column are raw vehicle ROIs, standardized ROI lighting, multi-scale Retinex (MSR) ROI results and finally the standardized functions of ROIs. The standardized features of the entire ROI are simply linked in one vector.

Figure 3.35: Demonstrates the Cootes and models make Petrovic and Petrovic and Cootes features extraction and global representatives of vehicles.

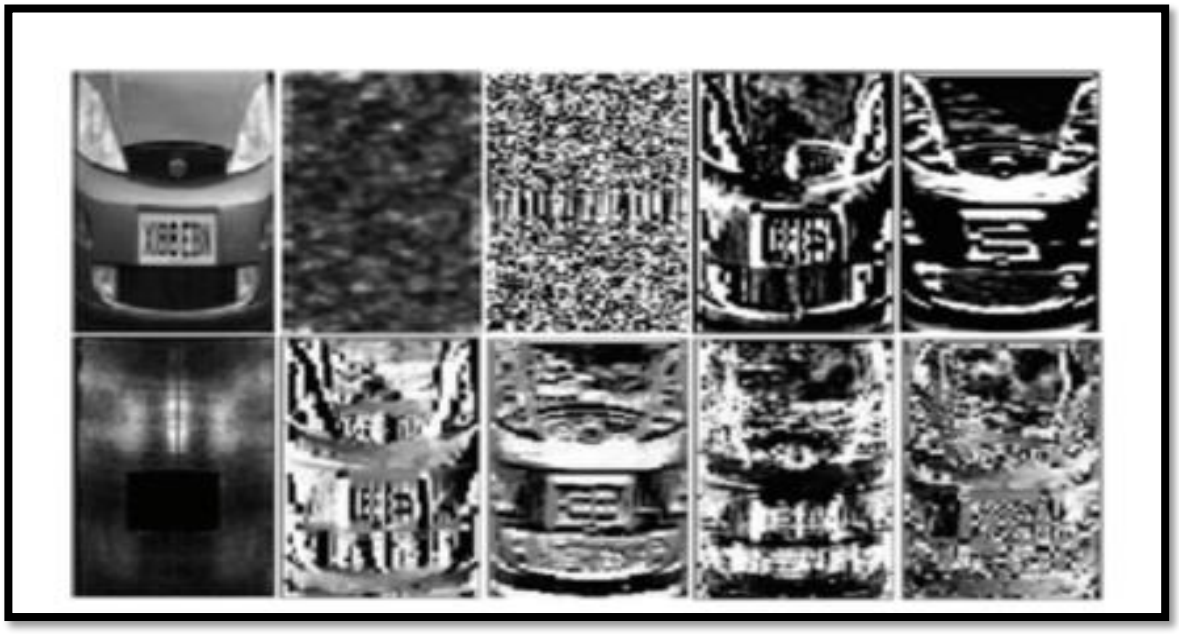
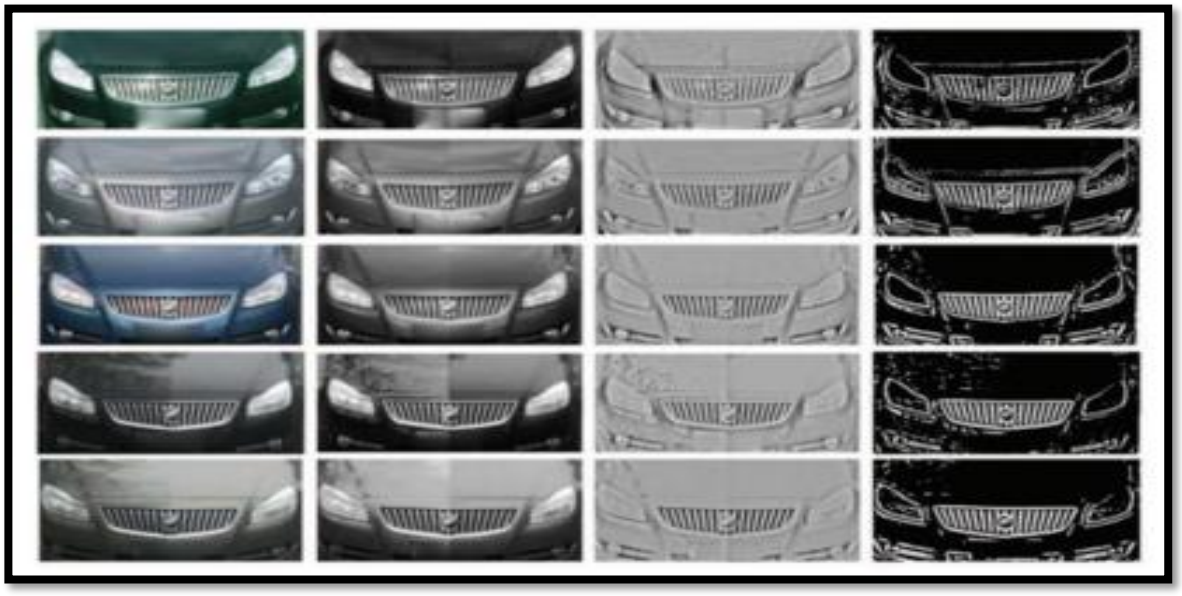


Figure 3.36: Demonstrates the Feature's extraction and global representations of obstacle as vehicle



3.4.5 Approaches proposed by BoF-ODMS

The overview of the BoF-ODMS approaches proposed. Besides the Extraction Features, the proposed BoF approaches to ODMS involve three main steps as demonstrated by Figure 3.37:

(1) Building Of Offline Dictionary

(2) Global Features Representation

(3) Classification.

In this work, In real-time ODMS, we are investigating two dictionary building schemes:

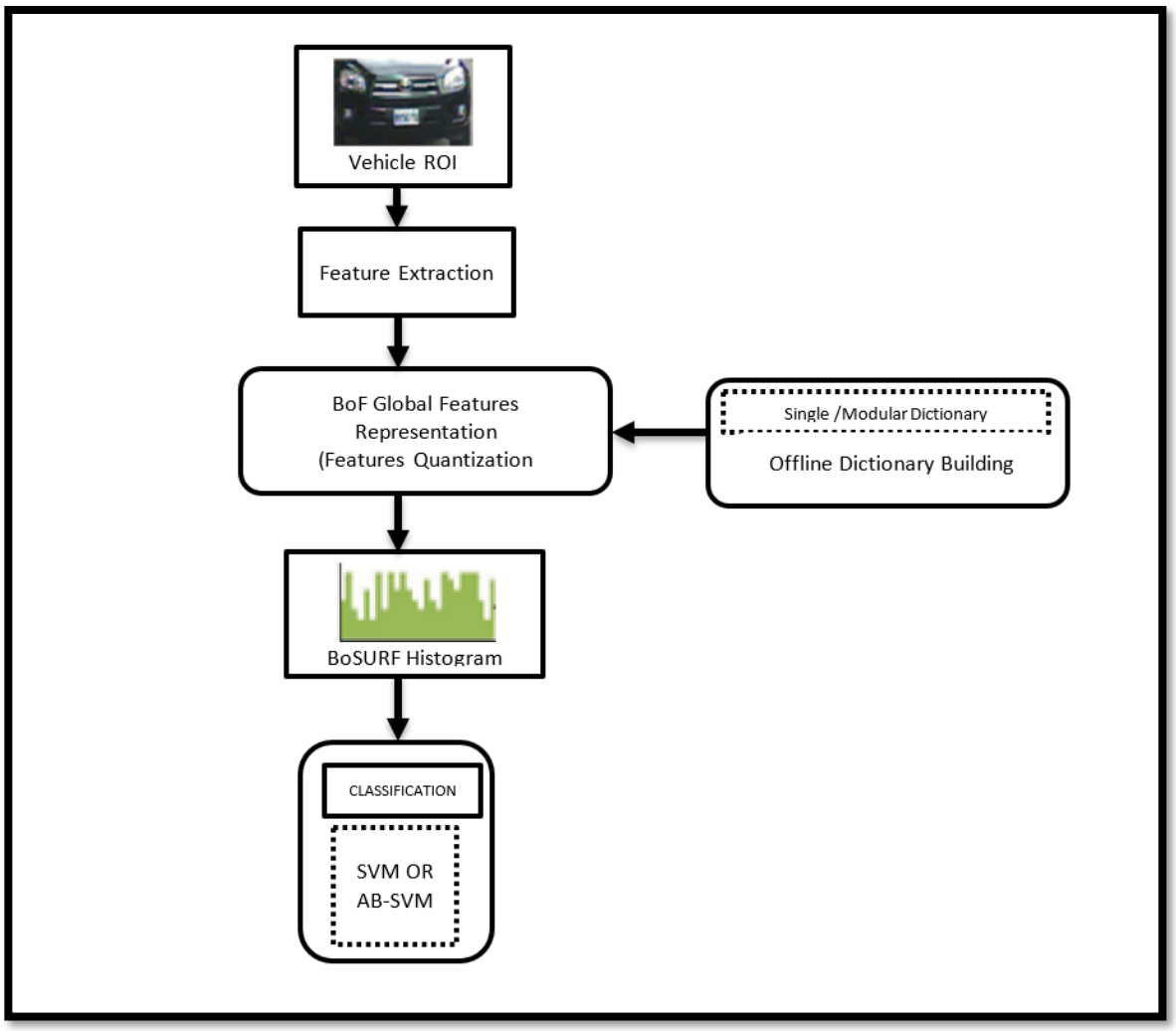
(1) Single Dictionary (SD)

(2) Modular Dictionary (MD).

The SD bases on the standard dictionary construction method in the BoF framework in which dictionary codewords are learned in all combined classes from the group of data (i.e., local image features). The MD, however, consists, each with a model class, of many individual dictionaries. Each sub dictionary is coded by the training data of the respective

model class. For building the BoF representations of the ROI, SD or MD vehicle local features (e.g., SURF).

Figure 3.37: Demonstrates Overview of the BoF-based ODMS approaches proposed.



3.4.5.1 Features Extraction for Obstacle Detection Model System

SURF became popular with multiple computer vision applications for features extraction. In relation to object recognition, image classification, etc., it has been demonstrated that there is higher accuracy and speed in comparison with other descriptors. Both the offline and global characteristics structure rely on the features of local images, for example SURF. In fact, any other good function descriptor in our BoF ODMS approaches can easily replace SURF. We provided a brief description of the way in which SURF is extracted and represented for completeness.

3.4.5.2 Offline Dictionary Building for ODMS

Each class pictures extract their local features (e.g., SURF). The main functions are kept in a bag or in a dictionary (codewords). The entire face appearance is recorded and described with the construction dictionary for each vehicle model. The dictionary is a compact representation of the main characteristics of the trainings of all classes. As a text document contains textual dictionary terms, an image consists of a visual codeword for the dictionary. Vehicle pictures are depicted as BoF functionalities, which are histograms of dictionary codes. Usually, the dictionary is made offline and only used for training and testing if necessary. This section describes the proposed and investigated single and modular dictionary systems. An overview of the systems SD and MD is presented in Figures 3.38a and 3.38 b.

The Pseudocode for the Offline Dictionary Building step is given in Algorithm 1.

Let I represent training pictures for NC number of classes in Equation 3.7,

Where Ii represents the set of class I images used in the dataset.

$$I = \{I1, I2, \ldots, INc\} \tag{34}$$

We bring the Fj's set of local features from every j-th image in Ii from the Equation 3.8, where fp is pth local in image, and Npj is the number of local features from the j-th image.

$$F_j = \{f_p | p = 1, \ldots, N\ p_j\ \} \tag{35}$$

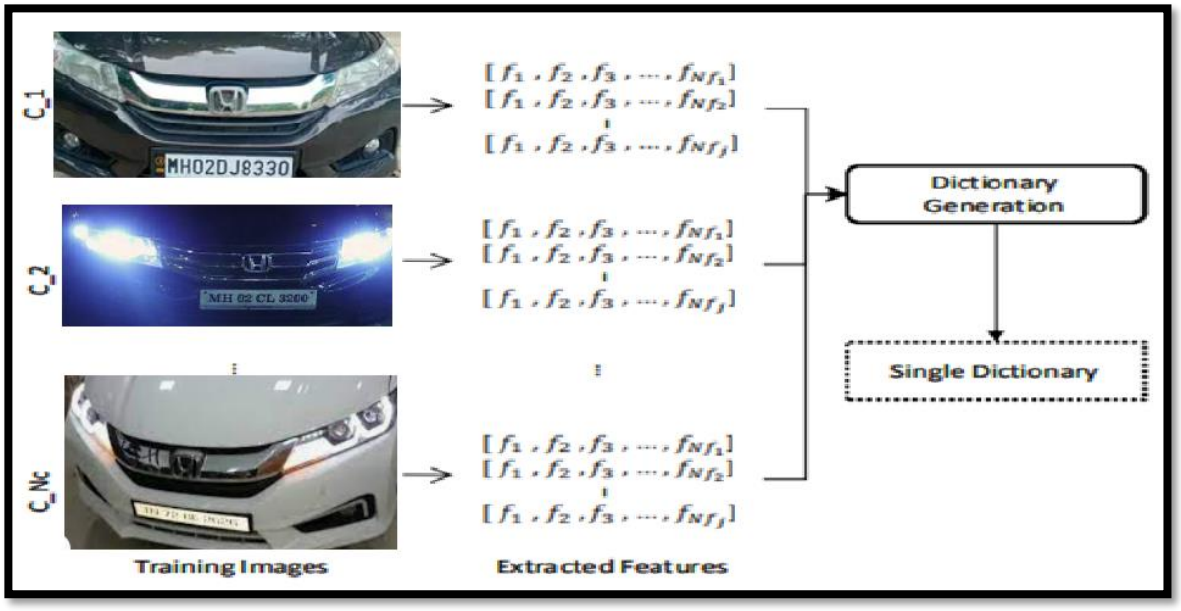
The set with all the features of Ni class I images is

$$F_i = \{F_j | j = 1, 2, ..., N_i\} \quad (36)$$

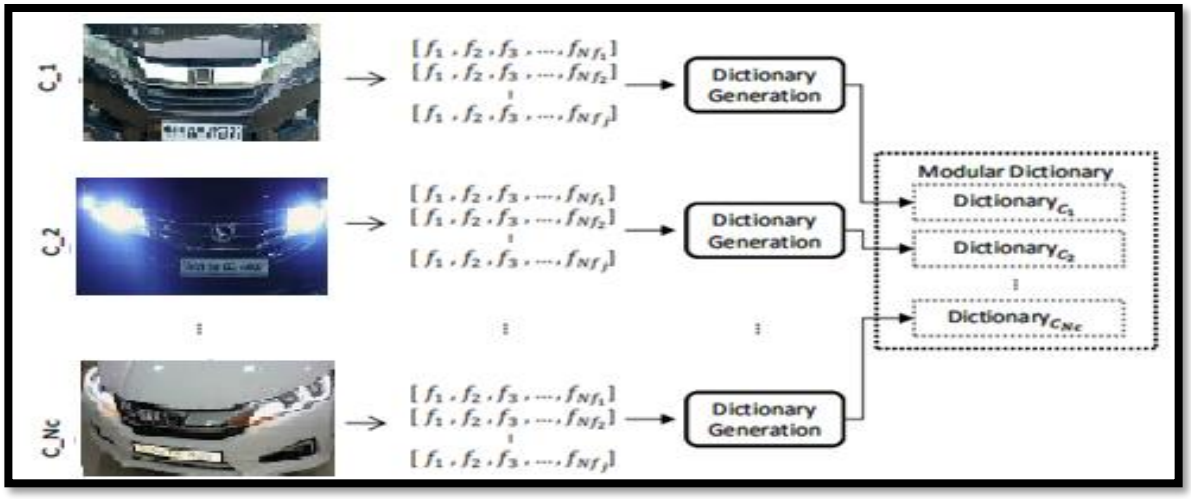
F in Equation 37 shows the pool of features from images of all classes:

$$F = \{F_i | i = 1, 2, ..., Nc\} \quad (37)$$

Figure 3.38: Demonstrates the Offline Dictionary Building in our BoF-ODMS approaches: (a) Single-Dictionary Building Scheme, (b) Modular Dictionary Building Scheme



(a)



(b)

3.4.5.2.1 Generation of Dictionary

Each fx functional descriptor can be seen in the function space as a point (e.g., for SURF, it would be a 64-dimensional space). The dictionary generation process consists of grouping all these training data points into a certain number of clusters in large spaces. We use the K-means technique to initialize the clusters and select the original set of cluster centers. The clustering technique k-means is then applied to the clustering. First, its nearest 100 neighbors can be found from data points for each cluster center. If there was a database It is only assigned to the nearest center near more than one cluster center. Each cluster center is subsequently updated to its average and its members (nearest neighbors). The new cluster center closest to each data point will be then re-grouped. This cluster centers update process, followed by the reaggregation of data points, is repeated several times, say 1000, in order to guarantee clusters stability. The cluster centers are stored in the corresponding dictionary as codewords.

3.4.5.2.2 Building of Single Dictionary (SD)

The main functionality for the build of the SD (D) is to be used through the generation of Dictionary by means of a combined set of training features of all classes, i.e., F of the equation 3.10. The total size of the SCD dictionary (S D) is determined by the number of clusters (or codewords). We call the BoF approach based on the BoF-SD system.

$$D = \{cw_k | k = 1, ..., S\ D\} \quad (38)$$

The reason behind the dictionary is to obtain the main characteristics of multiple and ambiguous vehicle manufacturing and models through the combined set of featured features from every class. This helps build global representations which are more discriminatory.

3.4.5.2.3 Building of Modular Dictionary (MD)

In this second scheme we create the main dictionary (denoted by DM) by combining the different dictionaries of each class, motivated by the results of each class (L. Hazelhoff et al 2014). This scheme is based on intuition, in the single dictionary schemes. Especially with a modular dictionary, the time consumed in dictionary construction is significantly reduced and flexibility is also provided. When classes have to be added (removed), the appropriate dictionaries can be flexibly attached to (the main dictionary) without requiring the whole dictionary to be reconstructed, thereby saving significant time. The Modular Dictionary (MD) consists of:

$$D_M = \{D_i | i = 1, 2, ..., Nc\} \quad (39)$$

$$D_M = \{cw_k | k = 1, ...,(S_{ID} \cdot Nc)\} \quad (40)$$

Where each di is the single class I dictionary, constructed using the same clustering process as mentioned by Section 3.4.4.2.1 to retain S ID key features (codewörter). The size and the cwk number of the overall DM dictionary is S D = S ID · Nc. BoF-ODMS approaches are known as BoF-MD based on the MD scheme.

3.4.5.2. Consideration of Size of the Dictionary

The dictionary size (S D) is a key parameter which affects how quickly processes, discriminatory capacity, and generally use the built-in dictionary. Due to the reduced discriminatory capacity, a small dictionary could suffer. More than one feature could, although it was different in small dictionaries, be allocated to the same cluster. On the other hand, a large dictionary loses generalization capability, increases noise penalties and

overhead processing (Y.-G. Jiang et al 2007). As part of this research, we study the impact on total ODMS speed and accuracy of the various dictionary sizes (both for SD and MD schemes)

3.4.5.3 Feature Descriptor GIST

People can classify with a glance an object or a scene without very quickly looking at the details in a picture. We can, for example, immediately recognize the scene without thinking about the details or the existence of other objects after seeing an image of large buildings, oceans or trees. The scene GIST refers to the information in a glance (around 200 milliseconds). GIST's first (Oliva et al 2001) feature descriptor is a "low dimension scene representation without any form of segmentation." As a single object, the descriptor focuses on the shape of a whole frame and computes the feature vector. For the scene classification the GIST descriptor was initially suggested. SIFT and SURF concentrate on individual highlights and the HOG feature descriptor is calculated on the basis of a single window (patches) and then a concatenated one. GIST descriptor concentrates on the format of an entire image as one object and computes the vector feature. The GIST descriptor does not know the presence and relationships of local objects. Thus, GIST offers a holistic image of a scene. Three steps can be taken to obtain the GIST feature descriptor algorithm:

1.Initial transformation of the original picture into m-scales and n-orientations. Then all these mxn images are fitted with a set of Gabor filters. The Gabor filter examines the presence in any particular direction in an image of potential specific frequency content. The output dimensions of the Gabor filter are identical to the input image; therefore, mxn maps with the same size as the input image are generated.

2.Each characteristic map is divided into a multi-block grid. The characteristic map values are averaged in each block. Now we have several average values for each map. The division into the grid of the characteristics and the average values for each block in the grid reduces the dimensions of the characteristic map.

3. Finally, the GIST descriptor or image feature vector is concatenated by an average of all blocks of all feature maps.

4For all images regardless of their image resolution, the GIST algorithm produces the feature descriptor of the same size. We have created a four-scale GIST description and 8-

screens, creating 32 images and dividing feature maps into a 4x4 or 16-scale grid that generated a 512-dimensional vector (16 averaged values x 32 feature maps).

3.4.5.4 BoF-ODMS Global Features Representation

The second phase of the proposed BoF-ODMS approach is to integrate local images into global BoF representations using the dictionary D or DM (constructed in Offline Dictionary Building). For a particular image of Ij of class I its BoF is a normalized histogram Hi J, of voting in the dictionary codewords (by its local features). You can show the histogram Hi j by:

$H_{ij} = [h_1, h_2, ..., h_{S\|D}]$	(41)

where the bins (hk's) hold the number of votes to the respective codewords (cwk's), respectively (k = 1, ..., S D). The next dictionary codeword adds a unit vote to each feature match. For the purposes of creating the BoF features, the image Ij's histogram is matched with the closest cwk codeword (D or DM) of each local function fp to the fp to Ij FIj. This increases histogram bin hk's voting count. This step is also known as quantization of characteristics. The histogram undergoes a normalization procedure after all the FIj features have been quantized. This way, after matching every feature of a given image, we get the final histogram and call it BoF histogram or feature. Algorithm defines how the global BoF representation histograms will be generated for a particular image. For the set of local features, the BoF histogram Fj of a specific image Ij of class I is calculated as follows:

(42)

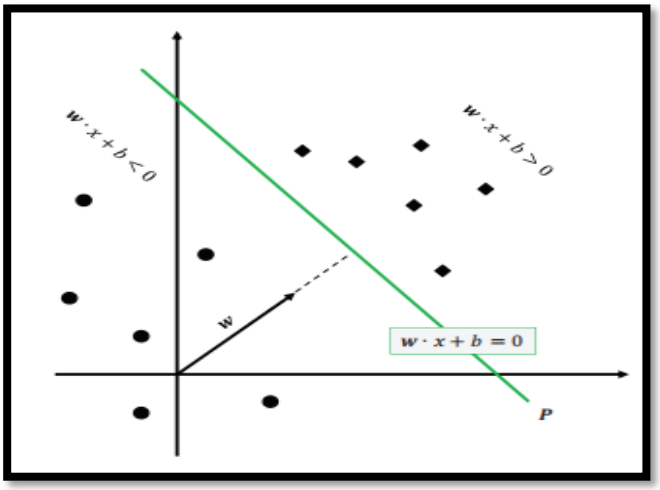
$$H_{ij}(k) = \frac{1}{Nf_j} \sum_{p=1}^{Npj} \begin{cases} 1 & \text{if } k = \underset{t \in [1, S_D]}{\operatorname{argmin}} dist(cw_t, f_p) \\ 0 & \text{otherwise} \end{cases}$$

Where dist(a,b) is the Euclidean distance from characteristics of the Ij, and where j(k) is hk; fp is the local pth feature, N fj is the total number of local characteristics from the Ij image.

3.4.6 Classification Techniques

This section contains a brief description of the techniques of classification used in state-of-the-art ODMS work. Here we present support for vector machines as well as forestry-based classifiers such as Random Forests and Sparse Materials.

Figure 3.39: Demonstrates A hyperplane P which divides two class 2D data points. The circles are equivalent to class L1 data points, while the diamonds are equivalent to those of label +1.



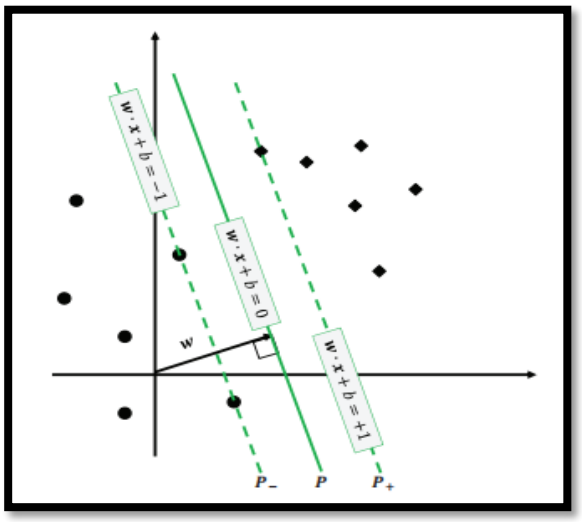
3.4.6.1 Support Vector Machines

The support vector machine (Burges et al 1988) (Vapnik et al 1995) is a highly effective binary classifier where the support vectors (SVs) represent the best separation between two classes in a subset of training data. Based on its distance from these support vectors, a data sample is then classified. To build an SVM classification, a multi-class set of many such binary classifiers can be used.

3.4.6.1.1 Binary SVM Classifiers

Consider a N linearly separable d-dimensional data observation training set (Xi, Yi), which includes Xi¨ ∈ Rd and labels yi¨ ∈{~1, +1}. A Hyperplane P, also known as the decision boundary, can be defined as w· x + b = 0, where the vector w is normal to P, b is bias and where the perpendicular distance from origin to P is |b/|k|. For a picture of this data separation by P, see Figure 3.40. The support vectors are all data points of the label +1 and d1 at the shorter distances d+ and d to P, respectively (SVs). Then the margin around P is d+ + d- wide. In the case of linear separation of the training data considered, the following two inequalities can be assumed to be met by all data points (xi),

Figure 3.40: Demonstrates A the hyperplanes P_, P, and P+.



$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1, \; y_i = 1 \tag{43}$$

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq +1, \; y_i = +1 \tag{44}$$

Joining the inequalities of Equations 4.1 and 4.2, we get:

$$\mathbf{yi} \cdot (\mathbf{xi} \cdot \mathbf{w} + \mathbf{b}) \, 1 \geq 0, \, 1 \leq \mathbf{i} \leq \mathbf{N} \tag{45}$$

The SVs of +1 and d-1 are included in the hyperplanes P+1 and P-1, and with w·x+b =±1 respectively Therefore, the distances perpendicular to P+1 or P-1 are |−1-b|/||w|| and |1-b|/||w||,which means that the margin is P+1 and P- 1 is 2/||w||.

The Lagrangian Formulation

The hyperplanes P+1 and P 1 can be chosen to maximize the margin width of 2/||w||, which corresponds to the minimum ||2|| achieved through quadratic programming, for optimizing the classification performance.

The minimization problem can be defined as a Lagrangian formulation L:

$$L = \frac{\|\mathbf{w}\|^2}{2} - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1] \tag{46}$$

where αi ≥ 0. Where αi is 0. Where αi is 0. The inequality of Equation 46 is replaced by a more easily handled Lagrangian multipliers αi.

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Leftrightarrow \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x_i} = 0, \quad \forall i \tag{47}$$

$$\frac{\partial L}{\partial b} = 0 \Leftrightarrow \sum_{i=1}^{N} \alpha_i y_i = 0, \quad \forall i \tag{48}$$

$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 \geq 0, \quad \forall i \tag{49}$$

$$\alpha_i \geq 0, \quad \forall i \tag{50}$$

$$\alpha_i(y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1) \geq 0, \quad \forall i \tag{51}$$

Substituting the Equations 49 and 50 into Equation 52, we get the dual formulation LD, which is to be maximized with respect to α.

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \tag{52}$$

Having found the optimal αi's from solving Equation (52), the optimal w and b can be then found using:

$$\mathbf{w} = \sum_{i=1}^{N} y_i \alpha_i \mathbf{x_i} \tag{53}$$

$$b = y_k - \mathbf{w} \cdot \mathbf{x_k}, \quad \alpha_k > 0 \tag{54}$$

where $\alpha_k > 0$ for the support vectors x_k. The classification function f(x) for a new test sample x_t is defined as:

$$f(\mathbf{x_t}) = sign(\mathbf{w} \cdot \mathbf{x_t} + b) = sign\left(\sum_{k=1}^{m} y_k \alpha_k (\mathbf{x_k} \cdot \mathbf{x_t}) + b \right) \tag{55}$$

In this way, to determine the label of an unknown data sample vector xt , we need only the support vectors, x_1, ..., x_m

Linearly Inseparable Data

The limitations are therefore loosened by means of a slack variable: 0 or 1 to be tolerated by the separating margin. To be able to tolerate samples The limitations contained in Equation 56 are: So, the constraints are relaxed using a slack variable $\xi \geq 0$, $1 \leq i \leq N$, to tolerate the samples within the separating margin. The constraints in Equation 57 are relaxed to:

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1 \geq -\xi_i \tag{56}$$

which translates into:

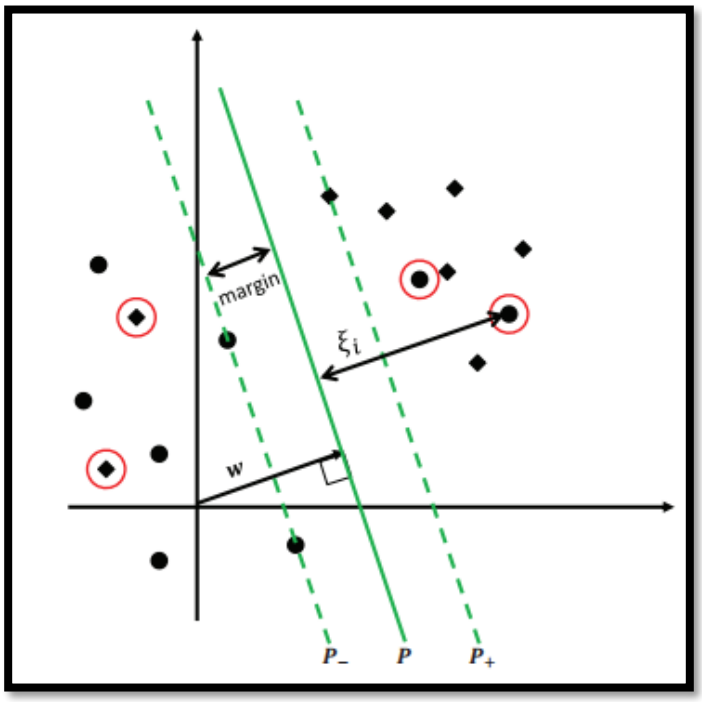
$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 - \xi_i, \quad y_i = +1 \tag{57}$$

$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 + \xi_i, \quad y_i = -1 \tag{58}$$

A data point xi on the wrong side of P is considered an error, when ξi exceeds 1. The upper bound for the training errors is ∑∀i ξi . The use of the slack variables is shown in

Figure 3.41.The primary problems of (and dual) optimization are then changed to accommodate the noisy data points in two distinct ways. (1) C-Classification of Vector support and (2) v-Support Vector Classification In addition, we compare BoF-ODMS' performance to C-SVC and v-SVC, in order to find the best multi-class SVM for ODMS purposes.

Figure 3.41: Demonstrating the use of slack variables, in case of linearly inseparable data



Machinery Nonlinear Support

In order to find a separator for linear, inseparable data the linear classifications can be generalized. This is done by matching the data with the nonlinear function — kit to another Hilbert space. The function A essentially maps the training data points to a higher-dimensional feature area, which then learns about the separating hyperplanes. The linear classification of data points in H equals the linear classification of the data points in IRd. The example of linearly inseparable IR2 data mapped on to IR3 space, where it is linearly separable, is shown in Figure 3.41. The dot product of the data points xi · xj used in training is replaced by Φ(xi) · Φ(xj). However, due to the high H dimension, this makes it difficult to deal with. The adjusted dot product is replaced with the kernel function K in order to avoid explicit è calculation.

$$K(\mathbf{x_i}, \mathbf{x_j}) = \Phi(\mathbf{x_i}) \cdot \Phi(\mathbf{x_j}) \tag{59}$$

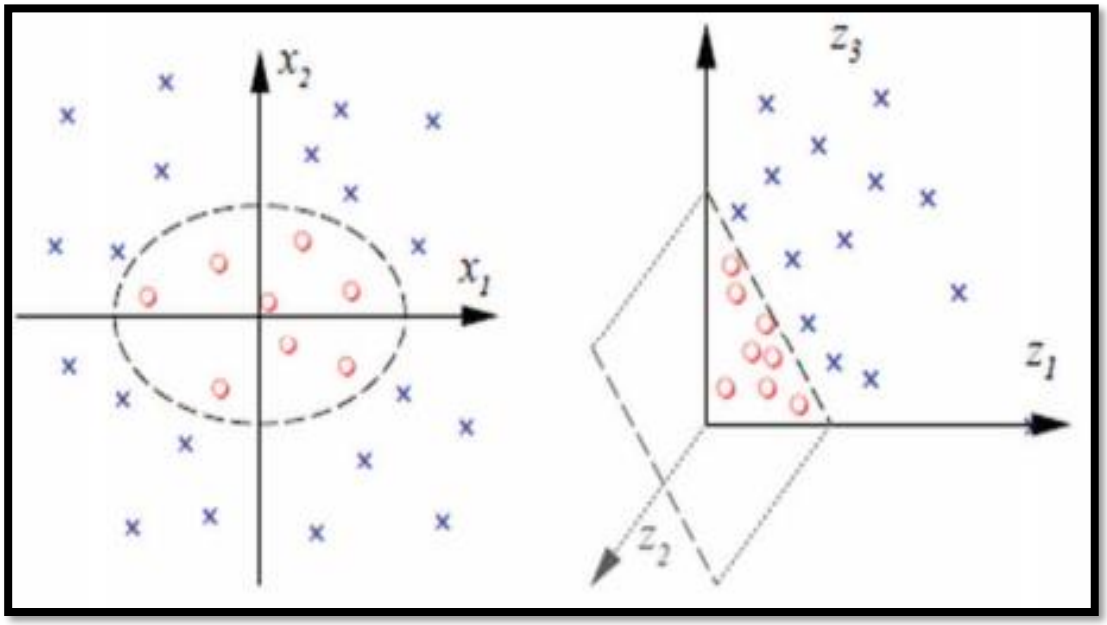
The kernel function K is symmetric:

$$K(\mathbf{x_i}, \mathbf{x_j}) = K(\mathbf{x_j}, \mathbf{x_i}) \tag{60}$$

and satisfies the Mercer criteria

$$\int K(\mathbf{x_i}, \mathbf{x_j}) g(\mathbf{x_i}) g(\mathbf{x_j}) d\mathbf{x_i} d\mathbf{x_j} \geq 0 \tag{61}$$

Figure 3.42: Demonstrates in IR2 (left), mapped to IR3 (Right), the linearly inseparable data points allow to learn the separation hyperplane.



Here, $(x_1, x_2) \rightarrow (z_1, z_2, z_3)$, where $z_1 = x_{21}$, $z_2 = \sqrt{2}x_1 x_2$, $z_3 = x_{22}$. (Weston et al 2015) for any function g, having a finite value of R g(x)2dx. This criteria ensures that the kernel matrix K(i, j) is positive semi-definite. For more details on the characteristics of kernel functions and their

construction, readers are referred to (Genton et al 2002). Here, we give a few examples of frequently used kernels:

• Linear Kernel

$$K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i} \cdot \mathbf{x_j} \tag{62}$$

• Quadratic Kernel

$$K(\mathbf{x_i}, \mathbf{x_j}) = (\gamma \mathbf{x_i} \cdot \mathbf{x_j} + r)^2, \ \gamma \in \mathbb{R}^+, r \in \mathbb{R} \tag{63}$$

• Polynomial Kernel

$$K(\mathbf{x_i}, \mathbf{x_j}) = (\gamma \mathbf{x_i} \cdot \mathbf{x_j} + r)^d, \ \gamma \in \mathbb{R}^+, r \in \mathbb{R}, d \in \mathbb{N} \tag{64}$$

• Radial Basis Kernel

$$K(\mathbf{x_i}, \mathbf{x_j}) = e^{-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2} \tag{65}$$

For classification, the function defined in

$$\begin{aligned} f(\mathbf{x}) &= sign(\mathbf{w} \cdot \mathbf{x} + b) \\ &= sign\left(\sum_{i=1}^{m} y_i \alpha_i (\Phi(\mathbf{x_i}) \cdot \Phi(\mathbf{x})) + b\right) \\ &= sign\left(\sum_{i=1}^{m} y_i \alpha_i K(\mathbf{x_i}, \mathbf{x}) + b\right) \end{aligned} \tag{66}$$

where x_i (i = 1, ...m) are the SVs.

3.4.6.1.2 C-Support Vector Classification

In SVM (Cortes et al 1995, Boseret al. 1995), the primary problem to be resolved in the C-Support Vector Classification (C-SVC) formula is:

$$\min_{\mathbf{w},b,\xi} \ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}\xi_i \tag{67}$$

with the following constraints:

$$y_i(\mathbf{w}^T\Phi(\mathbf{x_i}) + b) \geq 1 - \xi_i, \forall i \tag{68}$$

$$\xi_i \geq 0, \forall i \tag{69}$$

where C > 0 is the regularization parameter. In practice, to avoid the high dimensionality of the vector variable w, the following dual problem is solved instead.

$$\min_{\alpha} \ \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} \tag{70}$$

with the following constraints:

$$\mathbf{y}^T\boldsymbol{\alpha} = 0 \Rightarrow \sum_{i}^{m}\alpha_i \cdot y_i = 0, \forall i \tag{71}$$

$$0 \leq \alpha_i \leq C, \forall i \tag{72}$$

where Q is an N × N positive semi-definite matrix, $Q_{ij} = y_i y_j K(x_i, x_j)$, e = [1, . . . , 1]T is a vector of all ones, and $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$ is the kernel function

$$\mathbf{w} = \sum_{i=1}^{N} y_i\alpha_i\Phi(\mathbf{x_i}) \tag{73}$$

Then, for classification, the decision function is similar to Equation 4.24:

$$\begin{aligned} f(\mathbf{x}) &= sign(\mathbf{w}^T\Phi(\mathbf{x}) + b) \\ &= sign\left(\sum_{i=1}^{m} y_i\alpha_i K(\mathbf{x_i}, \mathbf{x}) + b\right) \end{aligned} \tag{74}$$

where x_i (i = 1, ...m) are the SVs.

3.4.6.1.3 v-Support Vector Classification

In the ν-Support Vector Classification (ν-SVC) formulation of SVM, (B. Scholkopf et al 2000) introduced the parameter ν ∈ (0, 1] in the optimization problem, to control the number of SVs. The ν gives a lower bound to the fraction of samples becoming SVs, and an upper bound on the fraction of training errors (i.e., the training samples lying on wrong side of the hyperplane).

The primal optimization problem is modified as:

$$\min_{\mathbf{w},b,\xi,\rho} \quad \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m}\xi_i \tag{75}$$

with the following constraints:

$$y_i(\mathbf{w}^T\Phi(\mathbf{x_i}) + b) \geq \rho - \xi_i, i = (1, \ldots, m) \tag{76}$$

$$\xi_i \geq 0, \rho \geq 0 \tag{77}$$

The dual problem of which is:

$$\min_{\alpha} \quad \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} \tag{78}$$

with the following constraints:

$$\mathbf{e}^T\boldsymbol{\alpha} \geq \nu \Rightarrow \sum_{i}^{m}\alpha_i \geq \nu \tag{79}$$

$$\mathbf{y}^T\boldsymbol{\alpha} = 0 \Rightarrow \sum_{i=1}^{m} y_i\alpha_i = 0, \tag{80}$$

$$0 \leq \alpha_i \leq \frac{1}{m}, \forall i \tag{81}$$

where Q and e are as described previously.

3.4.6.1.4 Multi-class SVM

A collection of many binary SVM classifiers are used to build a single multi-class SVM classifier. A collection of many binary SVM classifiers are used to build a single multi-class SVM classifier. Two popular ways of building the individual binary classifiers are 1-vs-1 or 1-vs-All. In the former approach, a binary SVM learns to differentiate between every two classes. So, there would be Nc (Nc-1)/2 binary SVMs in the multi-class SVM, where Nc is the number of classes in the dataset. In the latter approach on the other hand, a binary SVM is learned to differentiate each class against all other Nc-1 classes. Each SVM binary gives a class vote. The final classification output is the class that receives maximum votes in all binary SVMs. • **Majority Voting**: Each SVM binary gives a class vote. The final classification output is the class that receives maximum votes in all binary SVMs.

• **Last Square Error Outputs (LSE)-weighted:** Enhanced majority voting system by weighting binary SVM outputs.

• **Hierarchical dual-layer combination:** A second-layer classification evaluates first-layer classifying outputs.

3.4.6.2 Random Forests

The classification of the Random Forests is a group of random weak decision trees. Introducing Random Forests, Breiman introduced the algorithm (Breiman et al 2013). It combines Breimans idea of random subsets of training data (so-called "bagging") with the application of the Random Subspace Method in order to randomly subsample the characteristics of the decision trees, as laid down by (T. K. Ho et al 1995). In multi-class picture classification problems, the Random Forests have shown success. In Caltech-101 (Fei-Fei et al 2007) and Caltech-256, for example (Bosch et al.2007) have demonstrated that Random Forest has superior performance over SVM (Griffin et al 2007). In addition, Random Forest applied the problem of recognition of traffic signs (Zaklouta and Stanciulescu et al, 2012) and showed that the problem is above SVM. Some of the benefits of Random Forests include the ability of a parallel or distributed computer environment to be implemented and used for online learning. Random Forests were not explored with real-world datasets in the context of the ODMS, as far as we know. Section 6.4 examines the question of the use of a real-world dataset by Random Forest for ODMS problems, to respond to whether Random Forests are more successful than SVM in ODMS.

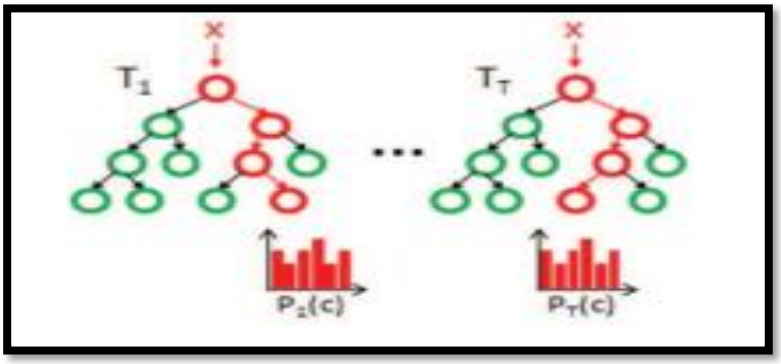
3.4.6.2.1 Random Forest Construction

A Random Forest R is composed of many Random Trees $T_t \in \{T1, T2, ..., TT\}$. Each random tree Tt is built according to the following procedure: From the set of all training data X containing Ntr number of samples, a subset X_i is randomly chosen (with replacement). Using this subset, the tree is grown further (without pruning) Within a tree, at each node, the algorithm randomly chooses a subset F of features. To split the X_i into X^l_i and X^r_i , the feature $f *j \in F$ with the maximum information gain 4 and a threshold $t \in [\min(f *j), \max(f *j)]$ are used.

$$X_i^r = \{j \in X_i | f_{jk} > t\}, \quad 1 \leq k \leq |F| \tag{82}$$

$$X_i^l = X_i \setminus X_i^r \tag{83}$$

Figure 3.43: Demonstrating the use the classification procedure for a text sample x with a Random Forest classifier. The final decision is based on combining posteriors ($P_t(c)$'s) from all T_t's. (Zaklouta and Stanciulescu et al 2012])



The gain of information is calculated using the following equation, which calculates entropy E on the basis of class frequencies in Xi:

$$\Delta = -\frac{|X_i^r|}{|X_i|}E(X_i^r) - \frac{|X_i^l|}{|X_i|}E(X_i^l) \quad (84)$$

The optimal fj to be divided is then found based on the maximum gain of information:

$$f_j^* = \max_j \Delta \quad (85)$$

The number of functions to choose (|F|) will depend on the dataset. With a large F, training speed decreases and overfitting problems can result. On the other hand, a small F increases the speed and strength of training. If F is too little, however, it can lead to problems that are not fitting.

3.4.6.2.1.1 Classification

Given a test sample x, each of the random trees T_t of the Random Forest R is traversed. In the lead node reached in T_t , the posterior probability that x belongs to the class c (where c = 1, ..., Nc) is given by $P_t(c|x)$. The final classification output c∗ is based on combining the decisions of all Tt's. Figure 3.42 illustrates the classification procedure

$$P(c|x) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|x) \quad (86)$$

$$c^* = \underset{c}{\operatorname{argmax}}\, P(c|x) \quad (87)$$

3.4.6.3 Sparse Representation-based Classification

Sparse representation (SR) has been a popular technique that has resulted in overwork in a statistical signal processing community that uses an over-complete dictionary of base elements and signal atoms to compute sparse linear representations of signals. These projects were aimed primarily at representing, compressing and not classifying high-dimensional signals. In recent years, new problem domains such as image classification/retrieval have been expanded to include theories and algorithms for SR. (Wright et al. 2010) have shown that SRs of images can encode and solve computer vision (CV) or image classification problems with content or semantic information. This gave

the CV community an interest and led to various techniques and frameworks for the image classification based on SR. SR has been successful in various image classification problems including: multi-modal biometrics recognition (Shekhar et al 2014) object recognition (Patel et al 2013) action recognition (Chiang et Al 2015) etc. (Chen et al. 2015) used the ODMS SR-based classification with encouraging precision, but slow processing speed. SR is used as an approach in the domain of CV and image classification to represent objects of interest through an overall dictionary that contains training images or global representations of their features.

3.4.6.3.1 Dictionary and Sparse Coefficient Learning

Assumed the training data x = [x1, . . ., xN] ∈ IR m×N, the area is to invention an optimal dictionary D = [d1, . . . , dK] ∈ IRm×K , such that an image's global feature representation vector xi can be represented as a sparse approximation over it. The xi could be approximated as a linear combination of few columns (atoms) from D, i.e., xi = Dαn, or, generalizing over the training data:

$$\mathbf{x} \approx \mathbf{D}\boldsymbol{\alpha} \tag{88}$$

where α = [α1, . . . , αN is a set of sparse decomposition combination coefficients. K-SVD (Aharon et al 2006) The optimal D and α algorithm is used to study. The following joint optimization issue can be formulated as this learning process:

$$\underset{\mathbf{D},\boldsymbol{\alpha}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \ \text{ subject to } \ \|\boldsymbol{\alpha}_i\|_0 \leq T, \forall i \tag{89}$$

where kα_i k$_0$ is the `0-norm, i.e., a count of non-zero values in α_i, T is the maximum number of coefficients other than zero desired. The sparse α solution can be achieved by solving the equation optimization Equation 90, by a greedy and iterative method, which finds optimum orthogonal matching (OMP) α_i for each x_i , while fixing D:

$$\boldsymbol{\alpha}_i = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\mathbf{x_i} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \ \text{ subject to } \ \|\boldsymbol{\alpha}_i\|_0 \leq T \tag{90}$$

There are two stages to the K-SVD algorithm: (1) sparse coding and (2) dictionary updates iterating to convergence several times to optimize D and α. Each αi in α is optimized in each iteration with the solution Equation 90 at the sparse coding phase of D fixation and the column

vectors in D are updated sequentially with a view to improving the x representation of its coefficients.

3.4.6.3.2 SR-based Classification

Given the optimal D achieved via K-SVD, an input vector scheme xt an IRm can be described in Algorithm 4 as the classification SR-based (SRC). The text as a linear combination of D atoms is approximated:

$$\mathbf{x_t} = \mathbf{D}\alpha_t = \alpha_1\mathbf{d_1} + \cdots + \cdots + \alpha_k\mathbf{d_K} \tag{91}$$

Where αk ∈ αt and dk ∈ D oscillating oscillations. By solving a problem of optimization with the second-order cone programming (Boukerche et al 2008) the sparse solution αt is found efficiently

$$\underset{\alpha_t}{\operatorname{argmin}} \|\alpha_t\|_1 \quad \text{subject to} \quad \|\mathbf{x_t} - \mathbf{D}\alpha_t\|_2^2 \leq \varepsilon \tag{92}$$

Where ε is the error tolerance for reconstruction. It is important to mention here that the time-consuming optimization process to find αt is a major drawback of the SRC scheme.

The Nc classes of dictionary D are: D= [D1,., ., DNc]. The input sample xt is approximated as the Dαt sample where τi: IRm→IRm function is to choose a group of coefficients of the class i. The xt residual is then calculated as ri(xt) in relation to class I i.e., as:

$$r_i(\mathbf{x_t}) = \|\mathbf{x_t} - \mathbf{D}\delta_i(\alpha_t)\|_2 \tag{93}$$

The classification feature will be shown below, where the result is the class to which the text has the minimum residual:

$$f(\mathbf{x_t}) = \underset{i}{\operatorname{argmin}}\, r_i(\mathbf{x_t}) \tag{94}$$

Algorithm 3.3 Algorithm for Sparse Representation-based Classification (SRC) Scheme

1: Input: The set optimal $D \in IR_{m \times K}$, a test feature vector $x_t \in IR_m$

2: Output: The predicted class label of x_t

3: Finding the sparse solution α_t , by solving Equation (92)

4: Computing the residuals (93)

5: return $f(x_t) = \text{argmin}_i\, r_i(x_t)$ (94)

3.4.6.3.3 Hamming Distance-based Classification

In order to avoid slow processing speed in the SRC system, Chen et al. proposed the Hamming Distance Classification System for ODMS on the basis of sparse representation basis (D). The premise is that if the test sample xt belongs to the class I then most of the non-zero values of the linear combination coefficients for the class I enterprises. The sparser the αt recovered is, the more feasible it is to classify text with accuracy. Given D = [D1, . ., DNc] the maximum distance of each di, by a metric of Euclidean distance, is found in each di. If dij is the dj vector of the jth column of class I, the maximum distance between samples is indicated by di:

$$\eta_i = \max_{1 \le j,k \le N_i} dist(\mathbf{d}_{i,j}, \mathbf{d}_{i,k}) \tag{95}$$

The Euclidean distance between vectors a and b is where dist(a, b).

In the context of the elements in D, Bi(xt) = [bi, 1,., bi., Ni], is determined by: the xt is encoded into the binary code B. (xt)= [B1(xt),.., BNc(xt)]

$$b_{i,j}(\mathbf{x_t}) = \begin{cases} 1 & \text{if } dist(\mathbf{x_t}, \mathbf{d}_{i,j}) \le \eta_i \\ 0 & \text{otherwise} \end{cases} \tag{96}$$

Let B^{o}_{i} be a binary code representation for elements of class i in D. Each $b^{o}i$, j corresponds to the element $d_{i,j} \in D_i$. The entries of B^{o}_{i} are non-zero (ones) only for b^{o}_{i}, j 's corresponding to $d_{i,j's}$, and zero otherwise: $B^{o}_{i} = [O_{1,1}, O_{1,2}, \ldots, O_{1,N1}, \ldots O_{i\,1,Ni-1}, 1_{i,1}, 1_{i,2}, \ldots, 1_{i,N}i, O_{i+1,1}, \ldots, O_{Nc,N}]$, where $N = \sum^{Nc}_{i} N_i$ is the total number of elements in D for all N_c classes.

The classification function for x_t is:

$$f(\mathbf{x_t}) = \underset{1 \le i \le Nc}{\operatorname{argmin}} \xi_i(\mathbf{x_t}) \tag{97}$$

When = HamDist(B(xt),Boi), that is the distance of Hamming between B(xt) and Boi binary vectors, that is to say, the distance of xt from the ith class. A threshold-based refusal option, as is the case is used to account for unknown classes. If ξi(xt) ≥ θi (not assigned to any known class). Here, for every sample that belongs to class I ˙fi is the maximum permitted distance to class I as determined by θi = μi + τσi, where σi is ξi(x) and ¯sets its trust interval. In Equation 98, the decision function is now:

$$f(\mathbf{x_t}) = \underset{1 \le i \le Nc}{\operatorname{argmin}} \xi_i(\mathbf{x_t}) \quad \text{subject to} \quad \xi_i(\mathbf{x_t}) \le \theta_i \tag{98}$$

The main gap in the SRC system is the slow speed at which an input vector (a processing speed of 0.31 fps) is sparse (Chenet al 2015). This high cost of time makes SRC unenforceable in real time scenarios. In HDC, however, it takes a very long amount of time to compare all the workout samples for the binary code generation of a test sample which could degrade the processing speed of the ODMS system. The processing speed of HDC would decrease with the increase of classes or training images.

3.4.6.3.4 SRC+HDC Scheme

In order to improve the speed of SRC processing, (Chen et al 2015) will include the HDC scheme in an SRC scheme. The SRC+HDC system has been shown to be the best accurate and slightly quicker (at 0.46 fps compared to 0.31 fps in SRC). The classification feature in Equation 99 is changed in the SRC+HDC scheme to:

$$f(\mathbf{x_t}) = \underset{i}{\operatorname{argmin}} r_i(\mathbf{x_t}) + \frac{\xi_i(\mathbf{x_t})}{N} \tag{99}$$

3.4.6.4 Proposed Classification Approaches

Classification is the third step in the BoF- ODMS approaches proposed It includes training of the classification over BoF images, which will then be used in ODMS tests. In this project, we propose two BoF-ODMS Multi-Class Vector Support Machine (SVM). Since datasets are generally unbalanced, extensive experiments are required to determine the optimum SVM parameters for the given dataset. The two approaches presented here for BoF- ODMS for multi-class classification training are: (a) the single SVM Classifier and (b) the multi-class SVM Classifier ensemble based on attribute bagging.

3.4.6.4.1 Single Multi-class SVM Classifier

Local features like SURF are extracted and incorporated into BoF histograms using a single dictionary for each training image of certain classes in the training period (or the Modular Dictionary). These BoF histograms are collected from all training pictures and used to train the described SVM classification. The BoF histogram of the given image is produced with the same dictionary in training for testing. This histogram is a basis for the addition of a vote in its predicted class by each binary classifier that comprises the Multi class SVM. The highest voting class is assigned as the predicted model image class Pseudocodes are displayed in the training and test phase pseudocodes respectively. SSVM is referred to in this classification scheme.

3.4.6.4.2 Ensemble of Multi - class SVM Classifiers based on Attribute Bagging

We describe in this section the whole multi-class SVM classification approach that we offer to BoF-ODMS We explore the idea of building a group of multi-class classifier units which are trained in various random subspaces instead of training one single classifier over the entire set (i.e., random feature subsets). This concept is mentioned in various literature names: bagging characteristics (AB), multi-channel subsets, and the random subspace approach. The Random Subspace Method is a more generic word, which could be: (1) random sub-samples can be used for the production of a training dataset bootstrap sub-set, or (2) Apply random sampling over the space of the function to create random subsets of function vectors (used in this work). We prefer to use the term Bagging attributes as it best describes the technique used

to create subsets of functions. In testing, the predictions of each classifier in the ensemble are combined to produce the final prediction with a certain combination rule.

3.4.6.4.2.1 Motivation to use Attribute Bagging

The following observation provides the motivation to adopt Attribute Bagging (AB) for the training of different classifiers of the group. The training samples per class in the dataset used in that job are too few compared to the dimensions of our feature vector, which could cause overfitting difficulties for classificatory such as SVM. In the 80-20 Dataset versions we make from the proposed heterogeneous data set, for example, 182 are the most efficient vector length features in 2000. For each class, the average number is 182. The best performing feature vector length is 2000 for the 80-20 Dataset versions we create of Proposed Heterogenous Dataset. Note that the length of the characteristic vector corresponds to the size of the dictionary used to create the function, and that each attribute of the characteristic vector corresponds to the vote of the dictionary codeword. The use of a group of classifiers constructed with AB helps prevent excessive adaptation problems. A similar random subsets of feature vectors are created in order to avoid over-fitness by reducing the difference between the number of exercise data and the vector dimensions. Contrary to them, we do not create random subsets of the training dataset, but create random subsets of vectors for training. AB-based classification ensemble could perform better than individual stand-alone classifiers (i.e., classifiers trained over whole feature vectors). We examine whether the AB-based ensemble of multi-class SVM classifiers, referred to herein as AB-SVM, can improve the efficiency of ODMS compared with the single classifier scheme, based upon their findings.

Table 3.11: Terminology

Symbol	Meaning
N_{ss}	# of Random Feature Subspaces, # of Classifiers in Ensemble
S_{ss}	Size of (#Attributes in) a Random Feature Subspace
N_i	# of training images of class-i
N_{tr}	$= \sum_{j=1}^{Nc} N_i$ (i.e., total number of training images)
S_D	Size of S_D or M_D, #Codewords in dictionary
A_g	Set of randomly chosen attribute indices for g^{th} subspace
H_{ij}	BoF feature vector of image-j of class-i
H^g_{ij}	g^{th} feature subset of H_{ij}, extracted based on A_g
H_{g_i}	Training set of class-i from g^{th} feature subspace (containing all H^g_{ij}''s)
H_g	Training set of all classes from g^{th} feature subspace (containing all H^g_i's)
C_g	a multi-class SVM classifier trained over H_g
C	the ensemble of C_g 's

3.4.6.4.2.2 Creating Random Feature Subsets by AB

In a simple example, we will show the AB method of creating function subspace. Suppose we have BoF function vectors like F = [a1,.,., . a10], where axe is the x attribute value. Let NSs = 4 create the number of random sub-spaces, each with Sss = 5 attributes (or dimensions). Let Ag denote, as described in Equation 4.58, the set of randomly selected indexes for the subspace of the gth feature, where |A|=Ss and g=1,..., Nss:

$$A_g = \{x | x \in [1, S_D]\} \quad (100)$$

Please look at A1 = {9, 6, 2, 5, 8}, A2 = {1, 7, 9, 3, 4}, A3 = {8, 1, 6, 10, 3}, and A4 = {5, 1, 2, 6, 9}, for example. From each original vector F, the following Random Function subsets are extracted from the following F: F1 = [a9, a6, a2, a5, a8], F1 = [a1, a7, a9, a3, a3, and a4], F3 = [a8, a1, a6, a10, a3], and F4 = [a5, a1, a2, a6, a9], each of the following Random Feature sub-sets are extracted respectively. Then, we would build a group of Nss= four classifiers each trained in the respective subspace of the feature.

Let us use our training dataset to generalize the application for AB. Let hij be the vector of the boF feature for an image j of class I consisting of SD attributes' = [hj, 1,.]

$$K = SD \quad (101)$$

To create each Hgij sub-set feature, select Sss < SD attributes from Hi j, randomly based on the Ag attributes indexes. The unchanged sampling ensures that each attribute is only selected once within a subset. However, in more than one subset an attribute might be chosen. All class I (i.e. Hgi j's) feature subcategories are collected in Hgi in Equation 4.60, where the Hgi dimensions of Ni Alternatively are Sss.

$$\mathbf{H_i^g} = [H_{i1}^g, H_{i2}^g, ..., H_{iN_i}^g]^T \quad (102)$$

In the respective general training set for the gth feather subspace (Hg), the class weise bundles of feature subsets (Hgs) are then collected in all classes I = 1,...Nc.

3.4.6.4.2.3 ODMS Classification for ABSVM

In the Hg function subsets, the corresponding Cg multi classifier can be trained. We thus achieve a set of classifiers C (consisting of Cg's) for predicting the fabrication and model of the vehicle in certain test images. A greater value for Nss will produce a larger number of functional subspaces that will increase the chances that Cgs differ qualitatively, as discussed by (Bryll et al 2003).

The BoF vector of the input image Ht is sampled in subsets of Htg, based on the corresponding attribute index Ag, at the testing stage. The same sequence of indices of Sss attributes (given in Ag), chosen for creating Hg, is used to create each subset. The Cg classification is then used to predict the Htg label and add a vote to the winning group. A major voting class of all Cg's (g = 1, ., . – Nss) will be produced as the predicted model for the picture test.

3.4.6.4.3 Proposed Fast Multiclass Classification - Heterogenous Techniques

Multi-class SVM-based classification procedure that uses the curse to efficiently classify high-dimensional data. The dimension curse refers to various questions concerning the analysis and organization of data in very large areas. In a highly acknowledged article on problems related to dynamic optimization (Hajikhodaverdikhan et al 2018, He et al 2012) This expression was coinciding with Richard E. Bellman. Overall, the volume of the space is increased fast and the data are reduced as dimensionality increases. For any method requiring statistical significance, the sparsity of data is problematic. The amount of required data often grows exponentially in dimensionality in order to obtain a statistically sound and reliable result, thus preventing common data mining technologies being effective. It is therefore clear that the upper limit of real risk increases when the dimension is higher in relation to the number of training instances. In consequence, the SVM is overfitted.

3.4.6.4.3.1 Divide and Conquer SVM (DCSVM)

the key idea is that any binary classifier may, in practice, separate more than two classes. Which raises a natural question: which classes are separated (and with what accuracy) by each binary classifier? DCSVM combines the one-versus-one method’s simplicity of producing balanced, fast binary classifiers with the classification speed of the DAGSVM’s decision

graph. The essential difference consists of producing the most efficient decision tree capable of delivering the decision in at most k − 1 steps in the worst-case scenario, or O (log k) steps in the best-case scenario.

3.1 DCSVM training

Let us enter certain notes and then proceed to the algorithm's formal description. Given the set data D of k classes where a label l is assigned to each item x | 0 D, . . - k {1, .,.., we want a decision function dcsvm: d {1.,.., k}, so that dcsvm(x) = l is the corresponding label x = d. Given the data set D of k classes (label). As usual, we will take into account a split D = R = T of the data set D in two split sets R (training size) and T (test set) .To measure its accuracy, use the data in R to build our dcsvm() decisions function, and then the data in T. We also take into account R = R1 = R2 da. . – The disjoint sets Rl, where each x = Rl has the label l, l = 1,. – k.(Similarly, we consider T = T1 ∪ T2 ∪ . . . ∪ Tk, has label l, l = 1, . . . , k.) Let svmi,j : D → {i, j}, be a SVM binary classifier created using the training set Ri∪Rj , i < j and i = 1, . . . , k−1, j = 2, . . . , k. .. .., k. k(k−1)/2 of the binary classifiers are k(k−1)/2. The svm() decision function needs to be clearly specified.

Our objective is to create the decisions dcsvm() with a minimum number of binary decisions for the classification of k-classes without sacrificing the accuracy of the classification. We looked to the classification of Decision Trees to achieve this objective (such as Gini impurity, Information gain, etc). But we have a different situation. In most cases, decision-making bodies of DCSVM do not divide a set of classes into disjoint subsets. Certain classes may appear at the same decision node in both subtrees. We have therefore sought other measurements to help achieve our objective.

We start by defining a few measures we use in the process of identifying the shortest path to a multi-class classification decision. Definition 1 (Class Predictions Likelihoods) The class predictions likelihoods of a SVM binary classifier $svm_{i,j}(\cdot)$ for a label l ∈ {1, . . . , k}, denoted respectively as $C_{i,j}$ (l, i) and $C_{i,j}$ (l, j), are:

$$\mathcal{C}_{i,j}(l,i) = \frac{|\{\mathbf{x} \in R_l \mid svm_{i,j}(\mathbf{x}) = i\}|}{|R_l|} \quad (104)$$

$$\mathcal{C}_{i,j}(l,j) = \frac{|\{\mathbf{x} \in R_l \mid svm_{i,j}(\mathbf{x}) = j\}|}{|R_l|} = 1 - C_{i,j}(l,i)$$

The expected probability for I or j when a binary classifier ("svmi,j") is used to predict all data items in Rl is present for every class prediction probability. For each binary and class class in the training data set, these probabilities are calculated. Each svmi,j (·) binary classifier and classes are stored in a table in the following pair of likelihood predictions.

Definition 2 Definition 2 (All-Predictions Table) We set all class predictions in rows (every binary classifier corresponding to svmi,j) and columns (every class 1, .,k) to form table T where each entry is predicted in a pair of predictions as follows:

$$\mathcal{T}[svm_{i,j}, l] = (\mathcal{C}_{i,j}(l,i), \mathcal{C}_{i,j}(l,j)) \tag{105}$$

Two next steps for the classification quality of svmi,j (·) are formally defined. The uncleanness index measures how well a binary classification can be used to classify all classes as I and j for a certain precision threshold, respectively. A class l is definitely classified by svmi,j(·), if Ci,j(l,i) Ci,j (l, i) ≥ 1 − θ; of the way; as 'define' j if Ci,j (l, j) ≥ 1 − θ; all o; otherwise, classify it as 'undesired' I or j. The impurity index counts the number of decisions that a binary classifier makes. Better the separation, the lower the index. In terms of the number of classes I or J predicted, the balance index measures how "equilibria" a separation is. When the index is larger, the better.

Definition 3 Definition 3 (SVM Impurity and Balance Indexes) In order to achieve the accuracy threshold, we define the following: - the index of impurities, denoted as:

$$\mathcal{P}_{i,j}(\theta) = \left(\sum_{l=1}^{k} (\chi_\theta(\mathcal{C}_{i,j}(l,i)) + \chi_\theta(\mathcal{C}_{i,j}(l,j))) \right) - k \tag{106}$$

the stability index, signified as $B_{i,j}$ (θ), by way of:

$$\mathcal{B}_{i,j}(\theta) = min \left(k - \sum_{l=1}^{k} \chi_\theta(\mathcal{C}_{i,j}(l,j)), k - \sum_{l=1}^{k} \chi_\theta(\mathcal{C}_{i,j}(l,i)) \right) \tag{107}$$

$$\chi_\theta(x) = \begin{cases} 1 & \text{if } x > \theta \\ 0 & \text{if } x \leq \theta \end{cases} \tag{108}$$

The next SVM score is an accuracy measurement for classifying classes I and j of the binary classification svmi,j (·). The higher the rating, the better the rating accuracy.

Definition 4 Definition (SVM Score) The SVM classification svmi,j (·) score, known as Si,j is

$$\mathcal{S}_{i,j} = \frac{\mathcal{C}_{i,j}(i,i) + \mathcal{C}_{i,j}(j,j)}{2} \tag{109}$$

Chapter 4 Obstacle Detection Model System Results and Performance Evaluation

Chapter 4 provide the details about the study as well as discuss findings of ODMS. Three machine-learning and four image feature -extraction algorithms have been used in the research covered by this thesis. In order to improve the recognition rate, we also introduced a proposal for a fast multi-class classification – the heterogeneous technology classification. ODMS uses two approaches to machine learning: Vector and Random Forest Support. It uses 4 image features, namely Invariant Scale Transform, Robust Speed-Up Feature and Histogram Oriented Gradient and GIST. Furthermore, it uses 4 image features. All three machine learning algorithms were used with the four feature extraction systems. This thesis proposes and explores uncharted approaches to BoF-based automatic barrier detection system in real time. Three main steps are proposed for the BoF-ODMS approach (1) Building Offline Dictionary (2) Features Representation using BoF (3) Training as well as testing classification by means of classifier. Performance metrics and accuracy of ODMS approaches used for the analysis of processing speed are shown. We will discuss how vehicle ROIs will be extracted in ODMS works with the parameters of the optimised dictionaries and classifiers we propose for BoF-ODMS. SIFT and SURF features are local image features grouped into a visual dictionary using a Bag of Features (BoF) approach for creating a global functionality vector. In order to find the most appropriate size several dictionaries in different sizes are used. We have designed several HOG image function vectors of various sizes based on the configuration of the HOG parameter. All ODMS system variations are compared.

An Obstacle Detection Model System (ODMS) in relations of obstacle as car monitoring as well as identification an appearance basis, a framework provides good impact rather than the license plate hooked up to the obstacles as cars. The investigation focuses on obstacles as car identification predicated upon its parameter values. ODMS system therefore identifies an obstacle as car as well as provides with obstacle as car model information. The examines carried out many times but also

here you can discovery an average return. The evaluation of the system is based on 2 Straightforward indicators (accuracy besides rapidity) besides extra added measures, with matrix besides precision of class. Discussion of the computation time aimed at meaning mining besides global function depiction through training besides recognition stage. Then discussion ODMS findings in recognition then speed and the outcomes of entirely four methods of abstraction aimed at each machine learning algorithm. Then, we compare the results of the three algorithms of machine learning and discuss the results of dividing vector conquering machine support. Finally, by considering the problem of dimensionality of the obstruction detection system, the DCSVM algorithms proposed.

4.1 Performance Metrics

4.1.1 Speed

A decent object detection system desires toward encounter actual dispensation speed necessities, as well as being accurate, to be employed in actual-life scenarios. For the ODMS approach processing speed, we use the reverse of the medium time per image in extract features (in seconds), create a global display feature, and categorize it to predict the class label. We report in frames per second the processing speed of ODMS approaches (fps).

4.1.2 Accuracy

We have k = 1, 2, ..., ND different datasets, where the original Dataset is randomly tested 80-20. The label size for all Nc classes in a dataset should be set by L = {li|i = 1, 2, ..., Nc}. By following metrics, the accuracy of ODMS approaches can be shown:

• **Average Correct Classification Rates ($ACCR_{li}$)**: The accuracy of the class is determined by the amount of quantity of appropriately classified imageries of identical to quantity of assessment imageries of the other ND data set.

$$ACCR_{l_i} = \frac{1}{N_D} \sum_{k=1}^{N_D} \frac{\#\text{Correctly Classified Images of } l_i \text{ in dataset } k}{\text{Total \# Images of } l_i \text{ in dataset } k} \tag{4.1}$$

Table 4.1: OBSTACLE DETECTION MODEL SYSTEM Classes and #Images in each 80-20 random partition of the Proposed Dataset of ODMS

Vehicle	Model	Train	Test	Vehicle	Model	Train	Test
SUZUKI	BALENO	972	194	NISSAN	MAGNITE	36	7
SUZUKI	SWIFT	556	111	NISSAN	LEAF	119	24
SUZUKI	DZIRE	546	109	NISSAN	X-TRAIL	96	19
SUZUKI	VITARA BREZZA	240	48	NISSAN	TIDA	22	4
SUZUKI	ALTO	244	49	NISSAN CEFIRO	CEFIRO	103	21
SUZUKI	WAGON	36	7	NISSAN LIVNA	LIVNA	119	24
SUZUKI	CELERIO	145	29	NISSAN MARCH	MARCH	96	19
SUZUKI	IGNIS	265	53	NISSAN SENTRA	SENTRA	22	4
FORD EXCAPE	EXCAPE	84	17	NISSAN SENTRA	SENTRA	24	5
FORD LIATA	LIATA	157	31	NISSAN TEANNA	TEANNA	66	13
FORD MONDEO	MONDEO	208	42	NISSAN TIDA	TIDA	108	22
FORD TIERRA	TIERRA	26	5	NISSAN X-TRAIL	X-TRAIL	37	7
HONDA	CRV	496	99	RENAULT	KWID	22	4

HONDA	CITY	157	31	RENAULT	TRIBER	24	5
HONDA	JAZZ	208	42	RENAULT	KIGER	22	4
HONDA	CIVIC	26	5	RENAULT	DUSTER	24	5
HONDA	AMAZE	496	99	SUZUKI	SOLIO	34	7
HONDA	FIT	9	2	TATA	TIAGO	9	2
HONDA	WR-V	38	8	TATA	ALTROZ	9	2
HONDA CRV	CRV	113	23	TATA	TIGOR	9	2
HONDA FIT	FIT	9	2	TATA	NEXON	9	2
HYUNDAI	SANTRO	26	5	TATA	HARRIER	9	2
HYUNDAI	GRAND I10	16	3	TATA	SAFARI	9	2
HYUNDAI	AURA	12	2	TORD TARCEL	TARCEL	110	22
HYUNDAI	VERNA	26	5	TOYOTA	WISH	20	4
HYUNDAI	CRETA	16	3	TOYOTA	INNOVA	24	5
MAHINDRA	E20PLUS	12	2	TOYOTA	ALTIS	29	6
MAHINDRA	BOLERO	26	5	TOYOTA	CAMRY	34	7
MAHINDRA	XUV300	16	3	TOYOTA	SURF	56	11

MAHINDRA	THAR	12	2	TOYOTA	GLANZA	55	11
MAHINDRA	SCORPIO	26	5	TOYOTA	YAARIS	34	7
MITS.	OUTLANDER	12	2	TOYOTA	MARCH	22	4
MITS. LANCER	LANCER	16	3	TOYOTA	ALTIS	521	104
MITS. ZINGER	ZINGER	12	2	TOYOTA CAMRY	CAMRY	247	49
NISSAN	LIVNA	22	4	TOYOTA INNOVA	INNOVA	15	3
NISSAN	TEANNA	24	5	TOYOTA RAV4	RAV4	79	16
NISSAN	SENTRA	66	13	TOYOTA SURF	SURF	40	8
NISSAN	CEFIRO	22	4	TOYOTA VIOS	VIOS	242	48
NISSAN	SAFARI	24	5	TOYOTA WISH	WISH	175	35
NISSAN	SUNNY	66	13	TOYOTA YARIS	YARIS	155	31
NISSAN	TERRA	67	13		**TOTAL**	**8436**	**1687**

• **Mean Average Correct Classification Rate (mACCR)**: The overall accuracy of ODMS, the metric similar to that of which is averaged over splits of ND data set in Aggregate images are taken categorized mostly by cumulative number of successful pictures with in data set appropriately among all categories.

$$mACCR = \frac{1}{Nc} \cdot \frac{1}{N_D} \sum_{i=1}^{Nc} \left(\sum_{k=1}^{N_D} \#\text{Correctly Classified Images of } l_i \text{ in dataset } k \right) \quad (4.2)$$

4.1.3 Decision Capability

The confusion matrix is a good instrument for visualizing the discriminatory capabilities of ODMS approaches. While Indexes of a matrix line match tags with real data, but also section indices fit the class values Forecasted. this r^{th} rows besides c^{th} value in which r = 1, ..., Nc and c =1, Nc represents the percentage of images of class r that the ODMS approach predicts are class c. The ACCRlr is the principal diagonal value. This is the ACCRlr of class r for each of them (r, r). The confusion matrix helps us to identify classes that might seem similar (in the space) and lead to imprecise forecasts.

4.2 Best Parameters Selection

Many possible ways to identify important input dimensions of a model are available. For example, in a ridge regression, model coefficients w can be measurement (large coefficients mean that small changes in a feature value have a major impact on predictions (small length scales mean that small changes in a feature value have a large effect on predictions]). For random woods, Breiman proposed that the values in the ith column of the out-of-bag data (or validation) be perturbed and that the resultant loss of predictive precision be measured.

Algorithm 4.1: ODMS Best Parameters Selection

In line 10, learn refers to an arbitrary regression method that fits a function f to given training data. Note that input dimensions 1, . . . , k are parameters, k + 1, . . ., k + m are features.

Input: Training data $D_{train} = \{h(x_1, y_1), \ldots, (x_n, y_n)\}$; validation data $D_{valid} = \{h(x_{n+1}, y_{n+1}), \ldots, (x_{n+n'}, y_{n+n'})\}$; number of parameters, k; number of features, m; desired number $K \leq d = k + m$ of key inputs; bound on number of key parameters, $k_{max} \geq 0$; bound on number of key features, $m_{max} \geq 0$, such that $k_{max} + m_{max} \geq K$

Output: Subset of K feature indices $S \subseteq \{1, \ldots, d\}$

1. $I \leftarrow \{1, \ldots, d\}$;
2. $S \leftarrow \emptyset$;
3. for $j = 1, \ldots, K$ do
4. $I_{allowed} \leftarrow I \setminus S$;
5. if $|S \cap \{1, \ldots, k\}| \geq k_{max}$ then $I_{allowed} \leftarrow I_{allowed} \setminus \{1, \ldots, k\}$;
6. if $|S \cap \{k + 1, \ldots, k + m\}| \geq m_{max}$ then $I_{allowed} \leftarrow I_{allowed} \setminus \{k + 1, \ldots, k + m\}$;
7. for all the $i \in I_{allowed}$ do
8. $S \leftarrow S \cup \{i\}$;
9. forall the $(x_j , y_j) \in D_{train}$ do $x^S{}_j \leftarrow x_j$ restricted to input dimensions in S;
10. $f \leftarrow$ learn $(h(x^S{}_1 , y_1), \ldots, (x^S{}_n, y_n))$;
11. $err(i) \leftarrow \sqrt{\sum_{(\mathbf{x}_j, y_j) \in \mathcal{D}_{valid}} (f(\mathbf{x}_j) - y_j)^2}$;
12. $S \leftarrow S \setminus \{i\}$;
13. $\hat{i} \leftarrow$ random element of arg $\max_i$ err(i);
14. $S \leftarrow S \cup \{\hat{i}\}$;
15. return S;

We achieve optimum parameters with cross-validation using a different 80-20 data set for every step of our BoF-ODMS approaches. For both SD and MD schemes, we first find optimal dictionary parameters. Then, for single classifier and set classification schemes, we find the optimum classification parameters. For the requirements of an ODMS real-time system, the best trade-off between processing speed and accuracy is considered in each case.

4.2.1 Improved Dictionaries

For the optimal SD and MD we use SURF in the local features extraction step and single multi-class SVM in classification step to run extensive experiment on the BoF-ODMS approach. Dictionary Size, SD is the important parameter which affects processing speed and accuracy in the entire BoF-ODMS system. In the offline Dictionary Building, we found that SD = 2000 produced the highest compensation between speed and accuracy, with the SD schema changing from 100 to 4000. (As shown in Figure 4.2). We get a 95.54% exactness at a speed of 7.4 fps with SD = 2000. For SD > 2000, the accuracy is only slightly enhanced, as the speed falls quickly. S D=4000 leads to greater precision (96%) but the speed is reduced to 6.7 fps, for example. Therefore, unless otherwise indicated, we select SD = 2000 from our experiments based on SD (BoF-SD) BoF-ODMS. As for the MD scheme, we have done the same experiments by different sizes of individual (S ID) dictionaries that make up the principal MD and we have found the best interchange between the speed and the accuracy of the ID = 100 (which makes the overall MD size S D = S ID · Nc=100 · 29) The increase of the S ID over 100 gradually reduces both speed and precision. Therefore, we use S ID = 100 in our BoF-ODMS based experiments with MD (BoF-MD). In BoF- ODMS approaches, the algorithm 4.2 is used to incorporate the local features of certain pictures into global boF representatives using the D or DM (constructed in the off-line dictionary construction steps). For a distinctive IJ class I picture, its BoF picture is a basic HiJ bar chart with casting votes in dictionary plaintext. Hello, a hij histogram may be shown by: Hi j = [h1, h2, ..., hSD] where the bins (hk's) hold the respective number of votes (cwk's) (k = 1, ..., S D), each with the following codings. Each feature matches the closest codeword to the dictionary by adding a unit vote. In order to create the BoF image representation (histogram), each local feature of Ij Fp / FIj is matched to its closest cwk codeword (D or DM) in the dictionary, and the respective histogram bin hk is increased to the voting count.

Algorithm 4.2: Algorithm for BuildDictionary

1: Input: The sets of images from all classes, MI = {I1,I2, ...,INc}, Dictionary sizes for SD (S D), MD (S MD) and its sub-dictionaries (S ID)

2: Output: D or DM

3: Initialize: MF

4: Step 1: Collecting Local Features

5: for each class i ∈ [1, Nc]: do

6: Initialize Fi = {}

7: for each image j ∈ [1, Ni]: do

8: Fi j ← FeatureExtraction(Ii j) .

9: Fi = Fi ∪ {Fi j} . Fi j is set of local features in j

10: end for

11: MF = MF ∪ Fi

12: end for

13:

14: Step 2: Dictionary Building

15: Initialise:

16: Single-Dictionary D,

17: Modular-Dictionary DM

18: Sub-dictionaries (Di's) of DM

19: Build SD:

20: D ← Cluster (MF, S D)

21:

22: Build MD:

23: for each class i: do

24: Di ← Cluster (Fi, S ID)

25: DM ← DM ∪ Di

26: end for

Algorithm 3. A phase also recognized by way of quantizing features. The histogram undertakes a normalization technique once entirely FIj characteristics are quantized. Thus, after matching all features of a specified image, the outcome is concluding histogram besides demand the situation a BoF histogram or function.

Algorithm 4.3: Algorithm for ODMS Features Quantization Method

1: Input: The set FI = {fp|p = 1, ..., N fI} of local feature descriptors of an image I, the respective dictionary (D or DM)
2: Output: The computed BoF Histogram, HI
3: Initialize: HI ← [0, ..., 0] where |HI| = S SD or S MD
4: for each fp ∈ FI do
5: k = argmint∈[1,S D] dist(cwt, fp)
6: HI(k) ← HI(k) + 1
7: end for
8: return HI

The functionality and centre adaptation process is repeated until the centers no longer change place. A visual word represents the centre of each learned cluster. The size of the dictionary for visual words is similar to the number of clusters learned. Thus, each image function of the SIFT and SURF is matched with the clustering algorithm The histogram of happenings with visual words is also a pic to both a particular visual word. Eventually, a histogram have been normalized as well as the histogram is the overall picture. A pictorial dictionary has been created with only images of training datasets, while the same dictionary is encoded for both training and testing datasets. Algorithm 4 outlines the procedure for creating a BoF-based global image histogram. The BoF histogram for local characteristics is calculated for a particular picture Ij of the class i.

4.2.2 Optimized Classifiers

Algorithm 4.4: Algorithm for ODMS- GetGlobalRepresentation

1.Input: An input image I, The dictionaries (SD and DM), Mode (= S D or MD), Dictionary sizes for SD (S SD), MD (S MD)
2: Output: The computed BoF Histogram, HI
3: Initialize:
4: HI ← [0, ..., 0] where |HI| = S SD or S MD
5: Step 1: Features Extraction
6: FI ← FeatureExtraction(I)
7: Step 2: Features Quantization . See Algorithm 4.3
8: if Mode = S D then
9: HI ← FeatureQuatization (FI, D)
10: elseif Mode =MD
11: HI ← FeatureQuatization(FI, DM)
12: end if
13: Step 2: Histogram Normalization
14: for each b ∈ [1, ..., |HI|] do
15: HI(b) ← HI(b)/N fI . where N fI = |FI|
16: end for
17: return HI

We will find the optimal classifiers for BoF-ODMS based on the optimised dictionaries obtained. We use the OpenCV multimedia library based on LibSVM to construct the two classification schemes proposed for the BoF-ODMS (H. He, Z. Shao, and J. Tan et al,2015, C.-C. Chang and C.-J. Lin et al 2011).Local features such as SURF are extracted and integrated with BoF histograms using the single dictionary (or modular dictionary) for each train image of the detailed classes throughout the training phase. These BoF histograms are

collected as of entirely training images besides stay cast-off toward train the SVM classifier of the numerous classes. Thus, the BoF histogram of the examination image remains formed with similar phrasebook cast-off the workout. This histogram is used to add a vote to individually dual classifier that includes SVM multi-class. maximum secret ballot class remains allocated by way of the foreseen examination picture perfect class. Process typically includes aimed at training and tests, respectively, can be found in algorithms 4.5 and 4.6. SSVM shall refer to this classification scheme.

Algorithm 4.5 Algorithm for Training Phase of BoF-ODMS with SSVM

1: Input: The collection of sets of training images from all classes MI = {I1,I2, ...,INc}, The Single- or Modular-Dictionary (D or MD), Mode ∈ ['SD','MD'] indicating which dictionary to use

2: Output: The trained Single-SVM classifier, C

3: Initialize: The pool of all training BoF histograms H

4: Step 1: Generating BoF Global Representations of Images. Using Algorithm 4.4

5: for each class i ∈ [1, . . . , Nc] do

6: Initialize Hi = {}. Hi is set of class i's training BoF histograms

7: for each image j ∈ [1, . . ., Ni] do

8: Fi j ← Features Extraction (Ii j)

9: if Mode == 'SD' then

10: Hi j ← GetGlobalRepresentation (Fi j, 'SD')

11: elseif Mode == 'MD'

12: Hi j ← GetGlobalRepresentation (Fi j, 'MD')

13: end if

14: Hi ← Hi ∪ (Hi j, yi) . yi is the label of class i

15: end for

16: H ← H ∪ Hi

17: end for

18: 19: Step 2: Training SVM Classifier

20: C ← SVM.Train(H)

Algorithm 4.6 Algorithm for Testing Phase of BoF- ODMS with SSVM

1: Input: A test image It, its ground truth class-label yot , Mode ∈ ['SD','MD'] indicating which dictionary to use
2: Output: ypt (the predicted class label of It)
3: Initialize: Ht , the BoF histogram of It
4: Step 1: Generating BoF Global Representation . Using Algorithm 4.2
5: if Mode == 'SD' then
6: Ht ← GetGlobalRepresentation(Ft, 'SD')
7: elseif Mode == 'MD'
8: Ht ← GetGlobalRepresentation(Ft, 'MD')
9: end if
10: Step 2: Prediction
11: ypt ← C.Predict(Ht)

We propose for BoF-ODMS the set of multi-class SVM classifiers. Exploration the knowledge of generating an collaborative of separate multi classifiers accomplished popular numerous arbitrary subspaces as an alternative of exercise the only classifier happening the entire range of feature vectours (i.e., random feature subsets). The term "bagging," the multiple sub-sets and the random sub-surface method" is referred to as various names in the literature. A more general term could be: (1) using Random Subsamples over training data samples to generate bootstrap subsets for the training dataset or, Random Subspace Method (2) Apply the function space of Random Sub-Sampling to create the arbitrary subsections of directions. chosen toward usage the bagging since the situation finest defines the method cast-off towards generate function subdivisions. As soon as testing, the forecasts of a piece of classifiers of collaborative remain joint towards yield the concluding forecast with a certain combination rule.

The functional subsets in Hg are used to train multi-class C g. This enables us to produce a set of classificatory C (C-g 's) to predict the automotive model in certain photographs of the test. For Nss, a higher value will yield a greater number of subspaces, increasing the

chances, of having qualitatively different C g's. The BoF feature vector of the image is subsampled in the testing phase to the sub-sets H gt based on the respective indexes of attributes Ag. The sequence of Ss attribute (given by Ag) indices selected in the creation of Hg is used to create every subset. Classification the Cg stands at that time cast-off towards forecast Hgt, through calculation a vote towards the victorious class. The class which receives the majority of the votes among C g's (g = 1, ., Nss) is produced in the form and model predicted for the test image. The pseudo code for training & testing phases of BoF-ODMS with ABSVM is provided by algorithms 4.7 and 4.8.

Algorithm 4.7 Algorithm for Training Phase of BoF-ODMS with ABSVM

1: Input: The collection of sets of training images from all classes MI = {I1,I2, ...,INc}, The Single- or Modular-Dictionary (D or MD), Mode ∈ ['SD','MD'] indicating which dictionary to use, The collection A = {Ag|g = 1, . . . , Nss}, where Ag is the set of selected attribute indices for gth feature subspace.

2: Output: The trained ABSVM classifier, C = {Cg|g = 1, ..., Nss}

3: Initialize:

4: Hgi , ∀g ∈ [1, . . . , Nss], ∀i ∈ [1, . . . , Nc]

5: Hg, ∀g ∈ [1, . . . , Nss]

6: Step 1: Generating BoF Global Representations of Images . Using Algorithm 2

7: for each class i ∈ [1, . . . , Nc] do

8: for each image j ∈ (1, . . . , Ni] do

9: Initialise: Hi j, and H1 i j, . . . , H i jNss

10: Fi j ← FeaturesExtraction(Ii j)

11: if Mode == 'SD' then

12: Hi j ← GetGlobalRepresentation(Fi j, 'SD')

13: elseif Mode == 'MD'

14: Hi j ← GetGlobalRepresentation(Fi j, 'MD')

15: end if

16: for g = 1, . . . , Nss do

17: Hgi j ← SubSample(Hi j, Ag) . select Sss < SD attributes from Hi j, based on Ag

18: Hgi ← Hgi ∪ (Hgi j, yi) . yi is the label of class i

19: end for

20: end for

21: end for

22:

23: Step 2: Collecting Training Data

24: for g = 1, . . . , Nss do

25:for i = 1, . . . , Nc do

26: Hg ← Hg ∪ Hgi

27: end for

28: end for

29:

30: Step 3: Training the Ensemble of Classifiers

31: for g = 1, . . . , Nss do

32: Cg ← SVM.Train(Hg)

33: end for

Algorithm 4.8 Algorithm for Testing Phase of BoF- ODMS with ABSVM

1: Input: A test image It , its ground truth class-label yot , Mode ∈ ['SD','MD'] indicating which dictionary to use, Mode ∈ ['SD','MD'] indicating which dictionary to use, The collection A = {Ag|g = 1, . . . , Nss}, where Ag is the set of selected attribute indices for gth feature subspace.

2: Output: ypt (the predicted class label of It)

3: Step 1: Generating BoF Global Representations of Images . Using Algorithm 2

4: Initialise: Ht , and H1t , . . . , HtNss

5: Ft ← FeaturesExtraction(It)

6: if Mode == 'SD' then

7: Ht ← GetGlobalRepresentation(Ft, 'SD')

8: elseif Mode == 'MD'

9: Ht ← GetGlobalRepresentation(Ft, 'MD')

10: end if

11: for g = 1, . . . , Nss do

12: Hgt ← SubSample(Ht, Ag)

13: end for

14:

15: Step 3: Prediction

16: Pgt = {} . The list of prediction outputs of individual classifiers

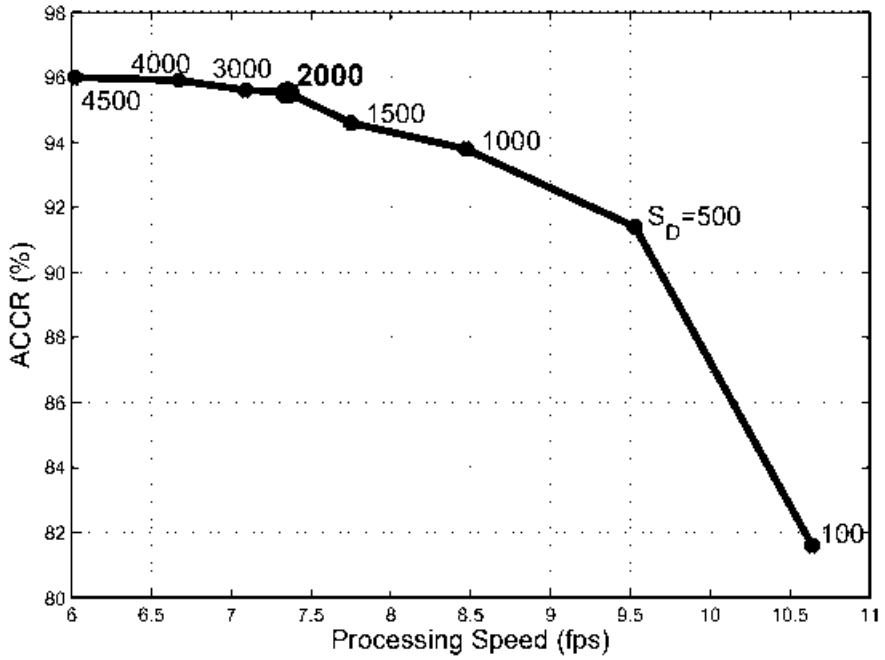
17: for g = 1, . . . , Nss do

18: Pgt ← Pgt ∪ Cg .Predict(Hgt)

19: end for

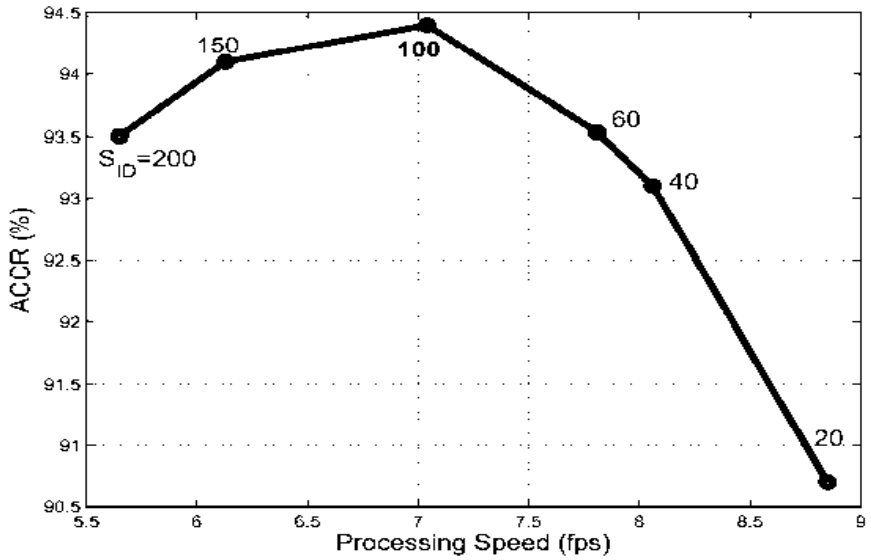
20: ypt = MajorityVoting(Pgt) . Select the class with maximum votes

Figure 4.1: The effect on Accuracy and Speed of ODMS based on BoF-SD of varying S D(100 to 4000).



The best compromise between speed and precision is achieved by SD = 2000.

Figure 4.2: Effects for the precise processing and speed of BoF-MD-based ODMS, varying size of individual dictionaries (SIDs) from 20 to 200.



We achieve the best balance between speed and precision at SID = 100. Given that the BoF feature vector size is rather high, where the number of feature sub-spaces Nss is not enough and therefore the classifier number, certain attributes may not be selected. If Nss = 2 is used and only the A1 and A2 features are chosen, then we can see that the attribute 10 is not available. Likewise, the fourth and seventh attributes if A1 and A3 are chosen. In the resulting function subsets, it would be left out. The missing attributes may have high capacity or involvement to discriminate. Based on experimental evaluations, as discussed below, we decide upon optimum values for Nss and Sss. In this work, we select the optimized values for sub-space functional sizes and the number of sub-space functions by studying their effect on the speed and precision of BoF-SD processing with AB-SVM (See Figures 4.4a and 4.4b). For every test with NSs = 10, 15, 20, and 25, we varied from Ss 100 to 1000. Figures 4.4a and 4.4b indicate that while with increased Sss and Nss accuracy is tending to increase, speed is decreasing. This is obvious because a higher Ss shows a greater dimensionality and a higher Sss shows a higher number of classifiers within the ensemble, both leading to more time consumption in the processing. We see accuracy vs. speed plot as shown in Figure 4.3 to find the optimal values of Nss and Ss for BoF-SD. In this chart, we see that speeds tend to fall drastically for a curacies greater than 94%, while accuracy only slightly improves towards 95%. Our aim is to achieve a processing speed of at least 5 fps to satisfy real-time needs with a minimum accuracy of around 95%. From the above figures,

the accuracy of Ss = 500 and N ss = 15 is about 95% and the speed is about 5 fps (represented by the highlighted green square datapoint in Figures 4.3a, 4.3b, and 4.3c). For the BoF-SD and BoF-MD approaches, based on the AB-SVM, sss = 1500. Similar testing has been carried out for BoF-MD and Nss based on AB-SVM = 15, Sss = 1500 yielded the best speed-accuracy compensation. In our experiments, therefore, we take the AB-SVM-based BoF-SD and BoF-MD approaches respectively (Nss = 15, Sss = 1500), respectively.

Figure 4.3: Effect on (a) average classification of the characteristic's subset size (S ss) and sub sample quantities (Nss)

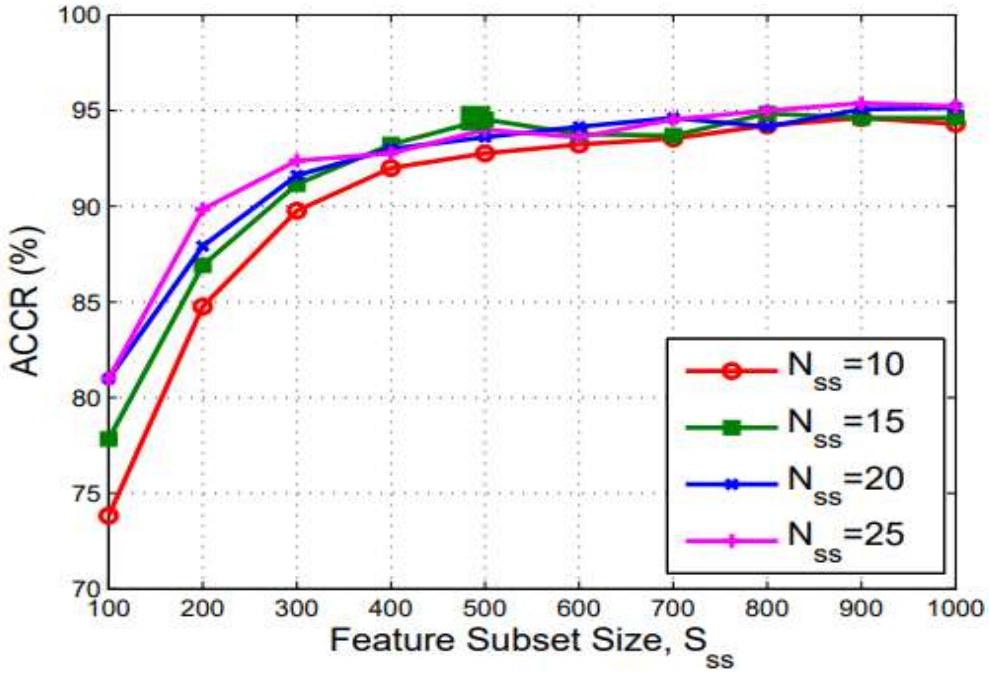


Figure 4.3: Effects on (b) Processing speed of the feature subset sizes (S ss) and number of subsamples (Nss)

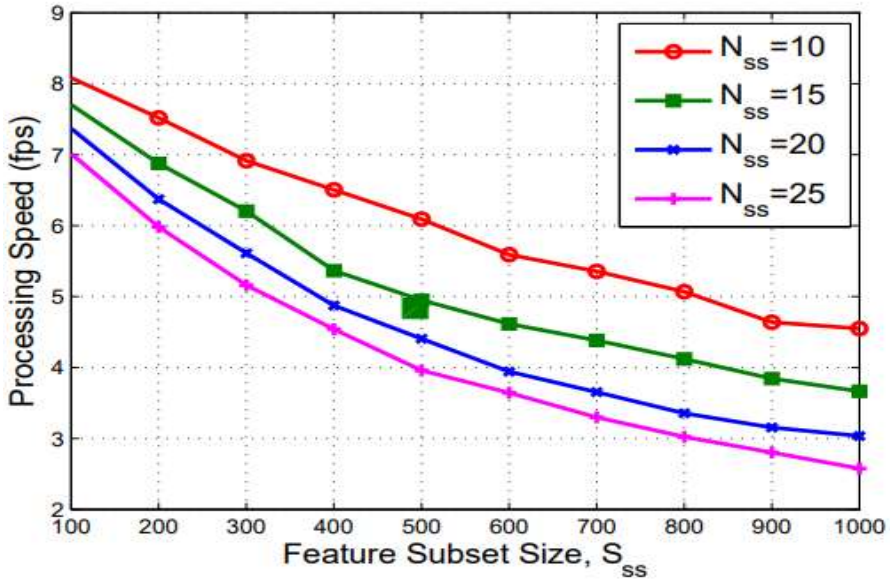
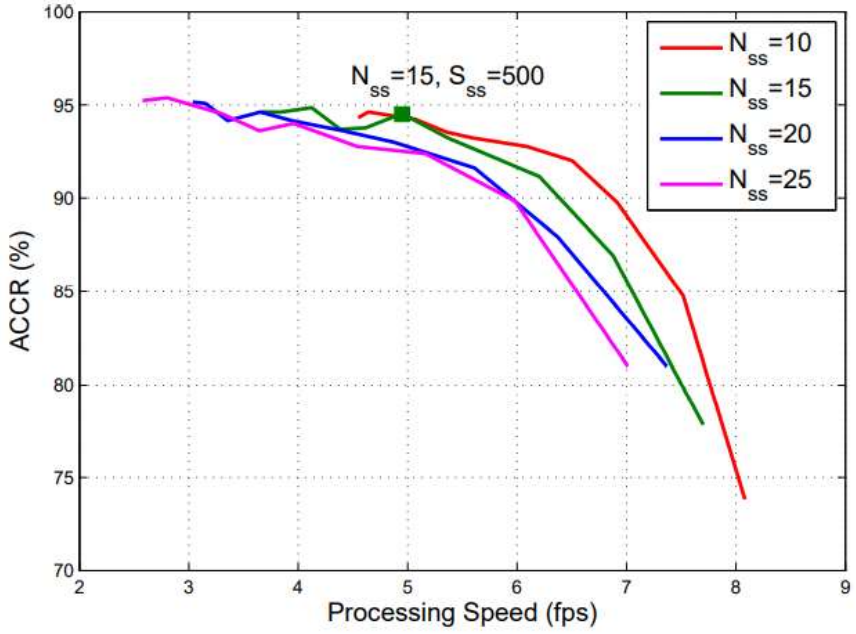


Figure 4.4: ACCR vs. AB-SVM Processing BoF-SD speed for the various sub-sample numbers (NSs) and subset feature sizes (S ss). Nss = 15 and Ss = 500 is the fastest compromise.



4.3 Obstacle as Vehicle Region of Interest

In most works, the narrow region of interest (ROI) on the vehicle is considered so that it covers the front (or rear) face instead of using the whole input image for extraction of features, global representation and classification. This eliminates input image background regions, which could otherwise reduce ODMS performance. The work that focuses on the extraction of this car ROI from input images are classified as systems for vehicle detection (VD), which have over the years been a very extensively examined subject. In S you can find an extensive survey of VD techniques and approaches (Sivaraman and M. Trivedi et al 2013) Most works are based on a pre-constructed VD module. From our point of view Vehicle detection is out of the scope of this thesis, methods for representating and classifying vehicles. Nevertheless, we provide a brief overview of VD techniques that have been used directly in ODMS works in order to be complete in this context. In ODM Structure (M. Fraz, and E. A. Edirisinghe, and M. S. Sarfraz, and al. 2014) licence plate (M. Sarfraz etal.) was popular in vehicles detected and located on a particular image, such as the bumper shadow (Y.-W. Hsieh, L.-C Chen and D.-Y. Chen and al 2014, M. Cheon, W. Lee, C. Yoon and M. Park et al 2011), air grille (A. Psyllos, C. Anagnostopoulos, and E. Kayafas et al 2011), etc. Speed and lower than other advanced VD techniques computer complexity. Since the goal of ODMS work is to increase traditional vehicle identification and classification systems, the detected licence plate is more efficient as a basis for the definition of vehicle ROIs. Although LPR structures remain extremely defective, license platter discovery methods take to remain tremendously healthy underneath numerous circumstances besides consume the compensations of enlarged giving out rapidity, fewer computer complication besides negligible disappointment (A. Al-Ghaili, S. Mashohor, A. Ramli, and A. Ismail et al 2015).

4.4 Performance of SSVM-based BoF-ODMS

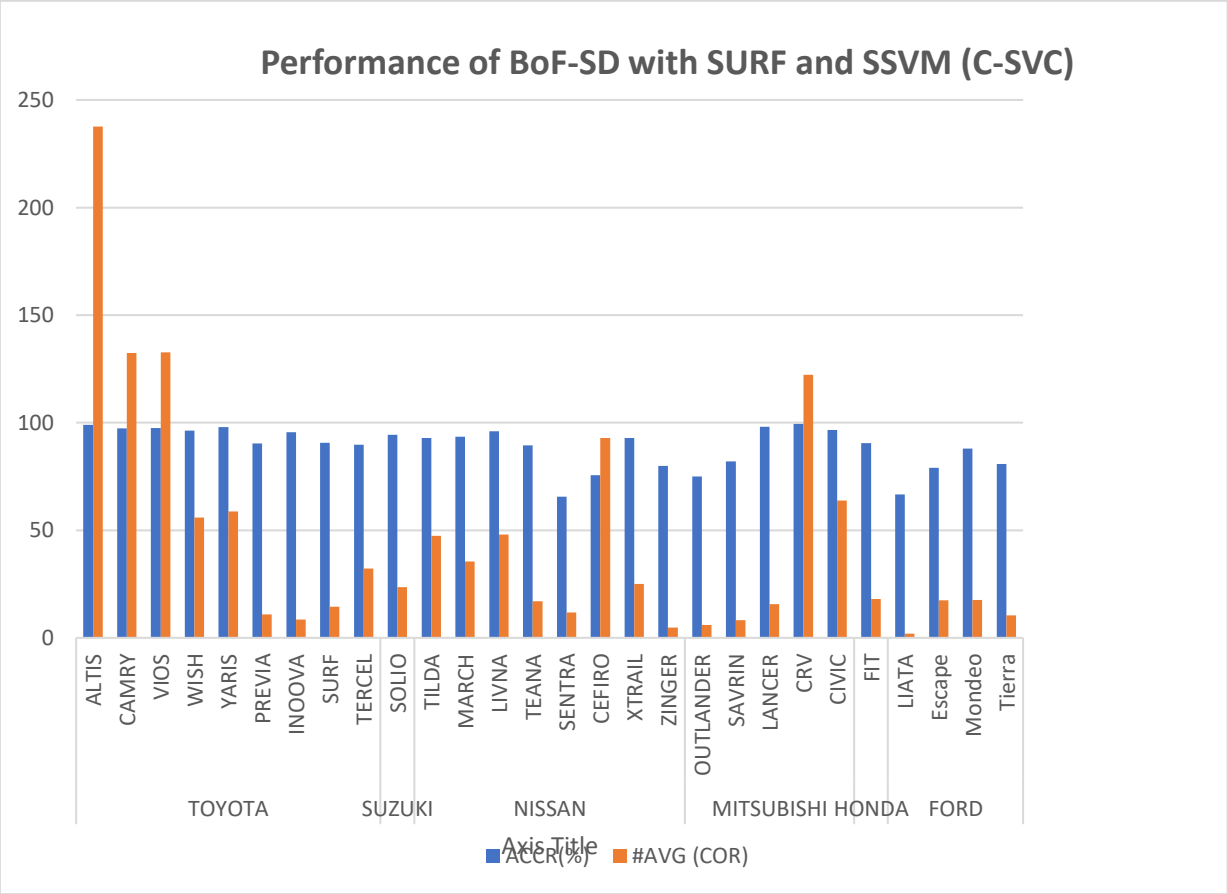
Assess BoF-ODMS performance for global representation under both the SD and MD schemes.

Table 4.1 SURF and SSVM performance of BoF-SD (C-SVC)

Sr. No	MAKE	MODEL	ACCR (%)	#AVG (COR)
1.	TOYOTA	ALTIS	99	237.6
2.		CAMRY	97.35	132.4
3.		VIOS	97.57	132.7
4.		WISH	96.38	55.9
5.		YARIS	98	58.8
6.		PREVIA	90.38	10.9
7.		INOOVA	95.56	8.6
8.		SURF	90.63	14.5
9.		TERCEL	89.72	32.3
10.		RAV4	99.35	30.8
11.	SUZUKI	SOLIO	94.4	23.6
12.	NISSAN	TILDA	92.94	47.4
13.		MARCH	93.42	35.5
14.		LIVNA	96	48
15.		TEANA	89.47	17
16.		SENTRA	65.66	11.8
17.		CEFIRO	75.67	92.96
18.		XTRAIL	92.86	25.1
19.		ZINGER	80	4.8
20.	MITSUBISHI	OUTLANDER	75	6
21.		SAVRIN	82	8.2
22.		LANCER	98.13	15.7
23.		CRV	99.43	122.3
24.		CIVIC	96.67	63.8
25.	HONDA	FIT	90.5	18.1
26.	FORD	LIATA	66.67	2
27.		Escape	79.09	17.4

28.		**Mondeo**	**88**	**17.6**
29.		**Tierra**	**80.77**	**10.5**
		OVERALL	**94.84**	**1232**

Figure 4.5 SURF and SSVM performance of BoF-SD (C-SVC)



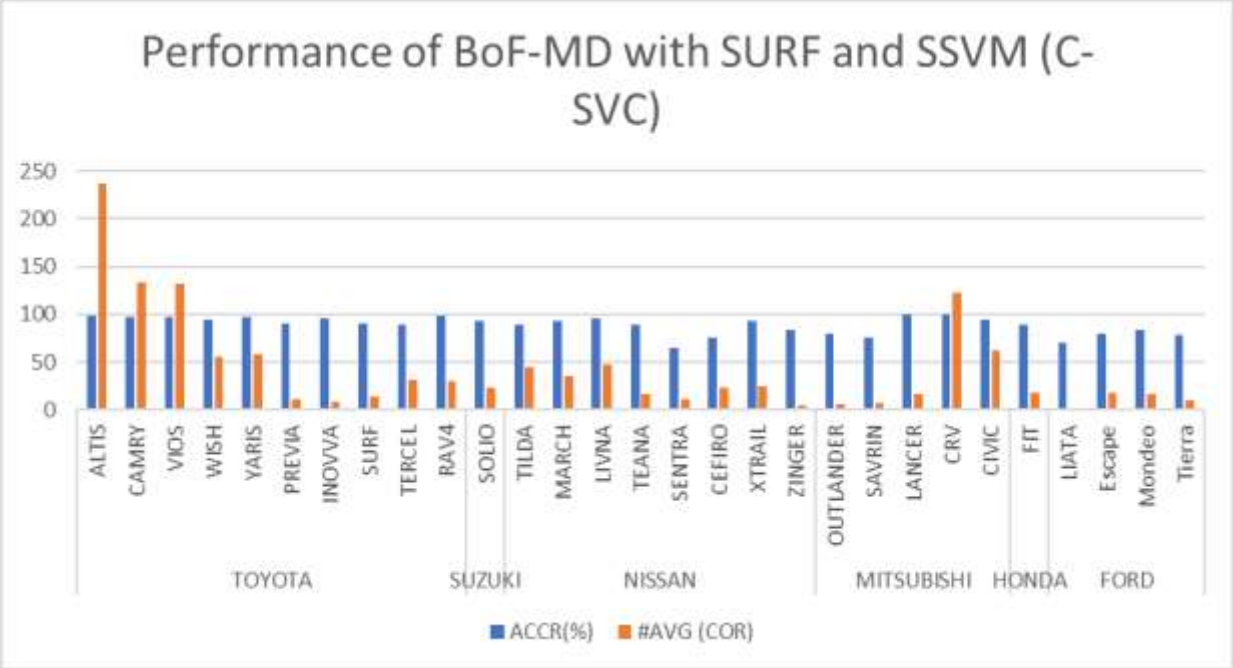
4.4.1 Accuracy

Table 4.2: SURF and SSVM performance of BoF-MD (C-SVC)

Sr. No	MAKE	MODEL	ACCR (%)	#AVG (COR)
1.	TOYOTA	ALTIS	98.36	236.8
2.		CAMRY	97.65	132.8
3.		VIOS	97.21	132.2
4.		WISH	95.00	55.1
5.		YARIS	97.50	58.5
6.		PREVIA	90.83	10.9

7.		INOVVA	95.56	8.6
8.		SURF	90.63	14.5
9.		TERCEL	88.61	31.9
10.		RAV4	98.06	30.4
11.	SUZUKI	SOLIO	93.20	23.3
12.	NISSAN	TILDA	88.82	45.3
13.		MARCH	93.42	35.5
14.		LIVNA	96.00	48
15.		TEANA	89.47	17
16.		SENTRA	65.56	11.8
17.		CEFIRO	75.67	22.7
18.		XTRAIL	92.96	25.1
19.		ZINGER	83.33	5.0
20.	MITSUBISHI	OUTLANDER	80.00	6.4
21.		SAVRIN	75.00	7.5
22.		LANCER	99.38	15.9
23.		CRV	99.27	122.1
24.		CIVIC	94.24	62.2
25.	HONDA	FIT	89.00	17.8
26.	FORD	LIATA	70.00	2.1
27.		Escape	80.00	17.6
28.		Mondeo	84.00	16.8
29.		Tierra	77.69	10.1
		OVERALL	93.70	1217.2

Figure 4.6 Performance of SURF and SSVM performance of BoF-MD (C-SVC)

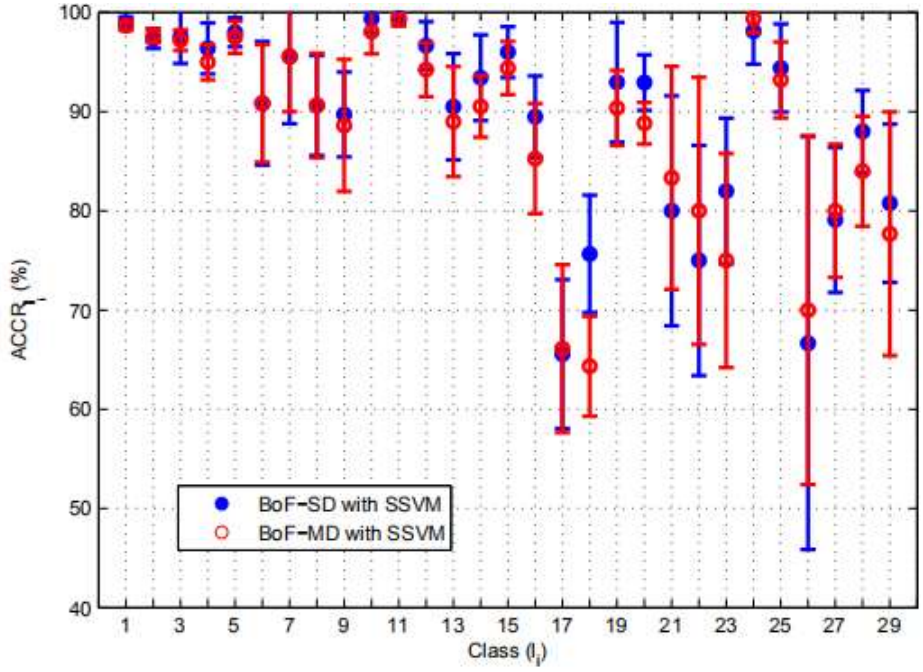


4.4.1.1 C-SVC based SSVM

We evaluate BoF approaches on the basis of SD and MD with the C-SVC formulation for SSVM. The ACCRli's and the average amount of images for BoF-SD and BoF-MD correctly classified. The accuracies of training-testing are measured over ND=10 random 80-20 data sets. SD = 2000 (for SD) and S D = S ID Nc = 100 · 29 The size of the dictionary used is (for MD). Figure 4 shows the class ACCRli's of the SURF- and SSVM-based BoF-SD and BoF-MD-related approach with 95% trust intervals over 10 various versions of the 80-20 training testing datasets. 1 Each of the 10 versions of the data set consists of the random selection of 80% of total images for training and the remaining ones for test. Although most classes are high. Our BoF-SD and BoF-MD approaches have mean correct classification rates of 94.84 percent, respectively 93.7 percent. Some classifies, like classes 18 (Nissan Cefifiro), 17 (Nissan Sentra), 18 (Nissan Cefifiro), 22 (Mitsubishi Outlander),23 (Mitsubishi Savrin) and 27, have consistently

performed badly across the ten different dataset versions (Ford Escape). It is worth noting that BoF-SD had ACCR 95-100 percent in some classes that are low in precision, e.g.: classes 21 (Mitsubishi Zinger), 26 (Ford Liata), 28 (Ford Mondeo), and 29 (Fort Tierra). Likewise, while the accuracy (averaged across all of the data sets) was 79.09 percent for class 27 (Ford Escape), at least one of 80-20 data sets was split, more than 90 percent was achieved for ACCRli. This clearly shows that accuracy depends greatly on the distribution of images in training and testing. Whereas we randomly assign images to test sets, the original data set assigns images to training and test sets.

Figure 4.7: Classwise ACCRli 's with 95% Confidence Intervals for BoF-SD and BoF-MD with SSVM (over the ND = 10 different 80-20 Dataset splits).



4.4.1.2 v-SVC based SSVM

We examined another SSVM classification formulation: the v-SVC. We evaluate v-SVC with BoF-SD since BoF-SD outperformed BoF-MD. BoF-SD results with SSVM based respectively on C-SVC and v-SVC. Interestingly, c-SVC was similar to C-SVC in terms of overall mACCR accuracy with these optimum values. The processing speed of BoF-SD with C-SVC, however, was around 7.5 fps. It averages 7.4 fps.

4.4.2 Speed

The average SD and MD-based BoF-ODMS processing speeds for SSVM are 7.5 fps and 6.99 fps, respectively, proving the BoF's suitability for ODMS applications in real time. Higher speeds are achieved by decreasing the S D with a slight precise compromise. The BoF parameters could be easily adapted to meet high speeds of processing, depending on the requirements of a specific application, with some precise compromise or vice versa. Common monitoring cameras have a 25-30 fps frame rate. But it would waste computer resources to run the ODMS on every incoming framework. Instead, every 5th incoming frame for O DMS purposes can be seamlessly processed and only 5-6 frames per second are actually required for processing. Both of our BoF approaches are therefore highly suitable for ODMS applications in real time.

4.5 Comparison of ODMS BoF-SD and ODMS -BoF-MD

4.5.1 Accuracy and Speed

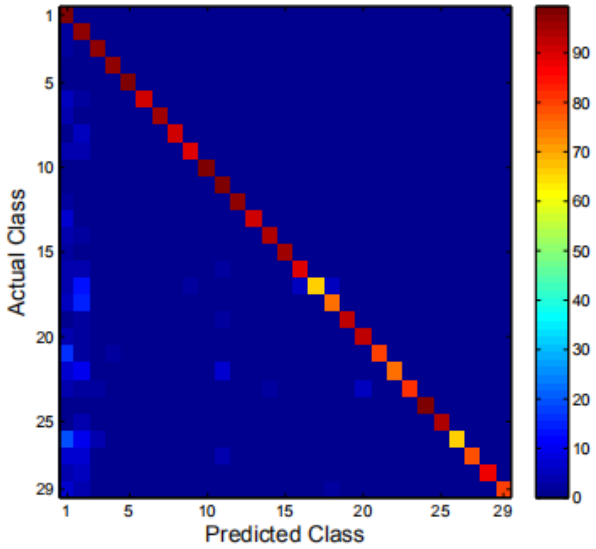
Observing BoF-SD and BoF-MD accuracy respectively, you can see that BoF-SD has a higher overall performance than BoF-MD. The results indicate otherwise, contrary to our expectations that BoF-MD would perform better than BoF-SD.

One reason could be a certain subsectionary size of all classes, which could lead to the selection of many features which are fewer discriminatory besides raucous by way of codewords popular the complete wordlist. In upcoming, a interpretation method aimed at a extra vigorous Modular Dictionary will be investigated. We note, however, that BoF-MD had better ACCR than BoF-SD for some classes. See: Toyota Camry, Nissan Sentra, for instance. In figures 4.2a and 4.2b, we show the confusion matrix (with SURF and SSVM) of the 29 model classes using BoF-SD and SSVM approaches. It can be observed that Classes 1 (Toyota Altis) and 2 are mostly the improper classification (Toyota Camry). A considerable increase in the number of training data available for those classes could be one of the main reasons for this effect, which could lead to a biased classification. when the dictionary could be distorted with such imbalanced training data by retaining more class 1 and 2 codewords than in other classes. The BoF-SD and BoF-MD processing vels with SSVM are respectively 7.5 fps and 6.99 fps. Because of its larger dictionary size, the BoF-MD works slightly lower than BoF-SD. While the overall size of SD is S D = 2000, the total size of MD is S D = S ID · 29 = 2900, which is 45 percent higher than SD.

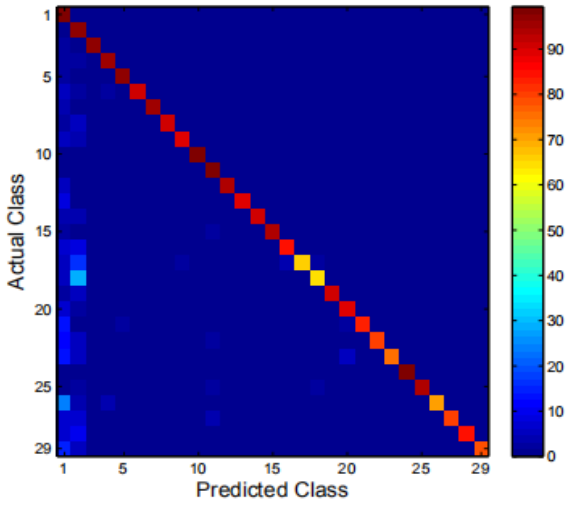
4.5.2 Dictionary Training Time

Even if BoF-MD-ODMS is somewhat less accurate and speed than BoF-SD, it is significantly less than SD's TDT r that the time to build and restore the MD, i.e., the TDT r dictionary training period, is necessary. Figure 4.3 shows that TDT r for MD and SD is extremely different. Contrary to SD, the size of a dictionary doesn't increase rapidly with TDT r of MD. Consequently, the costs of MD training and retraining are significantly lower than SDs in due course. An ODMS system should therefore only recognize those models instead of everything passing by. Re-build dictionaries because of added or removed desired applications in such applications.

Figure 4.8: Confusion Matrices for (a) BoF-SD and (b) BoF-MD, the average values over 10 80-20 Datasets are both SURF and SSVM.



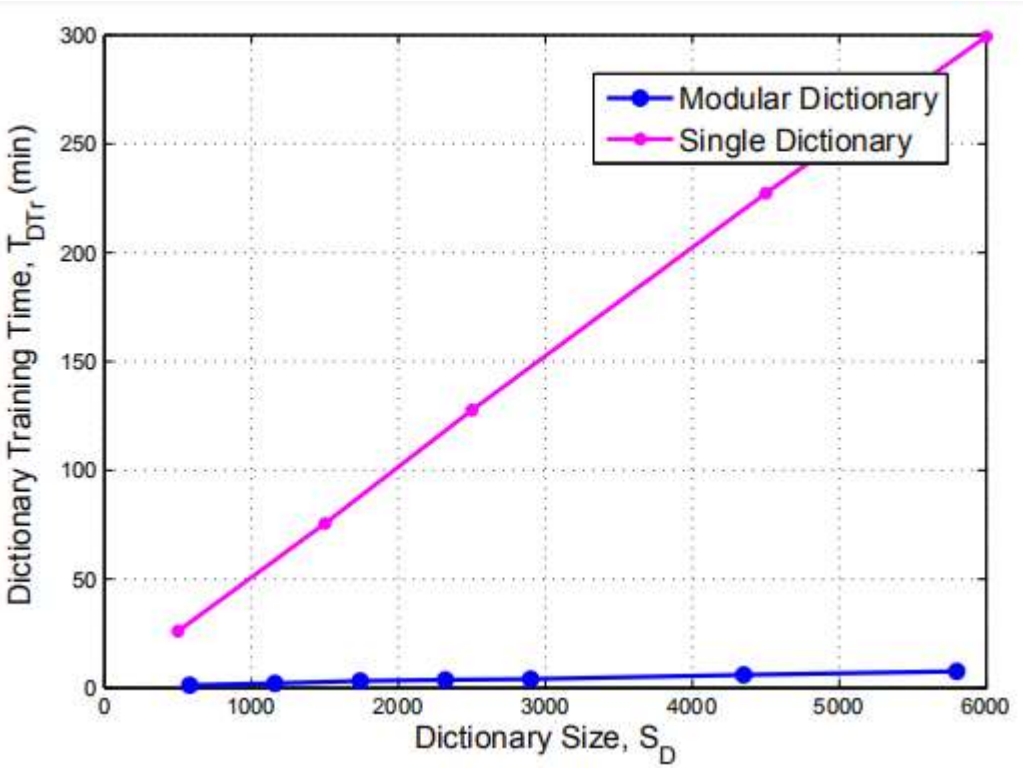
(a)



(b)

The confusion matrix for all 29 modelling classes shown in Figures 4.6(a) and 4.6(b), respectively, using BoF-SD, and BoF-MD approaches (with SURF and SSVM). One can observe that Class 1 (Toyota Altis) and 2 are the most inaccurate categories (Toyota Camry). A significantly greater quantity of training data for these courses could be one of the major reasons of this effect, which could have led to a partial classification. With these unbalanced training data, more coding in Classes 1 and 2 than in other classes might even have been distorted in the dictionary It is necessary to use model classes and BoF-MD is a better choice. But when the reconstruction of the dictionary is unnecessary, BoF-SD is a better option.

Figure 4.9: Dictionary effect (S D) for single and modular dictionaries on dictionary training times (TDT r).



4.6 Computation Time for Feature Extraction and Global Feature Representation

4.6.1 Histogram of Oriented Gradient Based Gist Feature (HOG-gist) Extraction

We will detail our histogram and our building recognition method of oriented gradient base gist feature (HOG-gist).

4.6.1.1. Orientation Gist Feature Extraction.

In the following five steps the orientation dial features can be extracted. The process is as follows:

(1) On the intensity channel I from I = (r + g+b), the image pyramid I(c){c = 0, 1, 2, 3} is produced in four spatial scales ranging from 1:1 (zero-scale) to 1:8 (three-scale) in 4 octaves. In each scale of I(c), a histogram is calculated with an oriented gradient.

(2) For calculation of the horizontal Gx (x, y) and vertical Gy(x, y) gradients of I(c), use the gradient filter [−1, 0, 1] without smoothing.

(3) Magnitude calculation | G(x, y)| and angle | gradient calculation | G, y)| | gradient calculation angle | G. Bueno, J. Salido and F. Dalal and B. Triggs et al 2005, O. D'eniz et al

2011

$$G(x,y)| = \sqrt{G_x(x,y)^2 + G_y(x,y)^2}, \tag{4.3}$$

$$\theta(x,y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right).$$

(4) Calculate a histogram with b guiding bins in 0 digits - 180 digits. Magnitude (|G (x, y) |) whose angle $\theta(x, y)$ is in the same bin is added as that bin.

(5) L2−Hys(Lowe style cut out L2 standard) (N. Dalal, B. Triggs, and al, 2005) can be normalized for histograms. We can have four b-dimensional vectors after calculating all histograms of oriented gradient in four scales. The guiding gist feature vectors of HOG-gist are these four histograms of oriented gradient, namely the orientation gist features of the HOG-gist.

For any real-time application, computer period remains an vital aspect. Since features necessity remain designed aimed at original contribution images, a computational period essential aimed at respectively stage must be evaluated in detail, preliminary through the mining of features of image HOG besides GIST. Through HOG, the image remains separated hooked on overlapping blocks besides considered besides a histogram constructed aimed at to each block. Entirely histograms of the blocks are simply combined to form an image feature vector. We also convert the original image for the GIST, in the other case we are computing the Gabor transform for respectively image hooked on a established of imageries through numerous balances besides orientations. The altered images remain parted hooked on blocks besides the average for each block is calculated. Finally, the image feature vector constructs by means of simple concatenation of averaged grid values.

Table 4.3: Calculation of computation time for HOG and GIST (100 images/ seconds)

Method	Configuration	Feature Extraction
HOG	24 x 6	2.41
	30 x 6	2.74
	33 x 6	2.79
	24 x 9	2.85
	33 x 9	3.28
	45 x 9	3.91
	36 x 12	4.04
	45 x 12	4.56
GIST		8.91

In the second column of the table the HOG block division configuration is specified. For each setup the total time required is given in seconds for every 100 images. The computational time increases by way of growth the quantity of slabs. GIST calculation period remains near dual periods advanced than the major HOG arrangement, besides around four times higher than the lowest HOG setup. For the HOG and GIST features, the training and testing data sets are the same process, so both HOG and GIST are trained and tested during the same time.

Algorithm 4.9: Feature Point Matching Algorithm

Input: Eigenvector set DES1, DES2

Output: Successful matched pairs

1: Initialization des1 ← ∅, des2 ← ∅, pair = 0

2: Use KD-Tree to index all the elements of DES2

3: for i=1, · · ·, m do

4: DES1 = [U1; U2; · · ·; Um], DES2 = [V1; V2; · · · ; Vn]

5: According to BBF algorithm, obtain the approximate 2-nearestneighbors Vj , Vk of Ui which is from DES1 in KD-Tree

6: if $T\left(\frac{d(U_i, V_j)}{d(U_i, V_k)}\right) = 1$ then des1 ← Ui, des2 ← Vj, pair = pair + 1

7: else Remove Ui (no valid matched point)

8: end for

The different side of an image is the focus of each feature description algorithm. Also, the similarity measurement of the local operator and the global function operator is not the same. As the global characteristic vector is one vector, its similarity is easier than its local characteristic. The similarity can be directly determined by the formula of the similarity distance. The similitude measuring function is used as Euclidean distance (Y. H. Zheng, J. Byeungwoo, D. H. Xu, et al 2015):

$$R = \| G1 - G2 \| \tag{4.4}$$

Where R refers to the similitude measure, the global vector of the various images is designated as G1 and G2, and the 2-norm of the CV is indicated by k · k. The most similar points in two images are to be found at the local feature point. Take a key point in an image and find the 2 key points with the smallest similarity to the first. This pair is accepted if the relationship between the nearest distance and the second nearest distance is less than one threshold. Let the ratio = d1/d2, with d1 being the nearest distance and d2 being the second closest.

4.6.2 GIST Feature Extraction

The GIST function represents the image's scene information (T. Xu et al 2014) The image is divided into several blocks before extracting the GIST function. The blocks are handled in advance by Gabor filters of various scales and directions and then average calculated results from different regions to obtain necessary information about characteristics. The image is divided into several blocks in advance to avoid loss of information and improve feature accuracy. Order overlays on several parts and then integrate the results to obtain the image's global GIST feature. Assume that the image to be processed originally is M bins-size. Divide it into nb — according to nb blocks, and each block constitutes a region. Ng = nb — the total number of blocks is recorded with nb = nb. The various blocks of the picture are marked with bb, where I = 1, . ng. Each block is in the same size as M' tall N' to facilitate calculation and processing. With the Gabor filter's self-similarity, a number of mathematical transformations and operations allow different Gabor filters to be obtained when the mother wavelet filter is provided. The Gabor filter mother wavelet is the following:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp[-(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})]\times\cos(2\pi f_0 x+\varphi) \tag{4.5}$$

Transform the mother wavelet mathematically and then a series of Gabor filters can be obtained in different scales and directions.

$$\begin{cases} g_{mn}(x,y) = a^{-m}g(x',y'), a>1 \\ x' = a^{-m}(x\cos\theta + y\sin\theta) \\ y' = a^{-m}(-x\sin\theta + y\cos\theta) \\ \theta = \frac{n\pi}{n+1} \end{cases} \tag{4.6}$$

If a −m denotes the scale factor, the value of −m denotes the angle of rotation, the number of scales, the number of directions, is indicated. After the calculation, there are m to n Gabor filters. Firstly, on the different regions of the original picture the same processing is carried out, followed by a cascade operation. The result is the spiritual characteristic of the image:

$$G_i^B(x,y) = cat(I(x,y) * g_{mn}(x,y)), (x,y) \in B_i \tag{4.7}$$

Where GB denotes a GIST block, its dimension shall be m × n × M' × N', the cascade operation shall be denoted and —to— the convolution operation shall be denoted ∗. The formula (3) in which GB denotes GIST as its block, is the size of the GIST function, its size is m × n × M' × N' cat(). The GIST features of each block are averaged for each individual filter. The results are integrated by rows, which give GBi the mean of GIST blocks in the GIST image function = GB1, GB2., GBng }, in which GG indicates the global GIST track. GBi is as follows calculated:

$$\overline{G_i^B} = \frac{1}{M' \times N'} \sum_{(x,y) \in B_i} G_i^B(x,y) \tag{4.8}$$

As the world image feature is m × n × ng, G^G is extracted. The entire image can only be described by G^G, but the image details cannot be clearly expressed.

4.6.3 SIFT Feature Extraction

The core of the SIFT algorithm is to identify the extreme points and extract invariants in the space in scales.

The specific process of extraction is divided into four steps:

Algorithm 4.10: Extraction algorithm of global GIST feature

Input: Image I(x, y) with the size of M × N

Output: Feature vector GG

1: for j=1, · · · ,m do

2: for k=1, · · · , n do

3: Calculate m × n Gabor filters gjk(x, y) by Formula (4.6)

4: end for

5: end for

6: Divide I(x, y) into ng = nb × nb blocks and label those different blocks with Bi

7: for i=1, · · · , ng do

8:Calculate $G_i^B(x, y)$ by Formula (4.7)

9: Calculate G_i^B by Formula (4.8)

10: end for

11: $G^G = \{\overline{G_1^B}, \overline{G_2^B}, \ldots, \overline{G_{n_g}^B}\}$

Step 1. Scale Space Setup. Firstly, in a pair of images I a gauze pyramid is created in gauze (x, y). I(x, y) is in size M = N, and the first-order space for the s+3 layers of gaussian images. The kernel is defined according to Gaussian convolution.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{4.9}$$

If x, y indicates the pixel position information, μ indicates the factor of the scale. The Gaussian picture below is as follows:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{4.10}$$

The scale factor σ increases in the proportion k from the bottom of the image. where $k = 2^{1/s}$, s is 3. The obtained L(x,y, 25σ) is subsampled to generate the first image from the second Gaussian scale of the size of 1/2M = 1/2N. If you have the fourth Gaussian image generated. The s plus 2 Gaussian (DoG) differences are derived from the subtraction of the s plus 3 sub-images of the Gaussian scale. This is the formula.:

$$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \quad (4.11)$$

Then the sequence scale for the first order Gaussian picture is: σ, k ∗ σ, k 2 ∗ σ, k 3 ∗ σ, k 4 ∗ σ, k 5 ∗ σ. The scale . The corresponding scale sequence is: σ, kσ, k 2 ∗ σ, k 3 ∗ σ, k 4 ∗ σ.. Three layers are selected to reach the maximum value: kσ, k 2 ∗ σ, k 3 ∗ σ. In the same way, treat the second order to obtain the second order DoG space scale sequences: kσ, k 2 ∗ σ, k 3 ∗ σ.At this time, with the first layer in the second order the final layer of the first order Dog space is continuous. The order number of the pyramid shall be determined by the size of the original image and the top image of the pyramid

$$Ordernumber = log_2\{\min(M, N) - t\}, t \in log_2\{\min(M, N)\} \quad (4.12)$$

where t is the logarithm of the minimum dimension of the top image.

Step 2. Key points determination. In dog-scale, compare the value of each pixel with that of the adjacent 26 pixels (9 pixels on the upper layer +8 pixels on the same layer and 9 pixels on the lower layer). If the pixel (x, y) value is the local extreme value, the point will be the SIFT key points set for the candidate. In addition, low contrast points can be found by adapting the three-dimensional quadratic equation. Do the expansion of the Taylor in the D(x, y, σ)area (x0, y0, σ0) in order to reach the quadratic term: The following is:

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \quad (4.13)$$

where $X = (x, y, \sigma)^T$, That is the offset of the points of the sample. The extreme

point Xa can be derived from the equation above

$$\hat{X} = -\frac{\partial^2 D^{-1}}{\partial X^2}\frac{\partial D}{\partial X} \tag{4.14}$$

put Formulation (4.13) into Calculation (4.14):

$$D(\hat{X}) = D + \frac{1}{2}\frac{\partial D^T}{\partial X}\hat{X} \tag{4.15}$$

If | D(Xˆ) |< 0.03, Low contrast point is deleted. The value i.e., D −to X is estimated in actual operation by the difference between the adjacent pixels of the sampled point. Find the edge point and remove the Hessian matrix. The following is the Hessian matrix:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{4.16}$$

where D_{xx}, D_{xy}, D_{yx}, D_{yy} the adjacent sampling difference can be estimated. Suppose that α and β indicate H values, which record the x and y gradients. The following can be calculated for the traces and determinants of the Hessian matrix:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{4.17}$$

$$Det(H) = D_{xx}D_{yy} - D_{xy}^2 = \alpha\beta \tag{4.18}$$

If α is the larger value of H and β is the smaller value, then let α = rβ:

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{4.19}$$

If r = 1, then (r+1)2/r reaches a minimum, with an increase from r, the value of (r+1)2/r increases. Larger r value means that α and β are more different. In other words, one gradient value is large and the other small, i.e. the edge information. The

threshold approach is used to assess the threshold value. If the formula meets a point $\frac{Tr(H)^2}{Det(H)} > \frac{(r+1)^2}{r}$, This item is deleted and r = 10.

Step 3. Determination of the key points' direction parameters. The sampling points that are located on the vicinity window of each key point are calculated in the gradient and direction. The formula for calculation is:

$$Grad(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{4.20}$$

$$\varphi(x,y) = \tan^{-1}\left(\frac{L(x+1,y)-L(x-1,y)}{L(x,y+1)-L(x,y-1)}\right) \tag{4.21}$$

Grad(x, y) indicates the magnitude of the gradient, where Grad(x, y) indicates the direction of the gradient. The main direction of the key point is the gradient histogram of 36 directions, while the direction of the histogram peak. A gaussian distribution function is weighted for the gradient direction information contribution of the sampling points to the core point. The gradient of the sampling point multiplies this function to represent weight. In relation to the direction of the key, whose value is more than 80 per cent of the peak in the histogram, the key is copied in various directions. A key point is currently found on the scale and specific orientation.

Step 4. Description of key point feature. Turn the axis in a pixel direction and then take the pixel in the centre to obtain the window 16 x 16. The window has 4 sub-windows each, respectively. Use the Gaussian blur technique to increase the weight of the neighborhood points closer to the main point and to reduce those distant points' weight values. To obtain 4 x4x 4 = 8 = 128-dimensional feature descriptor, calculated the accumulation gradient of 8 directions in each sub region.

.

Algorithm 4.11: Extraction algorithm of local SIFT feature

Input: Image I(x, y) with the size of M × N

Output: Eigenvector set des

1: Initialization des ← Ø

2: Calculate the Gaussian pyramid order n by the Formula (4.7)

3: for i=1, · · · , n do

4: for j=1, · · · ,m do

5: Calculate the m + 1 Gaussian images with the scale factor coefficient of k j according to Formula (4.8)

6: end for

7: Sub-sample the m – 1

8: Calculate D = $D = \{L_{k+1} - L_k\}_{k=2}^{m-1}$ by Formula (4.9)

9: Get extreme points D (x0, y0, σ0) of D = {L_{k+1} – L_k} $D = \{L_{k+1} - L_k\}_{k=2}^{m-1}$ m−1 k=2 and remove the edge points remove the edge points by Formula (4.11) -((4.17))

10: Calculate Grad $\{Grad_i\}_{i=1}^{36}$ and $Grad_k = \max(\{Grad_i\}_{i=1}^{36})$ ϕ of a certain neighborhood pixel for the key points by Formulas (4.18) and (4.19) and Gaussian weighted gradient the direction of key point is the direction of $Grad_k$

11: Generate desi by Step (4), des ← des S des_i

12: end for

Table 4.4: Calculation of Computation time for SIFT and SURF (seconds/100 images)

Method	Dictionary Size	Feature Extraction (1)	Dictionary Creation (2)	Encoding (3)	Training (1+2+3)	Testing (1+3)
SIFT	1600	9.51	4.23	16.29	30.03	25.80
	1800	9.51	4.52	17.87	31.90	27.38
	2000	9.51	4.83	19.49	33.83	29.00
	2200	9.51	5.16	21.19	35.86	30.70
	2400	9.51	5.55	23.63	38.69	33.14
	2600	9.51	6.03	24.89	40.43	34.40
	2800	9.51	6.42	27.18	43.11	36.69
	3000	9.51	6.84	28.69	45.04	38.20
SURF	1600	6.42	1.26	6.42	14.10	12.84
	1800	6.42	1.29	7.06	14.77	13.48
	2000	6.42	1.35	7.75	15.52	14.17
	2200	6.42	1.44	8.48	16.34	14.90
	2400	6.42	1.48	9.32	17.22	15.74
	2600	6.42	1.53	10.01	17.96	16.43
	2800	6.42	1.59	10.74	18.75	17.16
	3000	6.42	1.65	11.59	19.66	18.01

With an additional step of 200, we create a set of dictionaries with the sizes of 1600 to 3000 (visual words) for the SIFT and SURF features. The KNN Clustering based on the training dataset defines words with visual component meanings. Since the same dictionary is used to encode test images, a new dictionary for the test dataset is not required. The time required for calculating images for a training dataset includes the time needed for extracting, creating a dictionary and encoding functions. The computational time to encode a test dataset is only the time to extract features and encode, as dictionary creation is not needed for the testing data set. Table 4.4 provides the time to calculate SIFT and SURF. The second column lists dictionary sizes. Aimed at functional mining, dictionary making besides encrypting stages, the period aimed at

calculation stakes 3, 4 besides 5 aimed at 100 images, correspondingly. Popular the end two stakes of table correspondingly the total amount of time needed to process 100 images from training and test images is shown. The time required for processing 100 images is reported as all processing times specified in Table 4.4. These values are achieved by means of an average of processing time for the complete data set processing. The tables show that SIFT and SURF calculation times are much longer than the ones mandatory aimed at HOG in addition GIST. HOG necessitates the bottommost period and SIFT the highest calculation time.

4.7. Random Forest

The algorithm for learning by the Random Forest Machines is based on the production of a random forest. A arbitrarily nominated subsection of exercise data besides a aimlessly designated characteristic are used for every decision tree (features). Each random forest decision tree is travelled during classification for every vector input, and every tree forecasts the participation direction class built taking place the situation own configuration. Then prediction outcomes stay collective, besides an productivity class remains nominated by means of popular elective.

Figure 4.10 Process of Random Forest Algorithm for ODMS

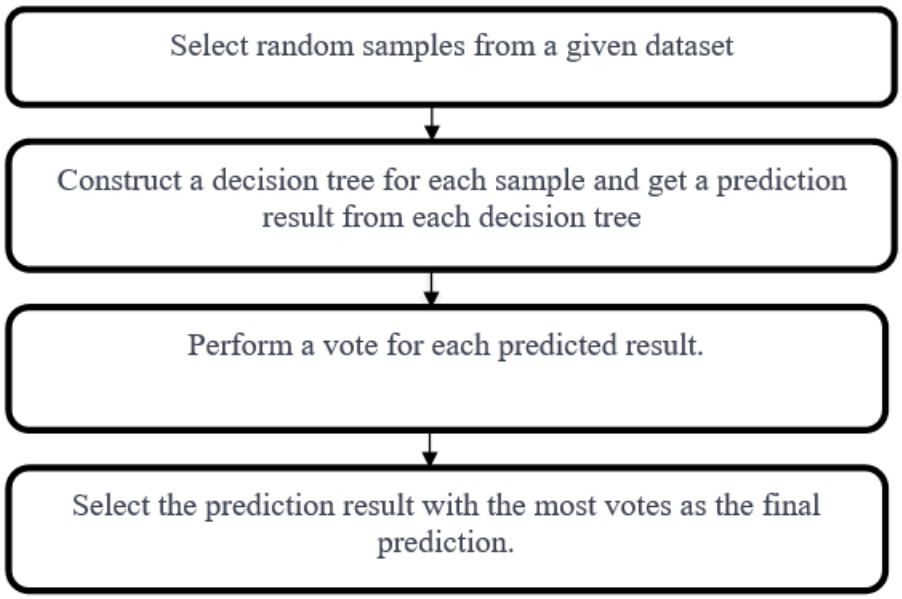
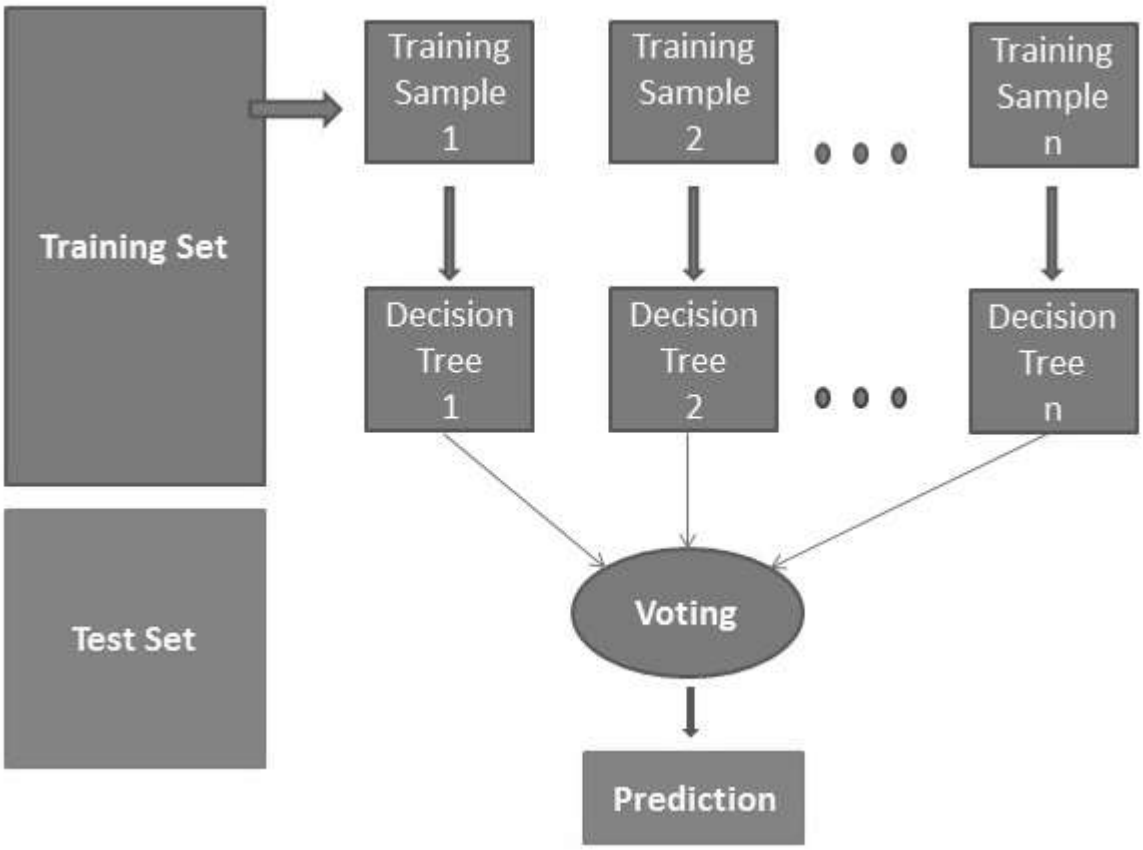


Figure 4.11: Demonstrates the Working the Random Forest Algorithm for ODMS

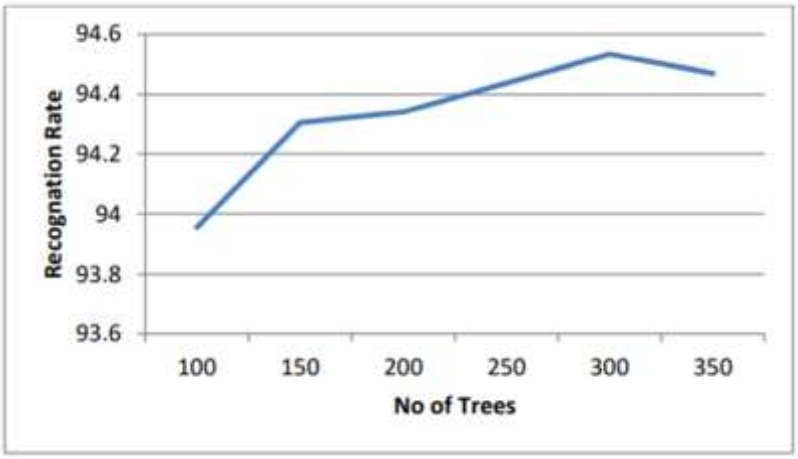


Random Forest uses several classification decisions trees. In case of an x test data, each decision tree for the input test data is crossed. Respectively tree varieties the situation choice approximately the involvement facts class. Thus, the subsequent stage stands towards convey collected entirely these scrawny trees' decisions for the final output. The algorithm for the final decision is used by Random Forest. The results of application of the ODMS random forest-based system to the proposed heterogeneous dataset can be discussed in this section. Deliberate the recognition frequency towards demonstrate the perceived precision of the ODMS and report the time for the test dataset. The Random Forest has been trained with 100, 150, 200, 250, 300 and 350 decision-making trees. In order to refer to the Random Forest Classification system, we use the term RF-ODMS.

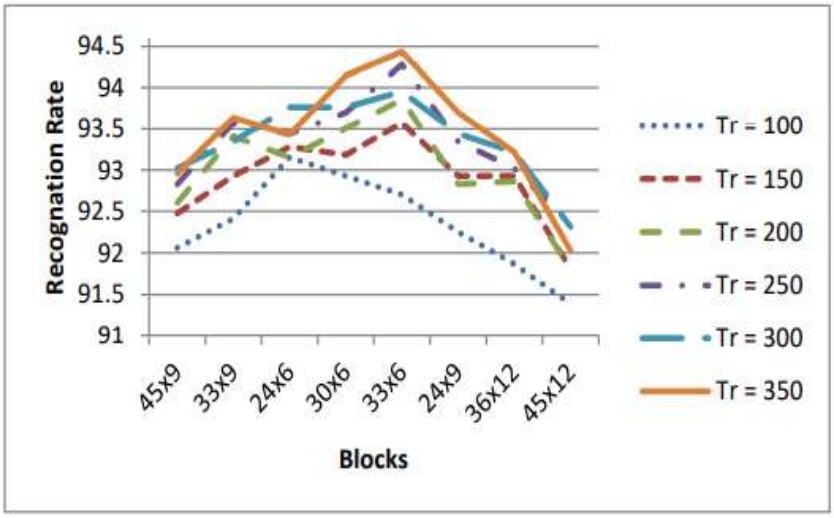
The two other important parameters for the random training of forests are the number of randomly selected (with replacement) samples for respectively decision tree besides the

quantity of casually nominated characteristics toward be considered for each decision tree. During the testing phase, the similar restrictions remain cast-off. noting that the recognition frequency reduced after the exercise subcategory aimed at separately Conclusion Tree remained compact besides the similar occurs once the exercise dataset aimed at the entire random forest remained compact. Consequently cast-off completely training information toward produce administrative trees. Similarly, create that the nominated features of Random Forest improved obey through square root of universal RF-ODMS characteristic. RF-ODMS credit tolls revealed trendy Number 4-6 aimed at a total of twenty-five data sets represented by IV feature mining methods GIST, HOG, SIFT besides SURF. Altogether 4 facts signify the perpendicular axis through the RF-ODMS finding amount although the parallel axis remains dissimilar. The parallel axis demonstrations the quantity of decision trees cast-off for casual forestry exercise in GIST characteristics as shown in Figure 4.12(a). The 4.12(b) horizontal axis is the number of blocks used to make HOG image characteristics. The horizontal axis of 4.12(c) and (d) represents the size of the SIFT and SURF dictionary. Different styled HOG, SIFT and SURF characteristics are indicated by different recognition rates for the random forest configuration (Number of Decision Trees) in Figure 4.12 b), (c) and (d). The detailed overview of test times is presented in Table 4.6. The period exposed in the bench demonstrations the period required near procedure 100 imageries general testing phase. the initial six supports of the table the calculation time is taken for Random Forest Tests: in the first column, 100 decision-making trees are processable; in the second column, 100 images are processed through one fifty executive trees besides consequently on. A end column demonstrations a period wanted towards excerpt image features besides put on a one hundred imageries towards global depiction.

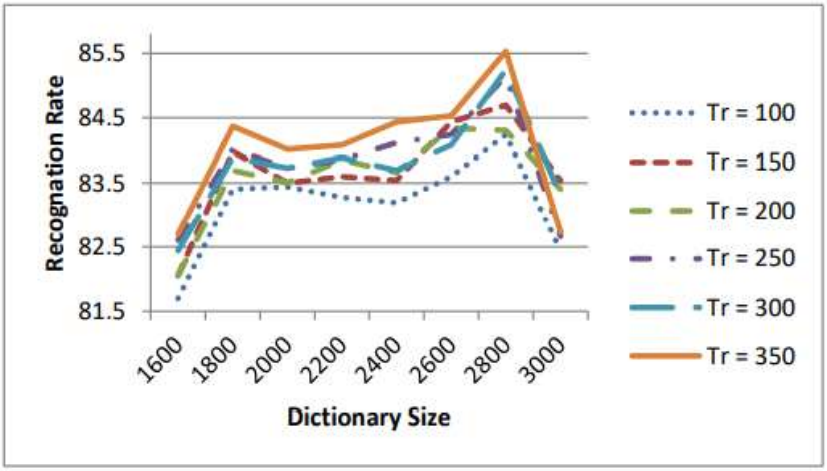
Figure 4.12: Demonstrates the Recognition rate for RF-ODMS



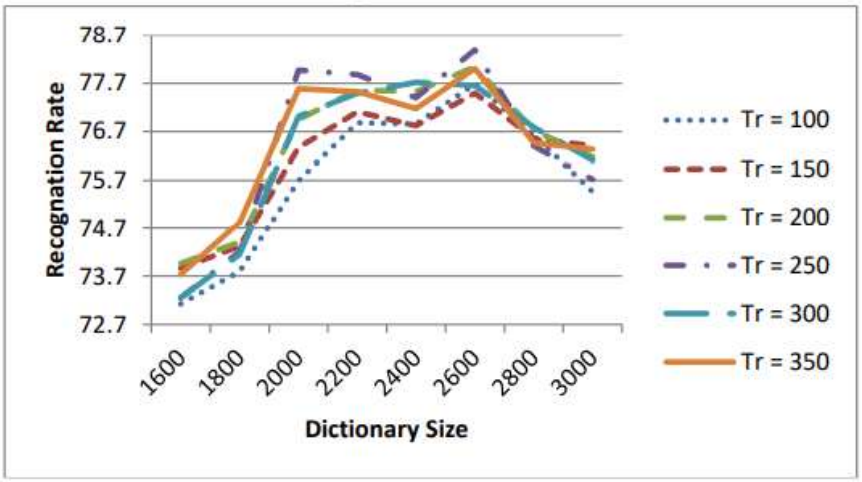
(a) GIST features



(b) HOG Features



(c) SIFT Features



(d) SURF Features

Table 4.5 Demonstrates the RF-ODMS testing time (seconds/100 images)

	Configuration	Trees 100	Trees 150	Trees 200	Trees 250	Trees 300	Trees 350	Feature Extraction
GIST		0.005	0.008	0.010	0.014	0.018	0.005	8.91
HOG	24 x 6	0.006	0.009	0.012	0.013	0.014	0.015	2.41
	30 x 6	0.008	0.009	0.011	0.014	0.016	0.017	2.74
	33 x 6	0.008	0.009	0.012	0.013	0.016	0.017	2.79
	24 x 9	0.007	0.010	0.011	0.016	0.016	0.017	2.85
	33 x 9	0.008	0.011	0.014	0.016	0.018	0.019	3.28
	45 x 9	0.007	0.012	0.011	0.013	0.016	0.018	3.91
	36 x 12	0.007	0.010	0.011	0.014	0.015	0.018	4.04
	45 x 12	0.007	0.009	0.011	0.014	0.016	0.018	4.56
SIFT	1600	0.011	0.012	0.014	0.019	0.025	0.027	25.80
	1800	0.011	0.016	0.017	0.020	0.024	0.027	27.38
	2000	0.012	0.012	0.015	0.018	0.021	0.028	29.00
	2200	0.009	0.012	0.015	0.019	0.022	0.028	30.70
	2400	0.010	0.012	0.016	0.019	0.022	0.028	33.14
	2600	0.010	0.013	0.017	0.019	0.022	0.028	34.40
	2800	0.011	0.014	0.016	0.021	0.025	0.029	36.69
	3000	0.012	0.015	0.017	0.020	0.024	0.029	38.20
	1600	0.013	0.016	0.022	0.024	0.029	0.030	12.84
	1800	0.014	0.018	0.021	0.026	0.029	0.030	13.48
	2000	0.013	0.017	0.021	0.026	0.032	0.034	14.17
	2200	0.016	0.021	0.024	0.029	0.033	0.034	14.90
	2400	0.016	0.020	0.025	0.031	0.035	0.036	15.74

SURF	2600	0.015	0.020	0.026	0.033	0.035	0.036	16.43
	2800	0.015	0.020	0.026	0.030	0.035	0.038	17.16
	3000	0.018	0.023	0.029	0.035	0.041	0.042	18.01

Table 4-6: Demonstrates the RF-ODMS testing processing speed (Frames/Second)

	Configuration	Trees 100	Trees 150	Trees 200	Trees 250	Trees 300	Trees 350
GIST		11.2	11.2	11.2	11.2	11.2	11.2
HOG	24 x 6	41.4	41.3	41.3	41.3	41.2	41.4
	30 x 6	36.4	36.4	36.4	36.3	36.3	36.4
	33 x 6	35.7	35.7	35.7	35.7	35.6	35.8
	24 x 9	35.0	35.0	35.0	34.9	34.9	35.0
	33 x 9	30.4	30.4	30.4	30.3	30.3	30.4
	45 x 9	25.5	25.5	25.5	25.5	25.5	25.5
	36 x 12	24.7	24.7	24.7	24.7	24.7	24.7
	45 x 12	21.9	21.9	21.9	21.9	21.9	21.9
SIFT	1600	3.9	3.9	3.9	3.9	3.9	3.9
	1800	3.7	3.7	3.7	3.6	3.6	3.7
	2000	3.4	3.4	3.4	3.4	3.4	3.4
	2200	3.3	3.3	3.3	3.3	3.3	3.3
	2400	3.0	3.0	3.0	3.0	3.0	3.0
	2600	2.9	2.9	2.9	2.9	2.9	2.9
	2800	2.7	2.7	2.7	2.7	2.7	2.7
	3000	2.6	2.6	2.6	2.6	2.6	2.6
SURF	1600	7.8	7.8	7.8	7.8	7.8	7.8
	1800	7.4	7.4	7.4	7.4	7.4	7.4
	2000	7.1	7.0	7.0	7.0	7.0	7.1
	2200	6.7	6.7	6.7	6.7	6.7	6.7
	2400	6.3	6.3	6.3	6.3	6.3	6.3
	2600	6.1	6.1	6.1	6.1	6.1	6.1

	2800	5.8	5.8	5.8	5.8	5.8	5.8
	3000	5.5	5.5	5.5	5.5	5.5	5.5

Table 4.5 measures all times in seconds. Finally, Table 4.6 provides the processing speeds per second for the test data set with regard to frames the processing speeds are measured on the basis of the accumulated values of the extraction, the global function representation time and the RF-ODMS test time.

4.8. Results Using ODMS- Ensemble of Multi-Class Support Vector Machine

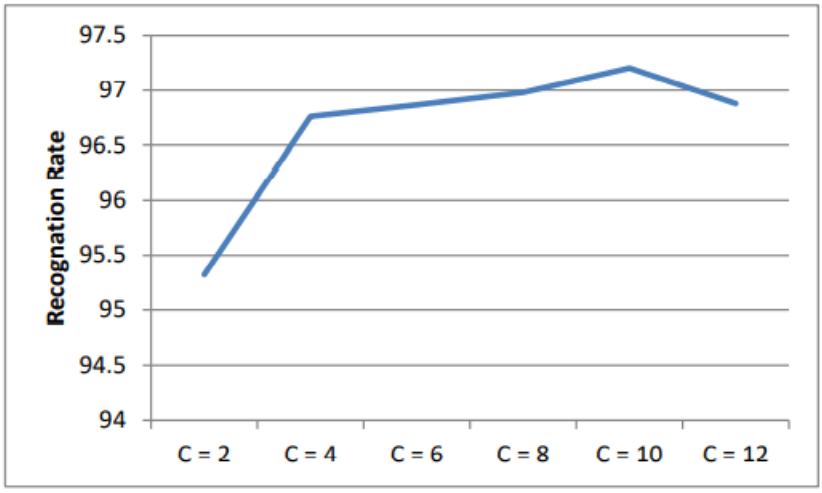
Thus, the algorithm of SVM generates numerous dual SVMs aimed at multiclass arrangement problematic; The outcomes towards SVM multi-class set remain cast-off to make a final decision on the incoming observation. This is done by means of a multiclass support system (SVM). In this section we discuss the performance of the multi-class ODMS support system with ODMS. We discuss the accuracy of the recognition rate and the computing time required for the test dataset. We are using a linear kernel to transform the training dataset with a number of kernels. A linear kernel is a good option if the functional count exceeds the number of observations. When the ODMS system was applied in the data set proposed, the numeral of structures aimed at respectively comment, reliant scheduled the method besides arrangement for extraction, varied between 512 and 5000. In the end, only one regularisation parameter C needs to be optimised for the linear kernel. The parameter C for regulating the hyperplane margin between the two data classes is used. A bigger value for C means that the two classes have a smaller gap. We have trained C = 2, 4,6, 8, 10 and 12 Support Vector Machine. We use the term ABSVM-ODMS to refer to the ODMS system with the Multi-Class Support Classification of Vector Machines.

The ABSVM-ODMS system detection rates can be seen in Figure 4.13 with

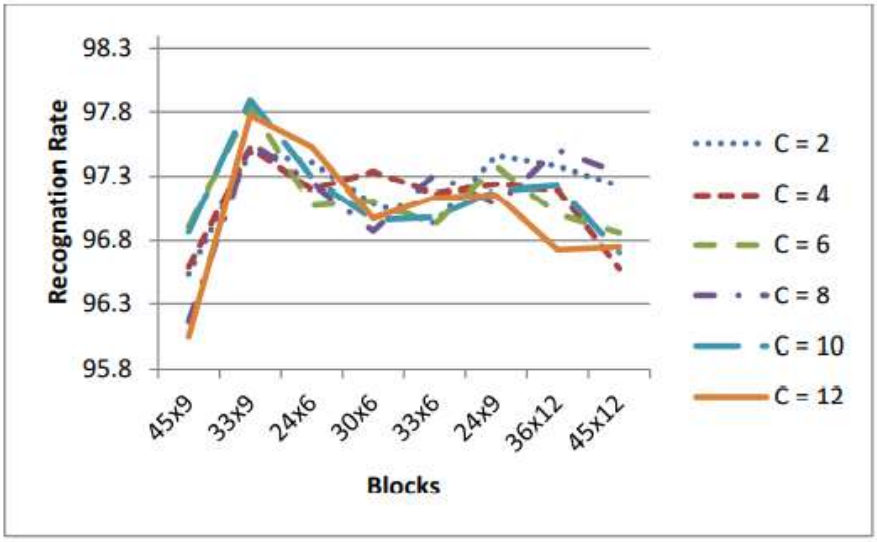
a total of 25 data representations for four different techniques of extraction of features GIST, HOG, SIFT, then SURF. A perpendicular axis of the 4 facts demonstrations the acknowledgement frequency aimed at SVM-ODMS, while the horizontal axis shows different data. The horizontal axis shows the range size used in the SVM training for GIST function, as shown in Figure 4-4.13(a). Figure 4.13 (b) shows the number of blocks that are used to build the HOG picture characteristics. The horizontal axes of Figure 4.13(c) and (d) represent the SIFT and SURF dictionaries. Dissimilar shaping lines aimed at the HOG, SIFT besides SURF structures trendy Number 4.13 (b), (c) besides the recognition rate are shown for various margin-sized configurations (ABSVM) (d).

The amount of time required to process a century of images during the testing period is displayed in the table. The first six columns on the table present the time needed to process one hundred images with C = 2, the second column provides the time necessary for computing one hundred images with C = 4 and so on. A detailed overview of test time is presented in Figure 4.13. The time is given in the table showing the calculation time required in the testing phase to process a hundred images. For the SVM trying stage, the primary six stakes popular table demonstration the computation period; the first column shows the period required aimed at dispensation a one hundred imageries through C=2 a next stake displays a period required to process a hundred imageries through C=4 besides subsequently.

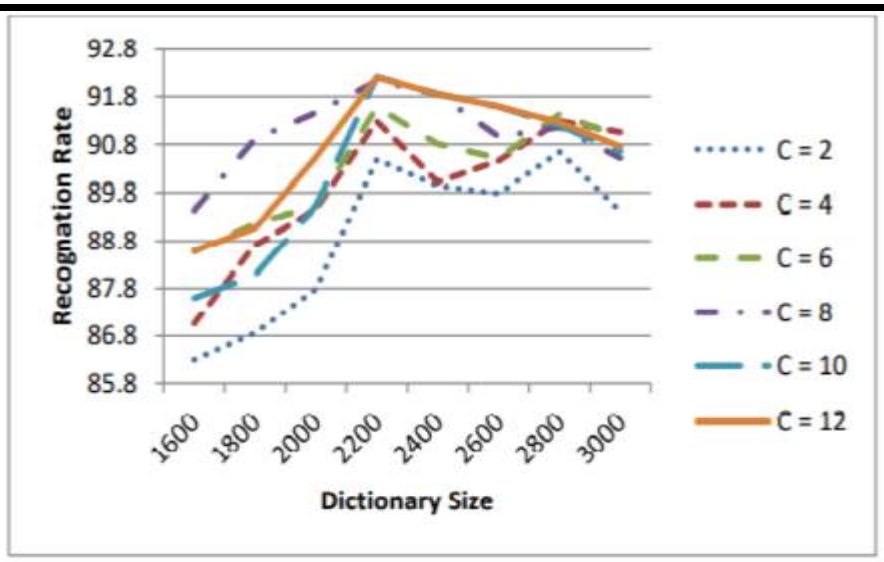
Figure 413: Demonstrates the Recognition Rate for ABSVMB ODMS



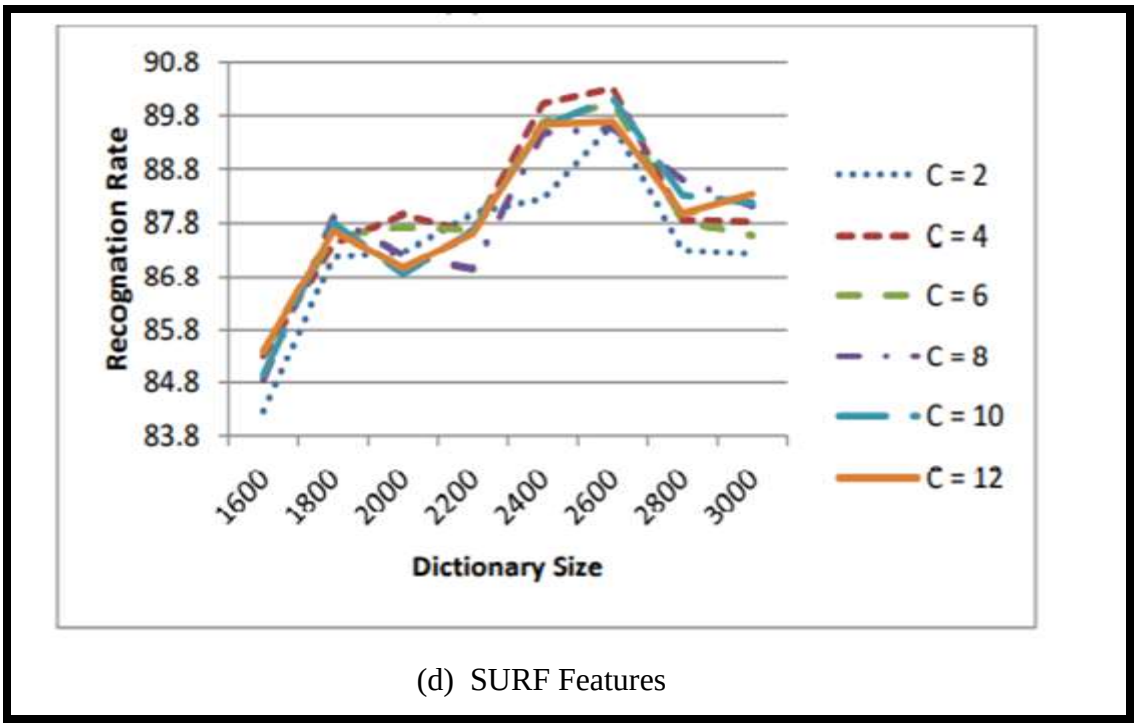
(a) GIST features



(b) HOG Features



(c) SIFT Features



(d) SURF Features

Table 4-7: Demonstrates the ABSVM-ODMS Testing Time (seconds/100 images)

	Configuration	Margin C = 2	Margin C = 4	Margin C = 6	Margin C = 8	Margin C = 10	Margin C = 12	Feature Extraction
GIST		0.207	0.268	0.269	0.267	0.269	0.268	8.91
HOG	24 x 6	2.007	2.009	2.011	2.014	2.016	2.005	2.41
	30 x 6	2.008	2.011	2.012	2.010	2.016	2.173	2.74
	33 x 6	2.609	2.609	2.586	2.604	2.566	2.627	2.79
	24 x 9	2.610	2.627	2.562	2.619	2.611	2.620	2.85
	33 x 9	3.272	3.247	3.267	3.258	3.256	3.273	3.28
	45 x 9	3.646	3.660	3.649	3.646	3.642	3.663	3.91
	36 x 12	4.659	4.573	4.545	4.557	4.553	4.569	4.04
	45 x 12	5.136	5.070	5.128	5.131	5.154	5.145	4.56
SIFT	1600	0.862	0.853	0.853	0.854	0.848	0.850	25.80
	1800	0.929	0.923	0.911	0.915	0.917	0.922	27.38
	2000	0.982	0.976	0.976	0.979	0.978	0.975	29.00
	2200	1.044	1.032	1.041	1.046	1.043	1.037	30.70
	2400	1.104	1.104	1.104	1.102	1.102	1.097	33.14
	2600	1.181	1.164	1.172	1.169	1.178	1.172	34.40
	2800	1.243	1.243	1.240	1.241	1.233	1.238	36.69
	3000	1.312	1.299	1.308	1.296	1.301	1.305	38.20
SURF	1600	0.763	0.747	0.751	0.743	0.745	0.740	12.84
	1800	0.811	0.797	0.795	0.790	0.792	0.794	13.48
	2000	0.871	0.868	0.857	0.867	0.873	0.857	14.17
	2200	0.937	0.917	0.930	0.920	0.919	0.928	14.90
	2400	0.996	0.996	0.998	0.994	0.998	0.985	15.74
	2600	1.053	1.055	1.050	1.046	1.055	1.053	16.43

	2800	1.117	1.119	1.120	1.112	1.112	1.120	17.16
	3000	1.213	1.195	1.198	1.199	1.197	1.191	18.01

Table 4.8: Demonstrates the ABSVM-ODMS Testing Processing Speed (Frames/Second)

	Configuration	Margin C = 2	Margin C = 4	Margin C = 6	Margin C = 8	Margin C = 10	Margin C = 12
GIST		11.0	10.9	10.9	10.9	10.9	10.9
HOG	24 x 6	22.6	22.6	22.6	22.6	22.6	22.7
	30 x 6	21.1	21.0	21.0	21.1	21.0	21.1
	33 x 6	18.5	18.5	18.5	18.5	18.5	18.5
	24 x 9	18.3	18.3	18.3	18.3	18.3	18.3
	33 x 9	15.9	15.9	15.9	15.9	15.9	15.9
	45 x 9	13.9	13.9	13.9	13.9	13.9	13.9
	36 x 12	11.5	11.6	11.6	11.6	11.6	11.6
	45 x 12	10.3	10.3	10.3	10.3	10.3	10.3
SIFT	1600	3.8	3.8	3.8	3.8	3.8	3.8
	1800	3.5	3.5	3.5	3.5	3.5	3.5
	2000	3.3	3.3	3.3	3.3	3.3	3.3
	2200	3.2	3.2	3.2	3.1	3.2	3.2
	2400	2.9	2.9	2.9	2.9	2.9	2.9
	2600	2.8	2.8	2.8	2.8	2.8	2.8
	2800	2.6	2.6	2.6	2.6	2.6	2.6
	3000	2.5	2.5	2.5	2.5	2.5	2.5
	1600	7.4	7.4	7.4	7.4	7.4	7.4

SURF	1800	7.0	7.0	7.0	7.0	7.0	7.0
	2000	6.6	6.6	6.7	6.7	6.6	6.7
	2200	6.3	6.3	6.3	6.3	6.3	6.3
	2400	6.0	6.0	6.0	6.0	6.0	6.0
	2600	5.7	5.7	5.7	5.7	5.7	5.7
	2800	5.5	5.5	5.5	5.5	5.5	5.5
	3000	5.2	5.2	5.2	5.2	5.2	5.2

4.9. Explanations on ODMS Performance

Two main elements are the functionality of the ODMS system. The first element is the image feature vector (data) representation that includes feature extraction techniques and techniques for global representation of features. The second element is the classification algorithm for machine learning. The ODMS system is affected by both functionality representation and machine learning algorithm. Observing ABSVM-ODMS and RF-ODMS precision in Figs 4.12 and 4.13, we can see that for all the four-vehicle classification feature extraction methods, ABSVM-ODMS performs better than RF-ODMS.

The four feature extraction techniques with various configurations have similar accuracy behaviors, irrespective of whether used for RF-ODMS or ABSVM-ODMS. SURF is followed by SIFT at the lowest detection rate. Approximately of the HOG besides GIST structures take the finest discovery degree, although the additional HOG depictions take a much higher detection degree than SIFT otherwise SURF. A rise popular the quantity of conclusion tree types trendy a Random Forest system firstly upsurges a recognition frequency, although the rise in the quantity of conclusion tree detailed afterward convinced thresholds marks a recognition concert. Though this boundary remains non established aimed at altogether differences of information groups, the acknowledgement rate reduced popular maximum cases after 300 decision-making treaties. The pattern of behavior by which the recognition rate increases and the number of decision trees decreases with each technique of extraction and

variation can be seen. Although the overall recognition rate is small, it follows the same behavioral pattern. Similarly, with SIFT/SURF increasing in dictionary sizes, the recognition rate increases to and then decreases for both ABSVM-ODMS and RF-ODMS.

In RF-ODMS, SIFT has a dictionary size of 2 800 words and a dictionary size of 2600 words is the best recognition rate for SURF. SIFT achieves its best rating for ABSVM-ODMS with a 2200-word dictionary size and the best rates of recognition for SURF are 2600 words.

HOG-based ODMS follows a similar pattern, which increases the number of blocks to a certain point, then decreases their recognition rate. With 33x6 blocks for HOG RF-ODMS and 33x9 blocks for ABSVM-ODMS, HOG achieved its best recognition rate. The highest rates of recognition for GIST, HOG, SIFT and SURF are 94,53% (Trees = 300), 94,43% (Trees = 33 x 6), 85,53% (Trees = 2800, Decisions Trees = 350), For RF-ODMS respectively, and 78.39% (Dictionary size = 2600; Trees Decision = 250). The highest rates of recognition for GIST, HOG, SIFT and SURF are 97.20 percent (C = 10), 97.89 percent (Blocks = 33x9, C = 10) and 92.20 percent (Dictionary size= 2200, C = 12). Based on our rates of recognition, we can conclude that GIST and HOG perform similar results with respect to RF-ODMS and ABSVM-ODMS, both of which are higher than SIFT and SURF. besides 78.39% correspondingly, aimed at RF-ODMS. A finest acknowledgment charges aimed at GIST, HOG, SIFT besides SURF remain 97.20% , 97.89% ,92.20% besides 90.11% correspondingly, aimed at SVM-ODMS. Container achieve happening the base the acknowledgement frequency that GIST besides HOG achieve correspondingly aimed at RF-ODMS besides ABSVM-ODMS popular relationships of acknowledgement frequency besides they together achieve improved than SIFT besides SURF structures. Since acknowledgement charges, we experimental that 2 important opinion mining methods for ODMS systems are not good. The ODMS system is a key performance factor in the real time processing speed. For all extraction techniques, but for HOG functions in the testing datasets, the processing velocity of RF-ODMS and ABSVM-ODMS is nearly equal; the RF-ODMS processing time almost doubles the speed of ABSVM-ODMS. The standards for Random Forest then C edge aimed at svm for number of decision trees consume actual tiny consequence happening the period aimed at calculation throughout the assessment stage, by

way of exposed popular. Likewise, by way of definite giving out rapidity throughout the stimulating stage prepares non importantly interrupt fluctuations popular trees figures before boundary C. GIST besides HOG sorting remains too closer to SIFT besides SURF classification functions. For Random Forest, the time needed to train and test each of these features is almost the same and the support for Vector Machine varies little. The time required for extraction of functionality, particularly the global display of functionality, creates an enormous gap between SIFT & SURF and HOG&GIST computational time which results in less processing speed of SIFT and SURF based ODMS systems.

ODMS processing speed is 35,7 frame per second for the HOG (33 x 6 blocks), with recognition rate of 87,43%, while ABSVM-ODMS is 15,9 frames per second, while the processing speed of the HOG features is 97,89%. GIST features RF-ODMS and ABSVM-ODMS systems yield a similar rate of recognition for both systems. The processing speed with GIST characteristics is lower than with HOG characteristics for the both systems. Table 4-3 and Table 4-5 also show that extraction of features requires more calculation time than the run-time of the machine algorithm, except for HOG-based ABSVM-ODMS, where the time of the machine learning algorithm is nearly equal to the time of extraction of features. Since there is no inter-dependence between pictures and every incoming image is processed separately, it is possible to parallel the feature extractions step in order to increase ODMS system processing. Another way to evaluate classification accuracy is to measure the highest 3 acknowledgement charges. Aimed at RF-ODMS through GIST besides HOG aimed at ABSVM-ODMS through GIST and for HOG(Blocks=33x9) and decision trees=300, we compute the Top-3 recognition rate for RF-ODMS with GIST (Block=33x9). We have been using altogether 4 outlines towards generate misperception matrices, compute class acknowledgement frequency besides regulate the impression scheduled the acknowledgement level of the training dataset size. The top 3 rates for HOGs (RF-ODMS), GISTs (RF-ODMS), HOGs (ABSVM-ODMS), GISTs and GISTs are 98.91, 98.88, 96.66 and 96.81 (ABSVM-ODMS), respectively. Overall, the recognition rate of GIST in RF-ODMS is slightly higher than that of HoG, but the processing rate of HIG is faster, which makes HoG the choice.

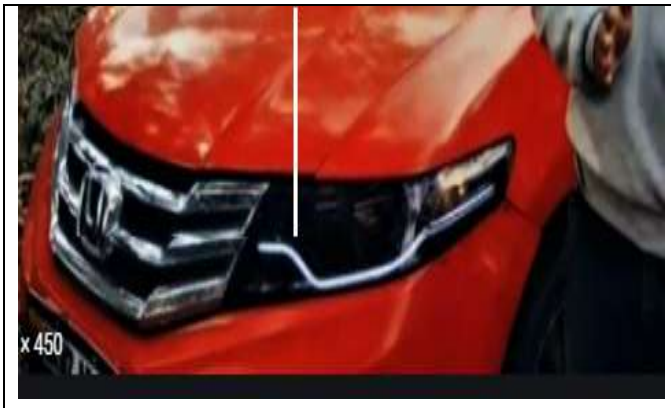
As far as recognition and processing speed is concerned, the proposed RF-ODMS and ABSVM-ODMS systems are comparable to seven other ODMS approaches in Table 4.9. All other current ODMS approaches are model-based, so we calculate vehicle classification results on the basis of the model. In terms of recognition rate and processing speed, our ODMS systems outperform other systems. The results of the ODMS systems we propose are shown in Table 4-9 and are the best results of all variations in functionality extraction and machine learning. In terms of recognition rates and processing speed our ABSVM-ODMS system performs both of their systems. The main features used for previous approaches such as SIFT, SURF, borders, angles, etc. are used to represent the pictures, which may explain why they are less efficient. For key point-based representations in our work we observed a lower performance. (Chen et al. 2015) used grid based symmetrical SURF functionality (such as HOG) with a sparse display and ABSVM grade, which may either be due to symmetrical SURF features or to sparse displays.

Table 4-9: Demonstrates the Assessment the ODMS effort through additional methods trendy relations recognition frequency besides dispensation rapidity

Method	**Feature Extraction**	**Classification**	**Dataset**	**Recognition Rate**	**Processing Speed (fps)**
Dlagnekov et al. (2005)	SIFT	Brute-force matching	790 vehicle images with 400x200 resolution	90.52%	2.19
Munroe and Madden et al (2005)	Canny edges	KNN, neural network, C4.5 decision trees	150 vehicle images with 5 classes	67.33%	0.93

Pearce and Pears et al (2011)	Canny edges, SMG, Harris corners,	K-mean nearest neighbors and Naïve Bayes	177 images from 21 vehicle classes	85.90%	1.73
Jang and Turk et al (2011)	SURF	Matching using Lucene search engine library + structural verification	16 different viewpoints for each toy car.	92.20%	3.1

Figure 4-14: Demonstrates the ODMS stimulating effects (a, b) obstacle as car underneath obstruction, (c, d) incompletely obtainable of photographic camera sight, (e) little illumination, (f) undesirable replication happening hood



(a)



(b)



(c)



(d)



(e)



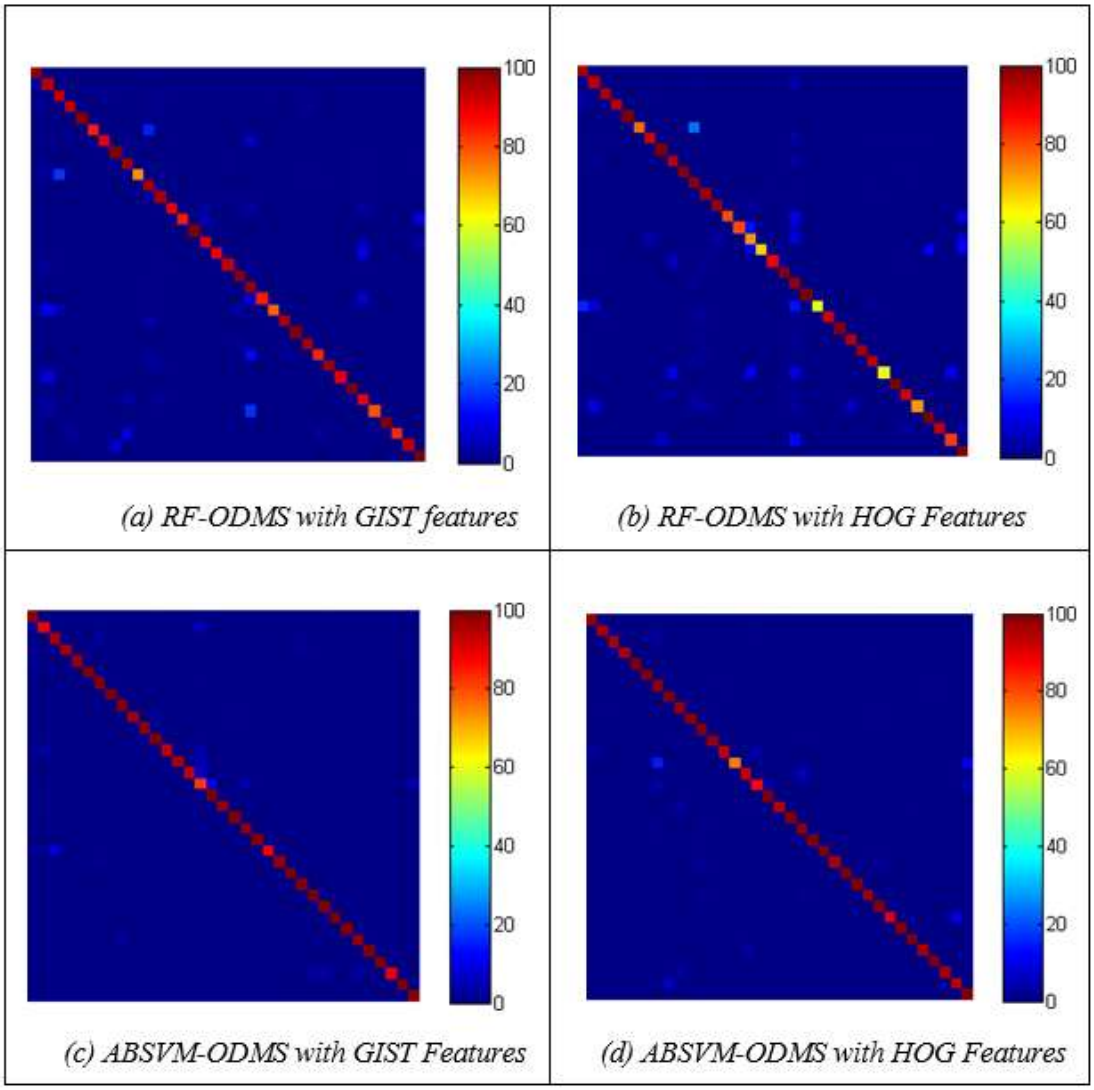
(f)

In challenging situations such as partially occcluded, partially outside the image frame, or low visibility due to low lighting, our proposed ODMS system works well. Figure 4-14

presents some of the challenges that have been successfully identified. Figures 4-14(a) and (b), respectively, show that the vehicle is partly obscured by a footstool. The vehicles are somewhat out of camera view in Figures 4-14 (c) and (d); the light in Figure 4-6 (e) is extremely poor. The vehicle is picked up in 4-14 (f) with an unexpected mirror of the vehicular hood.

4.10. Results for Obstacle detection Model System Using Ensemble of Multi-class SVM Classifiers

Figure 4.15: Demonstrate the Visualization of confusion matrices



In general, ABSVM-ODMS improves the detection rate in comparison with SVM-ODMS. ABSVM-ODMS is lower than other techniques for the recognition rate of SURF. The ABSVM-ODMS method however improves the recognition of the other three techniques of extraction (GIST, HOG, and SIFT). ABSVM-ODMS is the best detection rate achieved with 97,34 percent, with HOG features in the 33x9 block configuration. The disadvantage of the ABSVM-ODMS is that computational time is increased. The speed of processing is reduced by 10 percent for GIST features, with processing speed decreased by approximately 30% in the case of HOG features. Due to the fact that each image can be processed independently, the extraction time can be reduced using GPU or CPU multiple-core devices, etc.

4.11. Misperception Matrix besides Class Recognition Percentage

Thus, the confusion matrix stretches info happening respectively vehicle class besides how well it is anticipated. The confusion matrix helps visualize the capacity of the ODMS methods to discriminate. Figure 4-15, paragraphs (a), and (b), show the confusion matrix's, respectively, for ABSVM-ODMS with GIST and HOG functionality, in Figure 4-15 and Figure 4-15 (c), and (d), To create these confusion matrices, for Random Forest and Vector Machine Support, we have chosen the ODMS configuration with the highest recognition rates for HOG and GIST features. RF- ODMS has 300 decision-making bodies for GIST and 350 decision-making bodies for HOG; HOG is used with the 33x6-block configuration. The values of C = 10 for both GIST and GIST are created for ABSVM-ODMS. The value is HOG, with 33x9 blocks configuration. The vertical axis of the confusion matrix is the actual class and horizontal axis is the ODMS system class. The diagonal elements in the confusion matrix show correct vehicle identification and incorrect identification with a non-diagonal basics. An coarser crosswise component displays improved ratings, although a blue non-diagonal component demonstrations fewer improper assessments. The outcomes of RF-ODMS finding remain non completely slanting cantered, representative the incorrect organization. The consequences as of ABSVM-ODMS remain additional slanting through actual limited non-diagonal essentials

representative the advanced recognition frequency. Thus, the class acknowledgement rates remain quantified Since the bench you can see that less trained images are available in vehicles classes at lower recognition rates. However, fewer pictures do not always influence the recognition rate, For instance, with the 22 training pictures, HONDA CITY has a lower detection rate while SUZUKI IGNIS has good detection with the 21 training pictures. For both learning algorithms, the recognition rate is not similar (Random Forest and Support Vector Machine). Likewise, the feature representation techniques do not follow a pattern (GIST and HOG).

Table 4.10: Demonstrates the Class-wise recognition rate on the testing dataset for Obstacle detection model system

Sr. No	OBSTACLE AS CAR	MODEL	TRAINING IMAGES	TESTING IMAGES	RF GIST	RF HOG	ABSVM GIST	ABSVM HOG
1	SUZUKI	BALENO	260	504	98.21%	95.83 %	98.41%	97.42%
2	SUZUKI	SWIFT	224	258	94.19%	94.19 %	87.21%	95.74%
3.	SUZUKI	DZIRE	117	169	94.08%	95.27 %	96.45%	95.86%
4.	SUZUKI	VITARA BREZZA	79	243	94.24%	94.24 %	96.30%	95.47%
5.	SUZUKI	ALTO	34	35	100.00 %	100.00 %	97.14%	100.00%
6.	SUZUKI	WAGON	19	13	84.62%	76.92 %	100.00%	100.00%
7.	SUZUKI	CELERIO	109	122	90.16%	91.80 %	99.18%	98.36%
8.	SUZUKI	IGNIS	21	5	100.00 %	100.00 %	100.00%	100.00%
9.	FORD EXCAPE	EXCAPE	96	92	96.74%	94.57 %	98.91%	97.83%
10.	FORD LIATA	LIATA	34	84	73.81%	98.81 %	96.43%	98.81%
11.	FORD MONDEO	MONDEO	242	292	96.58%	98.63 %	98.29%	98.97%
12.	FORD TIERRA	TIERRA	119	128	96.09%	96.88 %	99.22%	97.66%
13.	HONDA	CRV	66	29	89.66%	96.55 %	93.10%	93.19%

14.	HONDA	CITY	22	20	85.00%	80.00 %	95.00%	75.00%
15.	HONDA	JAZZ	24	15	100.00 %	80.00 %	93.33%	95.00%
16.	HONDA	CIVIC	67	22	90.91%	72.73 %	81.82%	96.28%
17.	HONDA	AMAZE	36	9	88.89%	66.67 %	100.00%	100.00%
18.	HONDA	FIT	37	89	93.26%	87.64 %	97.75%	95.35%
19.	HONDA	WR-V	108	139	98.56%	99.28 %	99.28%	99.28%
20.	HONDA CRV	CRV	216	227	97.36%	96.92 %	98.68%	97.80%
21.	HONDA FIT	FIT	12	13	84.62%	100.00 %	100.00%	100.00%
22.	HYUNDAI	SANTRO	26	50	78.00%	60.00 %	88.00%	100.00%
23.	HYUNDAI	GRAND I10	68	45	95.56%	91.11 %	97.78%	96.86%
24.	HYUNDAI	AURA	36	12	100.00 %	100.00 %	100.00%	100.00%
25.	HYUNDAI	VERNA	107	77	98.00%	94.81 %	100.00%	100.00%
26.	HYUNDAI	CRETA	16	61	83.61%	96.72 %	100.00%	97.72%
27.	MAHINDRA	E20PLUS	155	147	96.60%	93.88 %	100.00%	99.32%
28.	MAHINDRA	BOLERO	9	10	90.00%	60.00 %	100.00%	93.35%

29.	MAHINDRA	XUV300	79	1	100.00 %	100.00 %	100.00%	100.00%
30.	MAHINDRA	THAR	50	45	88.89%	91.11 %	97.78%	100.00%
31.	MAHINDRA	SCORPIO	15	29	79.31%	80.50 %	100.00%	96.75%
32.	MITS.	OUTLANDER	38	10	83.33%	93.33 %	90.00%	98.65%
33.	MITS. LANCER	LANCER	40	30	95.75%	90.00 %	100.00%	98.90%
34.	MITS. ZINGER	ZINGER	34	16	100.00 %	100.00 %	100.00%	100.00%
Total Accuracy					92.24%	90.25 %	97.06%	97.34%

4.12 Proposed Fast Multiclass Classification

We proposed method of fast multiclass classification in an attempt to increase efficiency. The dimensional curse refers to different problems related to the analysis and organization of data in very large spaces. Generally, the size increases quickly and the data becomes sparse. The size of the space increases. For any method that requires algebraic consequence, the sparseness of information remains difficult. Thus, the compulsory quantity of information repeatedly growths exponentially popular dimensionality towards yield a statistically complete then dependable consequence, which avoids normal information mining methods existence well-organized frequently important towards over alteration. The excess occurs if models trained on a small dataset can more likely view non-existent patterns that lead to high variation and very high mistakes on a test set. These are the usual signs of excess. Therefore, the model learns the

specifics besides noise since the exercise dataset toward the amount that the presentation of the model influences destructively happening a original dataset. Thus, the accuracy is reduced in some scenario in which data have less training in the previous ABSVM-HOG results Overall precautions are 97.34. The rate of recognition of HONDA CITY is lower at twenty-two pictures while the recognition rate is good at SUZUKI IGNIS with 21 pictures. That means the n It is therefore clear that the upper limit of the real risk is increased when dimensionality increases in relation to the number of trainings. As a result, the ABSVM suffers from over-fitting.

4.12.1 Proposed Divide and Conquer Support Vector Machine (DCSVM) for ODMS

We are presenting the Divide and Conquer Support Vector Machine (DCSVM) multi-class SVM classification method that utilizes the curse of dimensionality to efficiently carry out high-dimensional data classification.

Algorithm 4.12 ODMS- DCSVM training

1: procedure TRAINDCSVM(R1, . . . , Rk, θ). Creates DCSVM classifier

2: Input: R = R1, . . . , Rk: data set, θ: likelihood threshold

3: Output: dcsvm()

4: svmi,j ← train SVM with Ri ∪ Rj

, i = 1, . . . , k − 1, j = 2, . . . , k, i < j

5: T [svmi,j , l] = (Ci,j (l, i), Ci,j (l, j)), for all svmi,j , l = 1, . . . , k

6: //Recursively construct a binary decision tree

7: //with each node associated with a svmi,j binary classifier

8: dcsvm ← empty binary tree

9: dcsvm.root ← new tree node

10: DCSVM-SUBTREE(dcsvm.root, T , θ)

11: return dcsvm . Returns the decision tree

12: end procedure

DCSVM training and procedures are outlined by algorithms 4.12 and 4.13. TRAINDCSVM, the SVM binary classifiers for all classes of pairs are trained (line 4) and the probabilities of predictions are saved in the forecast table (line 5). The dcsvm decision function is created as an empty tree (line 8) and recurrent DCSVM-SUBTREE procedure (line 10). The recursion process produces a left and/or a right node at each step (lines 12 and 19), or can stop to make a left and / or right label (lines 9 and 16, respectively). Each new node is associated with the svmi binary or with a class label if there is an end node (line 5) (lines 9 and 16). The "optimal" svm from a current prediction table is an important part of the DCSvM-SUBTREE procedure (line 4). For that, we use the SVM Impurity Index, Balance Index and Definitions Score 9 and 10. The order in which such measures are used can affect the shape and precision of the decision tree. The resulting tree favors precision over speed of decision if Score is applied to the Impurity and Balance Indexes to break a tie (may yield bushier trees). When the indexes of impurity and balance are first used, then the value, if a tie, can be more balanced. The speed of decision is promoted while accuracy may be sacrificed. A glass data set dcsvm decision tree is displayed. Clearly, the algorithm can create highly unbalanced DCSVM decision-making trees (where some classes are decided faster) (when most of class labels are leaves situated at about same depth).

Algorithm 4.13 ODMS-DCSVM training (continued)

1: procedure DCSVM-SUBTREE(pnode, T , θ) . Creates subtree routed at pnode

2: Input: pnode: current parent node, T : current predictions table, θ: likelihood threshold

3: Output: recursively constructs sub-tree rooted at pnode

4: svmi,j ← optimal svm in T , for given θ

5: pnode[svm] ← svmi,j

6: listi ← classes labeled as i or undecided by svmi,j

7: listj ← classes labeled as j or undecided by svmi,j

8: if length(listi) = 1 then . reached a leaf

9: pnode.lef tnode ← tree-node(label in listi)

10: else

11: T .lef t ← T minus svmm,n, m ∈ listj or n ∈ listj rows, and columns of classes not in listi

12: pnode.lef t ← new tree node

13: DCSVM-SUBTREE(pnode.lef t, T .lef t, θ)

14: end if

15: if length(listj) = 1 then . reached a leaf

16: pnode.rightnode ← tree-node(label in listj)

17: else

18: T .right ← T minus svmm,n, m ∈ listi or n ∈ listi rows, and columns of classes not in listj

19: pnode.right ← new tree node

20: DCSVM-SUBTREE(pnode.right, T .right, θ)

21: end if

22: end procedure

Suggestion 1: The decision tree built in algorithm 2 has a maximum depth k. Evidence. There is at least one label each in class lists of classes I and j(lines 6, 7, DCSVM-SUBTREE process): j. After a class column is removed from T at some n node of the tree, no node or leaf of the sub-tree rooted on that n node will be reappearing. Thus the number of classes decreases from k to k to 1 with every recursion, by at least 1 (lines 11, 18) ending in lines 9 or 16 with a left or right label node. Note that it is possible in practise for a scenario in which each decision tree label has depth k: if no svmi, j binary classifier is a good separator for classes other than I and j. (and therefore at each node only classes I and j are separated, while the other are undecided and will appear in both left and right branches). For obvious reasons, we call this the worst scenario. In practice, the opposite scenario is also possible: every svmi,j separates each class into two disjoint lists of approximately the same length. The decision tree of dcsvm in this case is also much balanced, but much smaller.

Suggestion 2. When each svmi,j produces a balanced, unjointed separation between classes, the dcsvm decision tree is constructed at a maximum level in dlog2 ke Proof. Suggestion 2. This is clearly the case where at each step in the recursion there is a node so that the left sub-tree and the other half the right sub-tree are allotted to the half of the classes. This generates a balanced binary tree with k leaves, hence a maximum dlog2 ke depth.

In order to take any dataset X and predict its label, the DCSVM classifier Algorithm 14 uses the dcsvm decision tree produced by Algorithm 13. The algorithm starts with the root node for the decision tree (line 4) and then the svm foresees the route (lines 6-12) for each node until a leaf node is reached. The DCSVM prediction of the input data item x is the label of the leaf Node (line 14).

Algorithm 14 ODMS-DCSVM classifier

```
1: procedure DCSVMCLASSIFY (dcsvm, x) . Produces DCSVM classification
2: Input: dcsvm: decision tree; x: data item
3: Output: Label of data item x
4: node ← dcsvm.root
5: while node not a leaf do . Visits the decision tree nodes towards a leaf
6: svmi,j ← node[svm] . Retrieves the svm associated to current node
7: label ← svmi,j (x)
8: if label = i then
9: node ← node.left
10: else
11: node ← node.right
12: end if
13: end while
14: return label of node. Returns the leaf label
15: end procedure
```

Table 4.11 Class-wise recognition rate on the testing dataset of ODMS-ABSVM and ODMS- DCSVM

Sr. No	OBSTACLE AS CAR	MODEL	TRAINING IMAGES	TESTING IMAGES	ABSVM HOG	DC-SVM
1	SUZUKI	BALENO	260	504	97.42%	97.42%
2	SUZUKI	SWIFT	224	258	95.74%	95.74%
3.	SUZUKI	DZIRE	117	169	95.86%	95.86%
4.	SUZUKI	VITARA BREZZA	79	243	95.47%	95.47%
5.	SUZUKI	ALTO	34	35	100.00%	100.00%
6.	SUZUKI	WAGON	19	13	100.00%	100.00%
7.	SUZUKI	CELERIO	109	122	98.36%	98.36%
8.	SUZUKI	IGNIS	21	5	100.00%	100.00%
9.	FORD EXCAPE	EXCAPE	96	92	97.83%	97.83%
10.	FORD LIATA	LIATA	34	84	98.81%	98.81%
11.	FORD MONDEO	MONDEO	242	292	98.97%	98.97%
12.	FORD TIERRA	TIERRA	119	128	97.66%	97.66%
13.	HONDA	CRV	66	29	93.19%	93.19%

14.	HONDA	CITY	22	20	75.00%	95.00%
15.	HONDA	JAZZ	24	15	95.00%	97.00%
16.	HONDA	CIVIC	67	22	96.28%	96.28%
17.	HONDA	AMAZE	36	9	100.00%	100.00%
18.	HONDA	FIT	37	89	95.35%	98.89%
19.	HONDA	WR-V	108	139	99.28%	100.00%
20.	HONDA CRV	CRV	216	227	97.80%	97.80%
21.	HONDA FIT	FIT	12	13	100.00%	100.00%
22.	HYUNDAI	SANTRO	26	50	100.00%	100.00%
23.	HYUNDAI	GRAND I10	68	45	96.86%	96.86%
24.	HYUNDAI	AURA	36	12	100.00%	100.00%
25.	HYUNDAI	VERNA	107	77	100.00%	100.00%
26.	HYUNDAI	CRETA	16	61	97.72%	97.72%
27.	MAHINDRA	E20PLUS	155	147	99.32%	99.32%
28.	MAHINDRA	BOLERO	9	10	93.35%	93.35%
29.	MAHINDRA	XUV300	79	1	100.00%	100.00%
30.	MAHINDRA	THAR	50	45	100.00%	100.00%
31.	MAHINDRA	SCORPIO	15	29	96.75%	96.75%

32.	MITS.	OUTLANDER	38	10	98.65%	98.65%
33.	MITS. LANCER	LANCER	40	30	98.90%	98.90%
34.	MITS. ZINGER	ZINGER	34	16	100.00%	100.00%
				Total Accuracy	97.34%	98.11%

Thus, the above DC SVM Accuracy and Speed for Prediction of class of vehicle provides more accuracies than ABSVM under when the training data is less as compared testing data. ABSVM has produced result for ABSVM-HOG Entire precisions remains **97.34%**. HONDA CITY consumes subordinate acknowledgement rate through twenty-two exercise images though SUZUKI IGNIS consumes decent recognition frequency through twenty-one exercise imageries. When ODMS-DCSVM used the Honda city with less training data with testing data still it generates the good recognition rate.

Chapter 5 Finding of Heterogenous Model, Conclusion and Future Work

This thesis proposes and investigates unexplored approaches for highly vigorous and real-time automated obstacle detect model system (ODMS) framework. We have made various representations of the dataset dependent on image feature extraction strategies. The Different classification techniques are utilized in this work as Accuracy and Performance are considered.

5.1 Finding of Heterogenous Model

Depict the general appearance of the front or back face for every obstacle as a car model utilizing the built dictionary. The word reference can be considered as a minimal portrayal of the key features from training images, of all classes. Similarly, as a text document involves textual dictionary words, a picture can be respected to be included codewords from a visual word reference. The obstacle as car' pictures are addressed as BoF features, which are histograms of events of the dictionary codewords. Building the word reference is normally done offline and just when required, with the goal that it very well might be utilized in the training and testing stages. The Single- and Modular-Dictionary schemes proposed and investigated in this work are described in this section. The dataset is divided into a testing dataset and training dataset and vehicles are divided into 7,232 classes in this dataset based on obstacle detection model system. The training dataset is labeled dataset is named with real vehicles' data information. The testing dataset additionally has model data yet it is simply used to contrast the ODMS framework prediction with real model data. In this way, the 80% Training data and 20 % Testing data the proposed dataset is prepared. A dataset is a collection of related observations. In machine learning, the dataset is a collection of related observation which is separated into training and testing datasets. Preparing Training dataset is utilized to train the machine learning algorithm while the testing dataset is utilized to test and assess the machine learning based solution. Datasets are utilized to exhibit the adequacy of the proposed arrangement. Obstacle as vehicle recognition dataset comprises of vehicle as car images, different image features are adopted to represent each image in the dataset. Dataset are likewise prepared with some unknown model of

the car during the testing time it recognize then it gets added to the training dataset with name unknown1 then at the time of monthly audit review that unknown obstacle is feed with appropriate model data in training preparing framework. So, the Proposed Heterogenous datasets are updated timely with new data as per need. Below the figures demonstrates the training of dataset.

Figure 5.1: Demonstrate the Review of Data



```
In [3]: anno_train = pd.read_csv("D:/PHD/DATASET/car_devkit/devkit/anno_train.csv")
        anno_train.sample(5)
        # anno_train.describe()
```

Out[3]:

	00001.jpg	39	116	569	375	14
1159	01161.jpg	64	106	399	307	82
7114	07116.jpg	81	189	769	427	57
4417	04419.jpg	14	59	408	290	117
6585	06587.jpg	25	53	238	167	53
3536	03538.jpg	4	98	496	322	36

Figure 5.2: Demonstrate the Review of Data



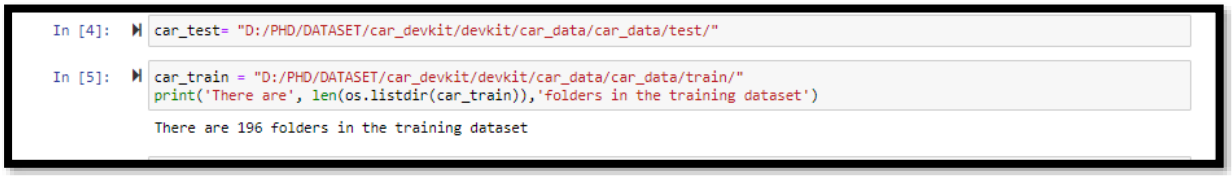
DATA REVIEW

```
In [2]: names = pd.read_csv("D:/PHD/DATASET/car_devkit/devkit/names.csv")
        names.sample(5)
```

Out[2]:

	AM General Hummer SUV 2000
133	Hyundai Elantra Sedan 2007
184	Toyota Sequoia SUV 2012
180	Suzuki Kizashi Sedan 2012
21	Audi S4 Sedan 2012
82	Dodge Caliber Wagon 2007

Figure 5.3: Demonstrate the training dataset is labeled with actual vehicles' information.

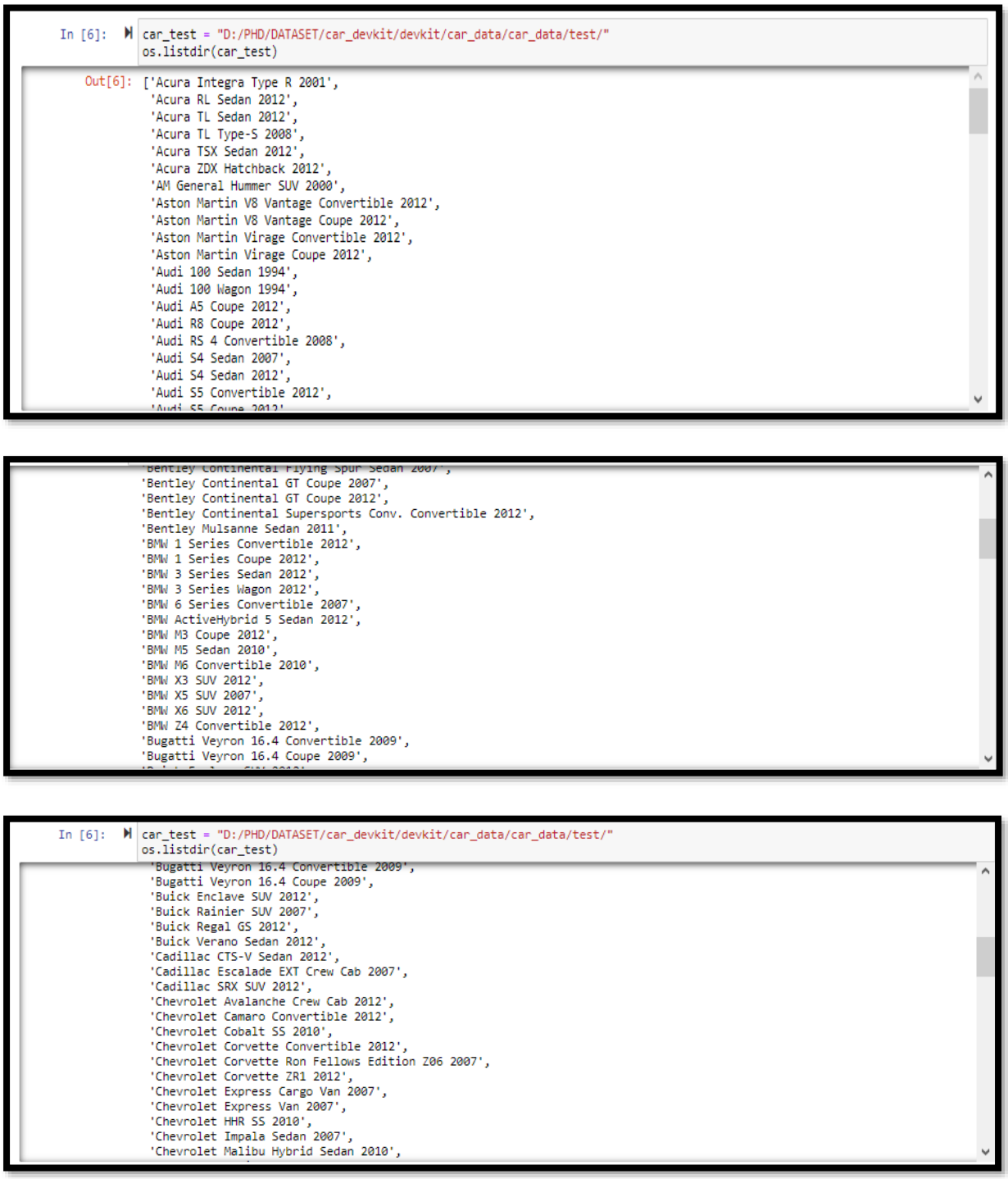


```
In [4]: car_test= "D:/PHD/DATASET/car_devkit/devkit/car_data/car_data/test/"

In [5]: car_train = "D:/PHD/DATASET/car_devkit/devkit/car_data/car_data/train/"
        print('There are', len(os.listdir(car_train)),'folders in the training dataset')

        There are 196 folders in the training dataset
```

Figure 5.4: Demonstrate the testing dataset also has model information but it is only used to compare the ODMS system prediction to actual model.



```
In [6]: car_test = "D:/PHD/DATASET/car_devkit/devkit/car_data/car_data/test/"
        os.listdir(car_test)

Out[6]: ['Acura Integra Type R 2001',
         'Acura RL Sedan 2012',
         'Acura TL Sedan 2012',
         'Acura TL Type-S 2008',
         'Acura TSX Sedan 2012',
         'Acura ZDX Hatchback 2012',
         'AM General Hummer SUV 2000',
         'Aston Martin V8 Vantage Convertible 2012',
         'Aston Martin V8 Vantage Coupe 2012',
         'Aston Martin Virage Convertible 2012',
         'Aston Martin Virage Coupe 2012',
         'Audi 100 Sedan 1994',
         'Audi 100 Wagon 1994',
         'Audi A5 Coupe 2012',
         'Audi R8 Coupe 2012',
         'Audi RS 4 Convertible 2008',
         'Audi S4 Sedan 2007',
         'Audi S4 Sedan 2012',
         'Audi S5 Convertible 2012',
         'Audi S5 Coupe 2012'
```



```
'Bentley Continental Flying Spur Sedan 2007',
'Bentley Continental GT Coupe 2007',
'Bentley Continental GT Coupe 2012',
'Bentley Continental Supersports Conv. Convertible 2012',
'Bentley Mulsanne Sedan 2011',
'BMW 1 Series Convertible 2012',
'BMW 1 Series Coupe 2012',
'BMW 3 Series Sedan 2012',
'BMW 3 Series Wagon 2012',
'BMW 6 Series Convertible 2007',
'BMW ActiveHybrid 5 Sedan 2012',
'BMW M3 Coupe 2012',
'BMW M5 Sedan 2010',
'BMW M6 Convertible 2010',
'BMW X3 SUV 2012',
'BMW X5 SUV 2007',
'BMW X6 SUV 2012',
'BMW Z4 Convertible 2012',
'Bugatti Veyron 16.4 Convertible 2009',
'Bugatti Veyron 16.4 Coupe 2009',
```



```
In [6]: car_test = "D:/PHD/DATASET/car_devkit/devkit/car_data/car_data/test/"
        os.listdir(car_test)

        'Bugatti Veyron 16.4 Convertible 2009',
        'Bugatti Veyron 16.4 Coupe 2009',
        'Buick Enclave SUV 2012',
        'Buick Rainier SUV 2007',
        'Buick Regal GS 2012',
        'Buick Verano Sedan 2012',
        'Cadillac CTS-V Sedan 2012',
        'Cadillac Escalade EXT Crew Cab 2007',
        'Cadillac SRX SUV 2012',
        'Chevrolet Avalanche Crew Cab 2012',
        'Chevrolet Camaro Convertible 2012',
        'Chevrolet Cobalt SS 2010',
        'Chevrolet Corvette Convertible 2012',
        'Chevrolet Corvette Ron Fellows Edition Z06 2007',
        'Chevrolet Corvette ZR1 2012',
        'Chevrolet Express Cargo Van 2007',
        'Chevrolet Express Van 2007',
        'Chevrolet HHR SS 2010',
        'Chevrolet Impala Sedan 2007',
        'Chevrolet Malibu Hybrid Sedan 2010',
```

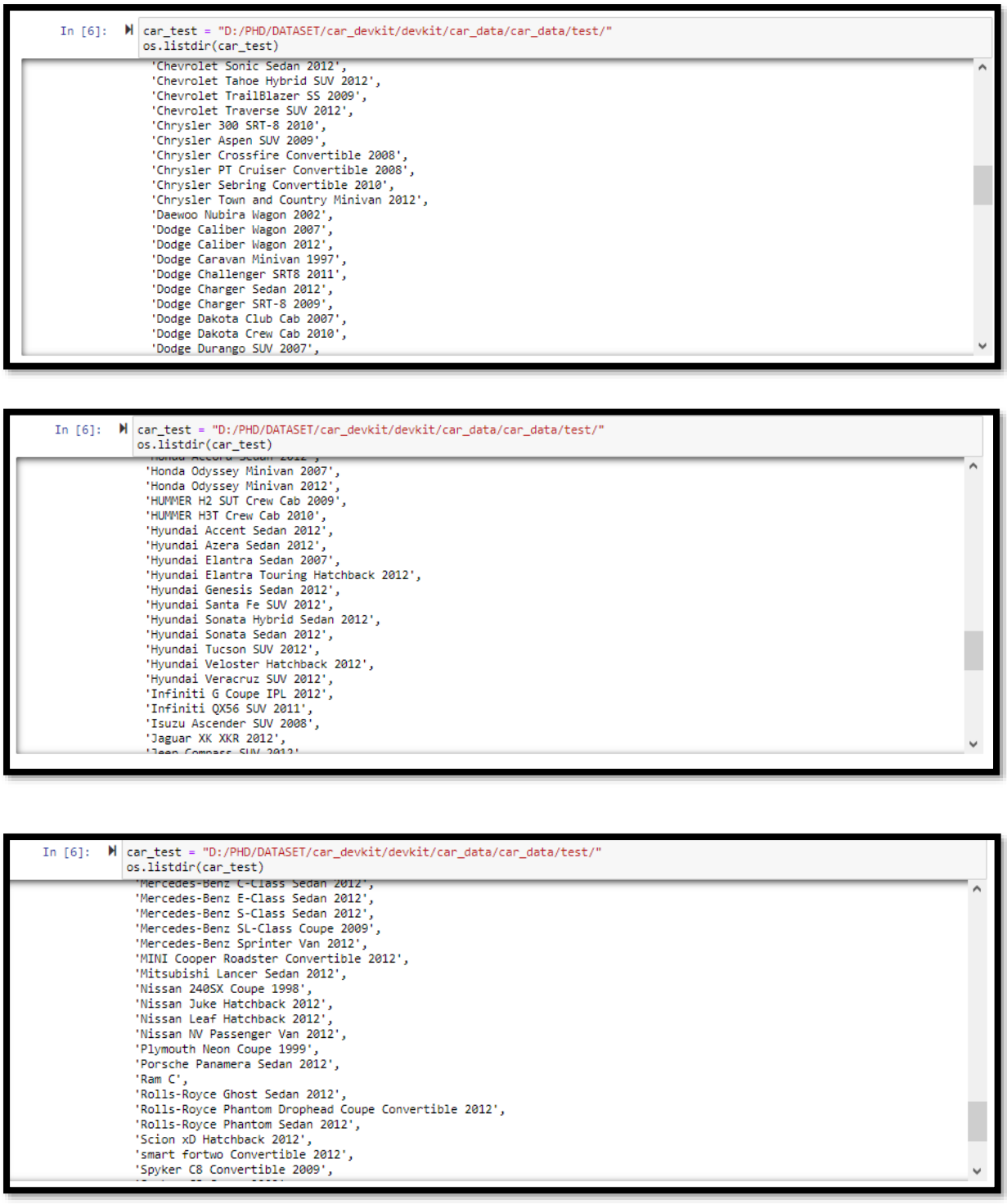
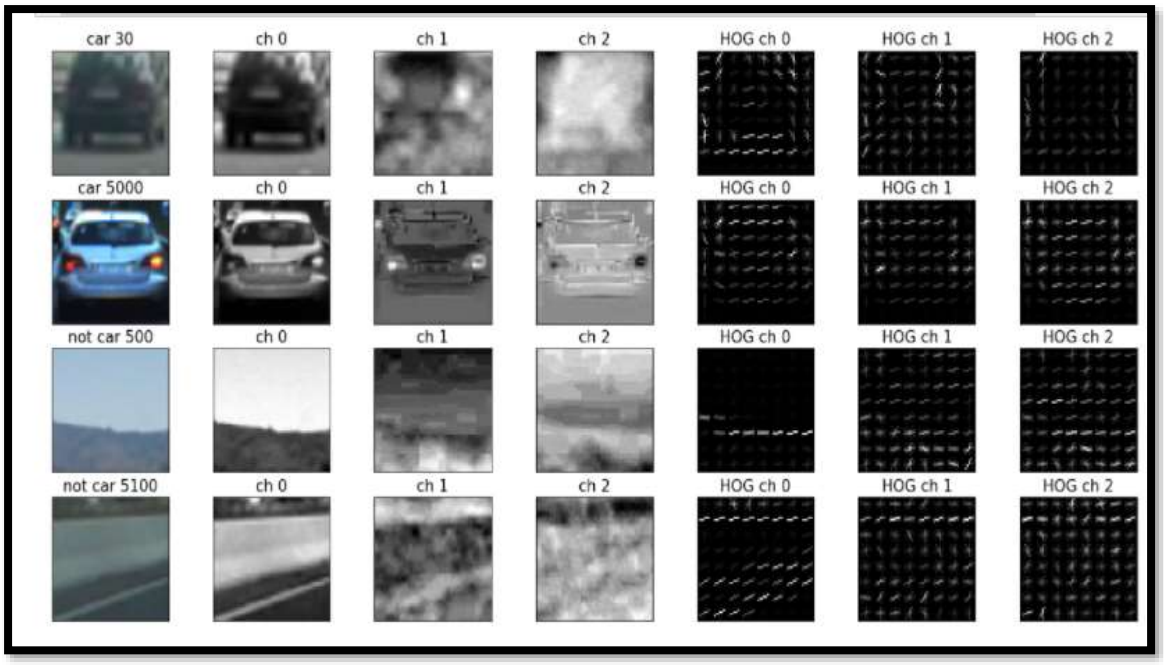
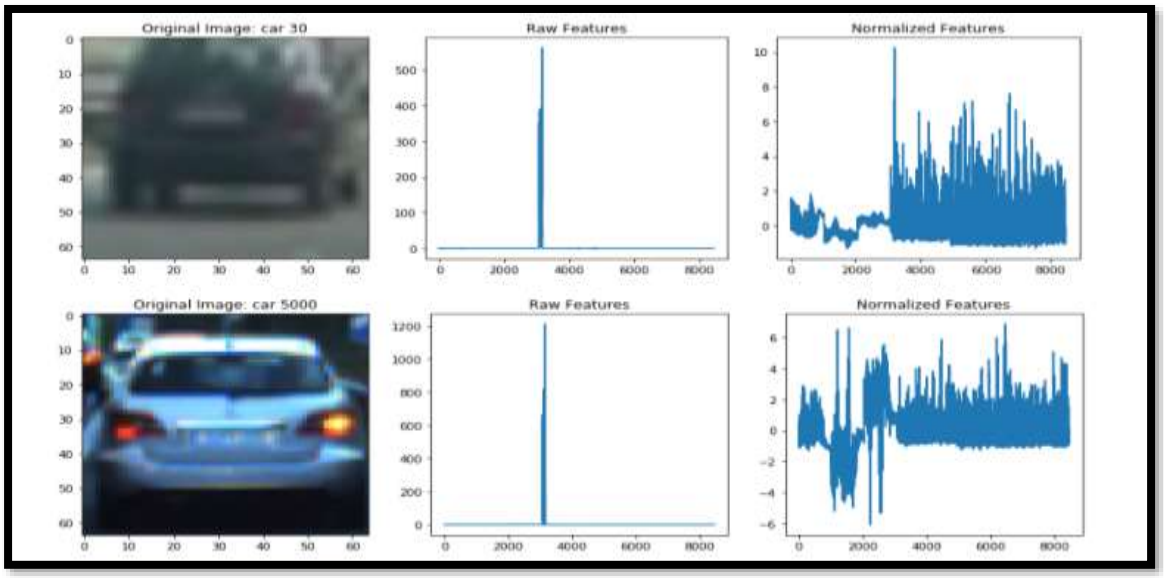


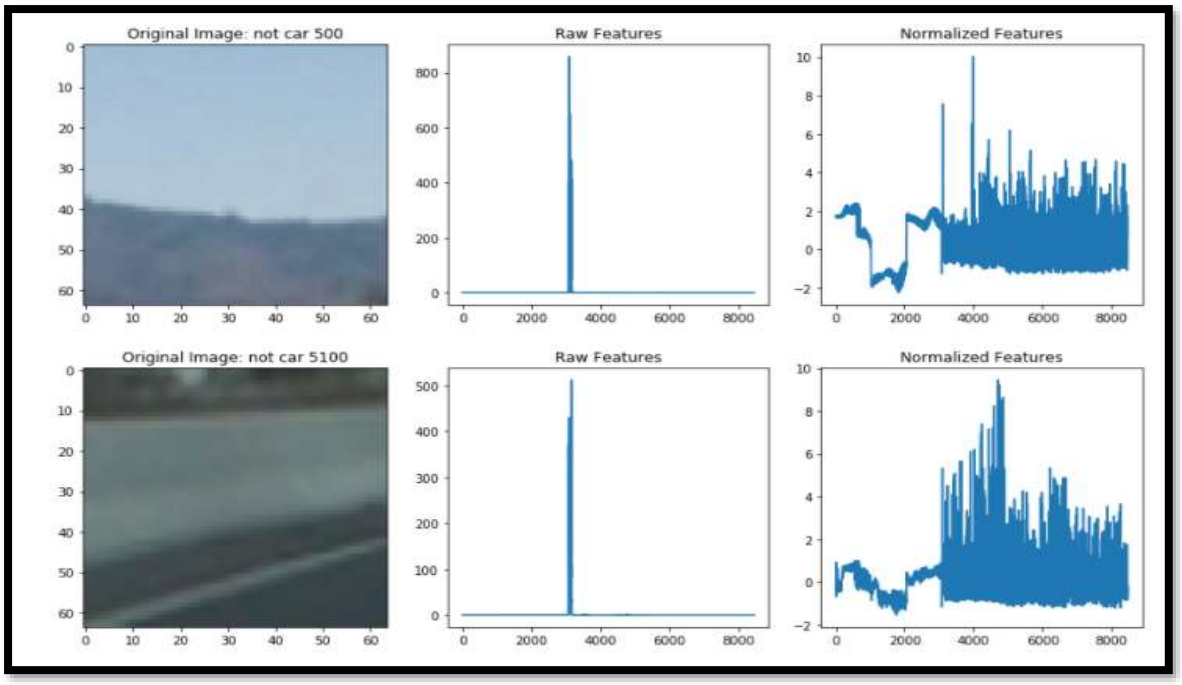
Figure 5.5: Demonstrates the Samples from Cars and Notcars sets


car 30
ch 0
ch 1
ch 2
HOG ch 0
HOG ch 1
HOG ch 2
car 5000
ch 0
ch 1
ch 2
HOG ch 0
HOG ch 1
HOG ch 2
not car 500
ch 0
ch 1
ch 2
HOG ch 0
HOG ch 1
HOG ch 2
not car 5100
ch 0
ch 1
ch 2
HOG ch 0
HOG ch 1
HOG ch 2

(a)


Original Image: car 30
Raw Features
Normalized Features
Original Image: car 5000
Raw Features
Normalized Features

(b)



(c)

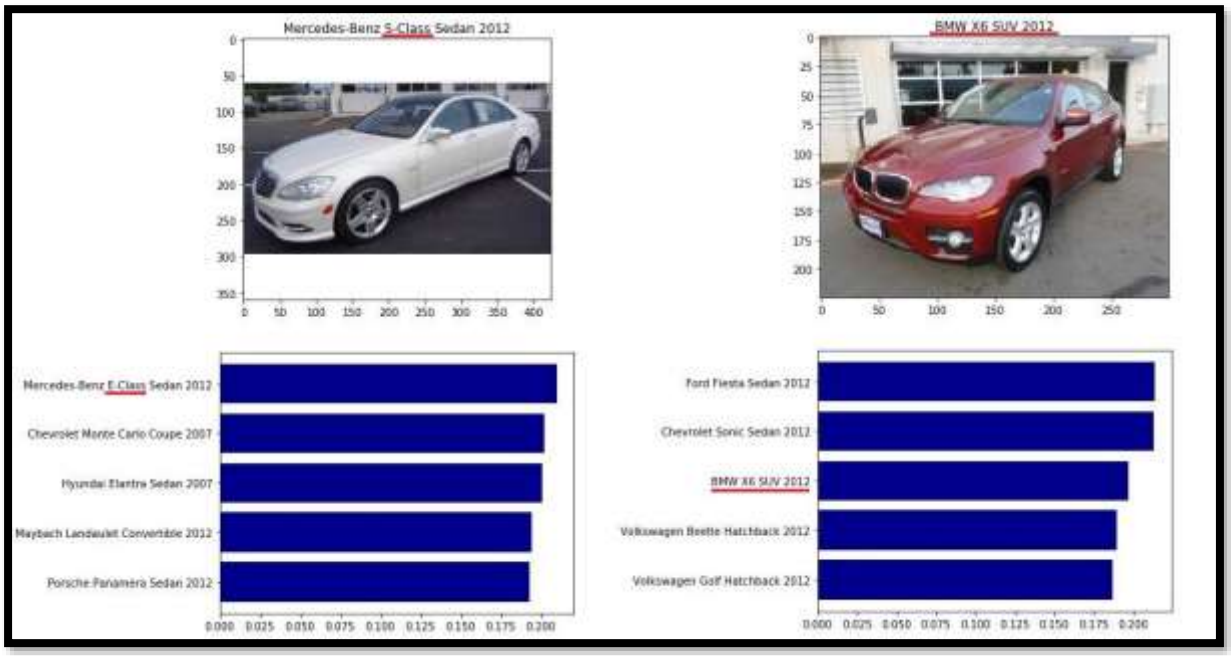
Figure 5.6: Demonstrates the Train SVM Classifier

```
Feature vector length: 8460
25.32 Seconds to train SVC...
Test Accuracy of SVC =  0.993
My SVC predicts:  [ 1.  1.  0.  1.  1.  0.  1.  0.  1.  1.  0.  1.  0.  1.  1.  0.  0.  0.
  1.  1.  1.  0.  1.  0.  0.  1.  1.  1.  1.  0.  0.  1.  0.  0.  0.  0.
  0.  1.  1.  1.  0.  0.  1.  1.  1.  0.  0.  1.  0.  0.  1.  0.  1.  1.
  0.  0.  0.  1.  0.  0.  1.  1.  1.  0.  1.  1.  0.  1.  0.  0.  1.  1.
  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  1.  0.
  1.  0.  1.  0.  0.  1.  1.  1.  0.  1.]
For these 100 labels:  [ 1.  1.  0.  1.  1.  0.  1.  0.  1.  1.  0.  1.  0.  1.  1.  0.  0.  0.
  1.  1.  1.  0.  1.  0.  0.  1.  1.  1.  1.  0.  0.  1.  0.  0.  0.  0.
  0.  1.  1.  1.  0.  0.  1.  1.  1.  0.  0.  1.  0.  0.  1.  0.  1.  1.
  0.  0.  0.  1.  0.  0.  1.  1.  1.  0.  1.  1.  0.  1.  0.  0.  1.  1.
  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  1.  0.
  1.  0.  1.  0.  0.  1.  1.  1.  0.  1.]
0.16893 Seconds to predict 100 labels with SVC
```

Figure 5.7: Demonstrates the Test prediction

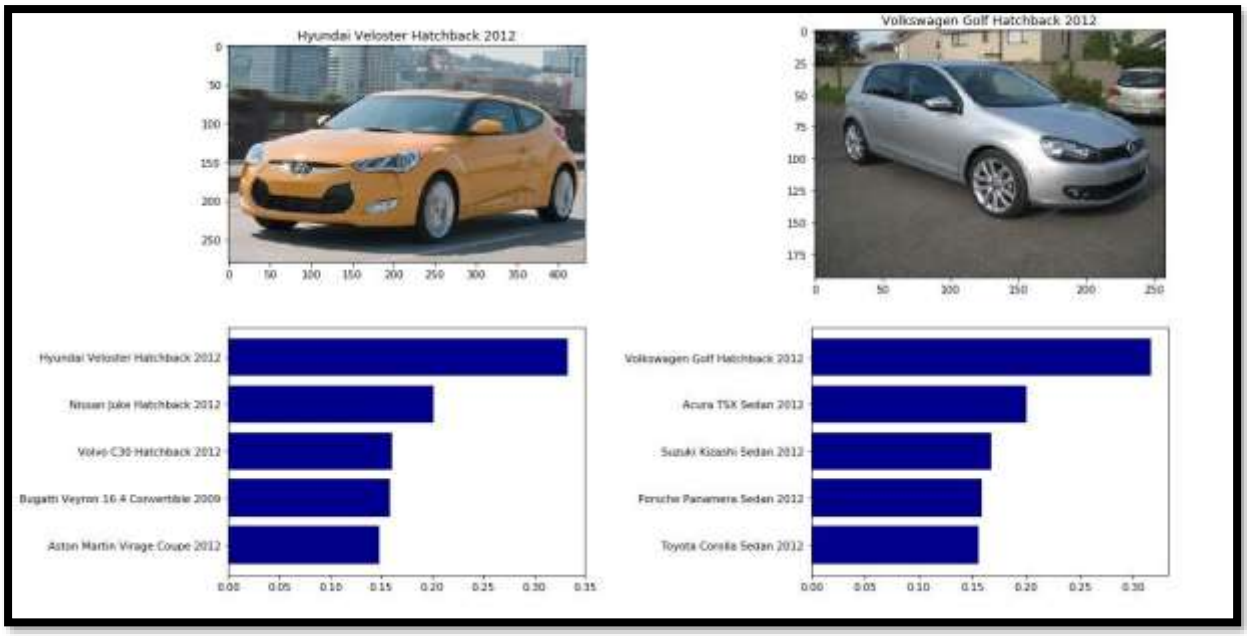


Figure 5.8 Demonstrates the Test prediction of car model



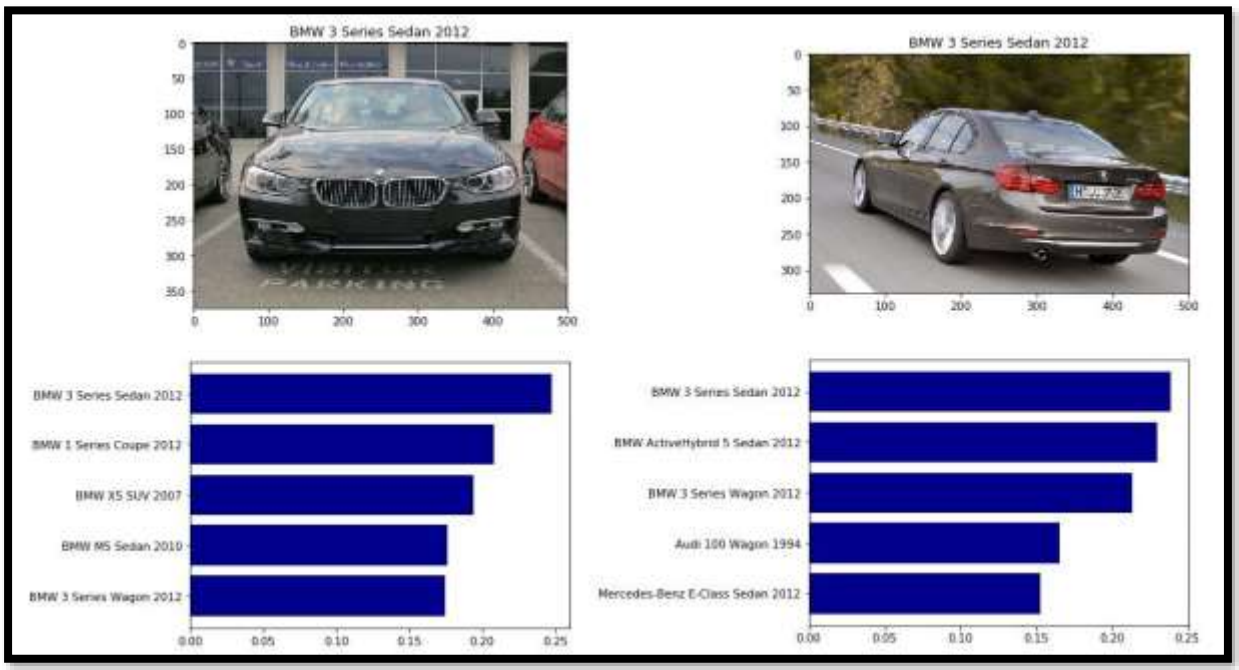
(a)

For the Mercedes-Benz S-Class Sedan 2012 the model anticipated Mercedes-Benz E-Class Sedan 2012 which is off base. The Mercedes-Benz S-Class Sedan 2012 likewise didn't show up in the best 5 anticipated classes. BMW X6 SUV 2012 anticipated semi-accurately in light of the fact that it showed up in third top classes. It assumed be the top classes in the expectation result not the Ford Fiesta Sedan 2012.



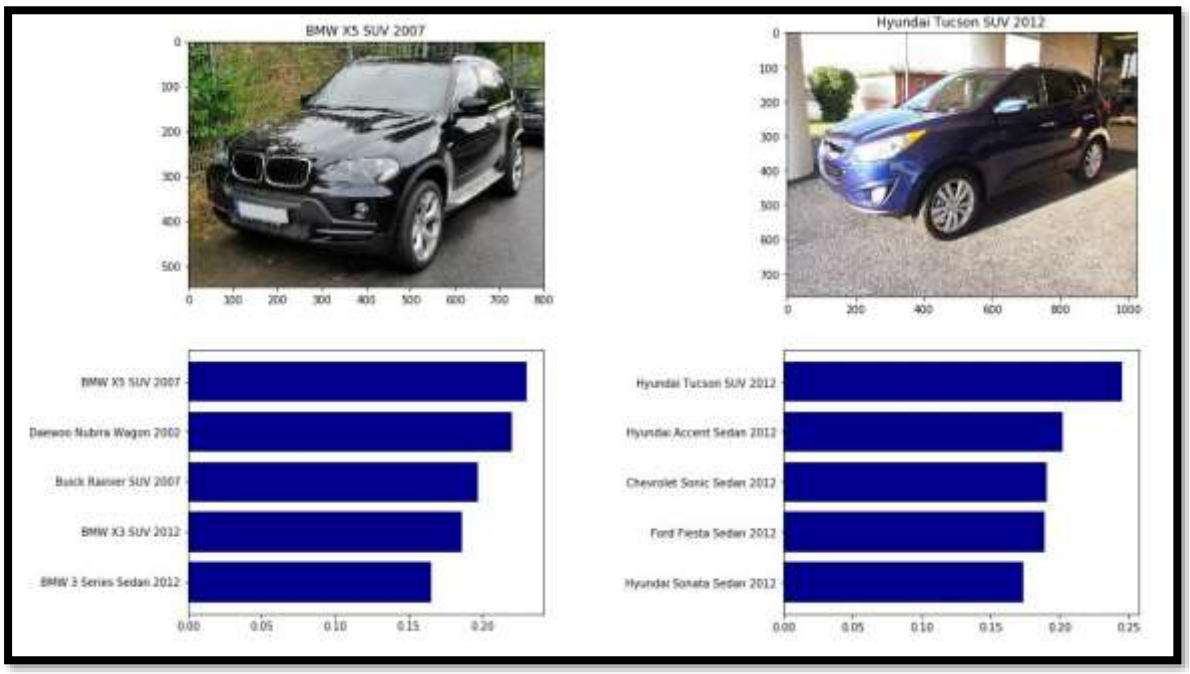
(b)

Hyundai Veloster Hatchback 2012 and Volkswagen Golf Hatchback 2012 were very much anticipated with over 30%.



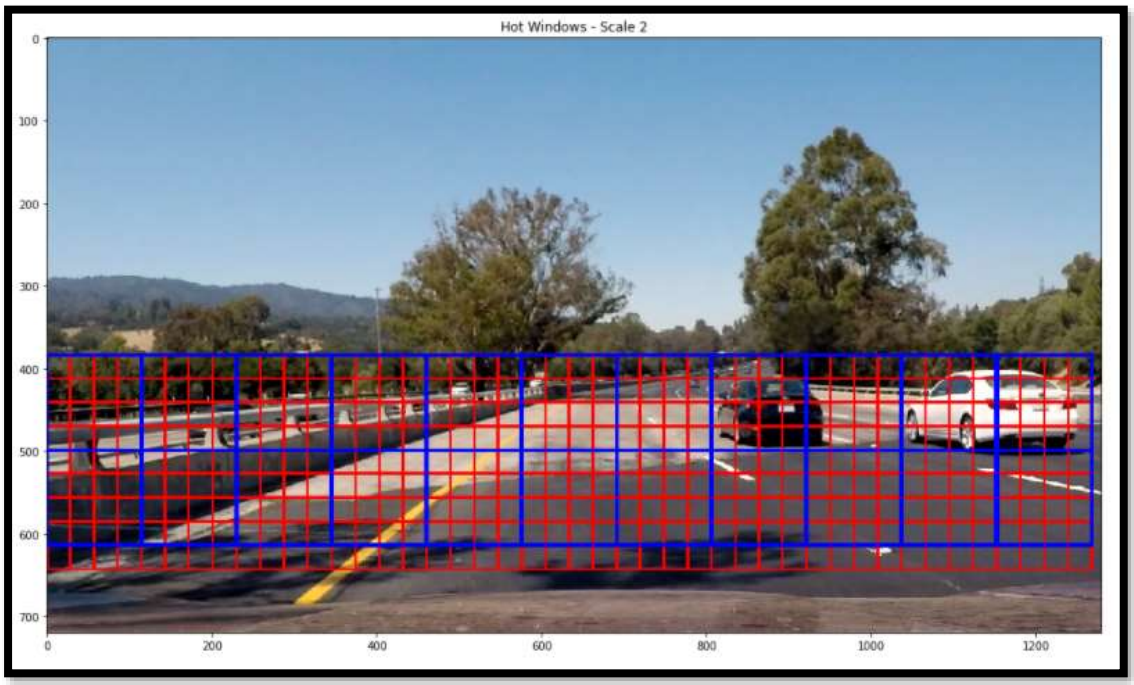
(c)

However, in this case BMW M5 Sedan 2010 does not performing, it got all top 5 predictions classes wrong. Any BMW car model did not show up in the result.



(d)

Figure 5.9: Demonstrates Multiscale grid for obstacle detection



(a)



(b)

Obstacles are set apart as pressing or normal based on their distance to the subject and the related movement vector orientation. Following, the detected obstacles are taken care an object classifier. We consolidate HOG descriptor into the Bag of Feature (BoF) recovery structure and show how this mix might be utilized for obstacle classification in video transfers.

Figure 5.10: Demonstrates Obstacle model output from partial view images of car Predict the models



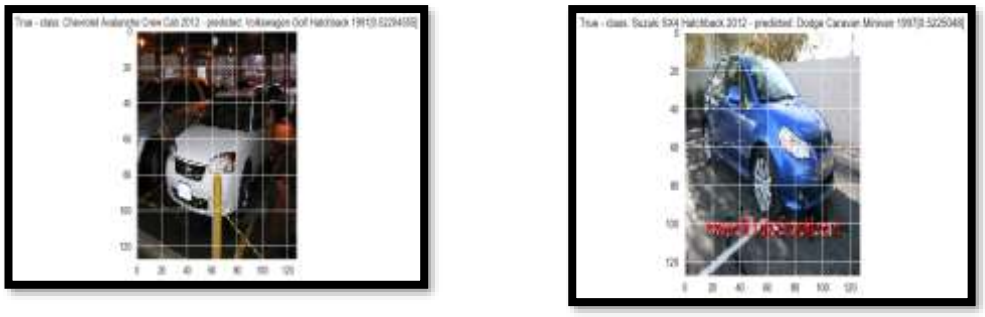
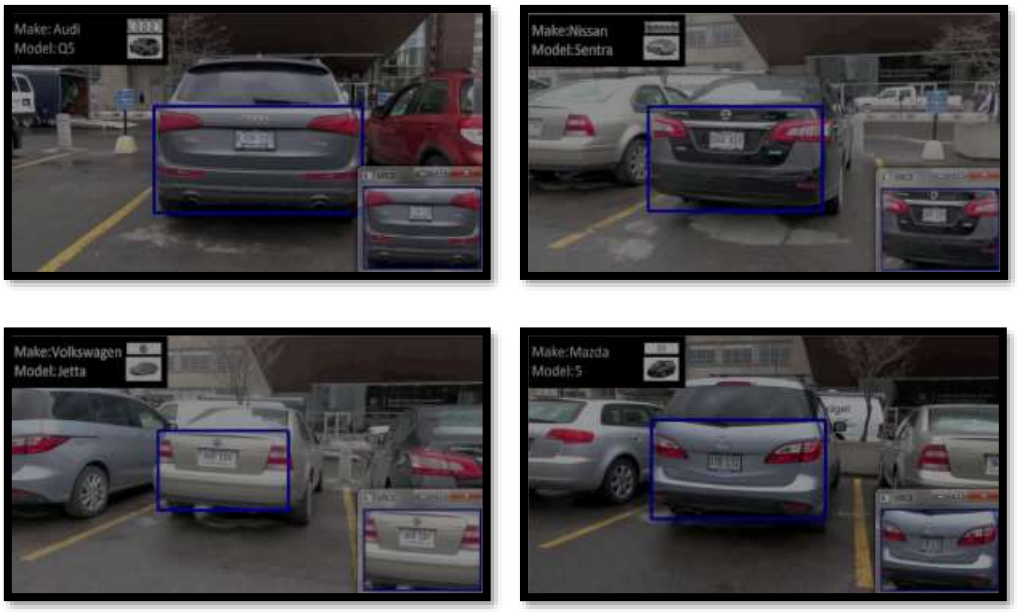


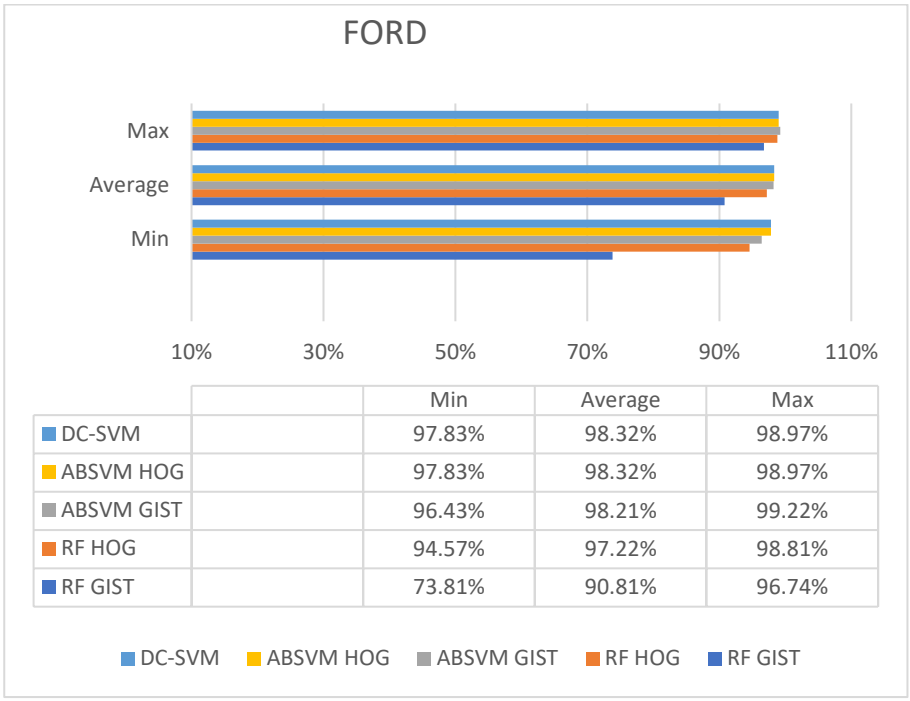
Figure 5.11: Demonstrates Real time testing using ODMS-DCSVM

Each of the 12 of the impediments as obstacles car make model were perceived effectively.



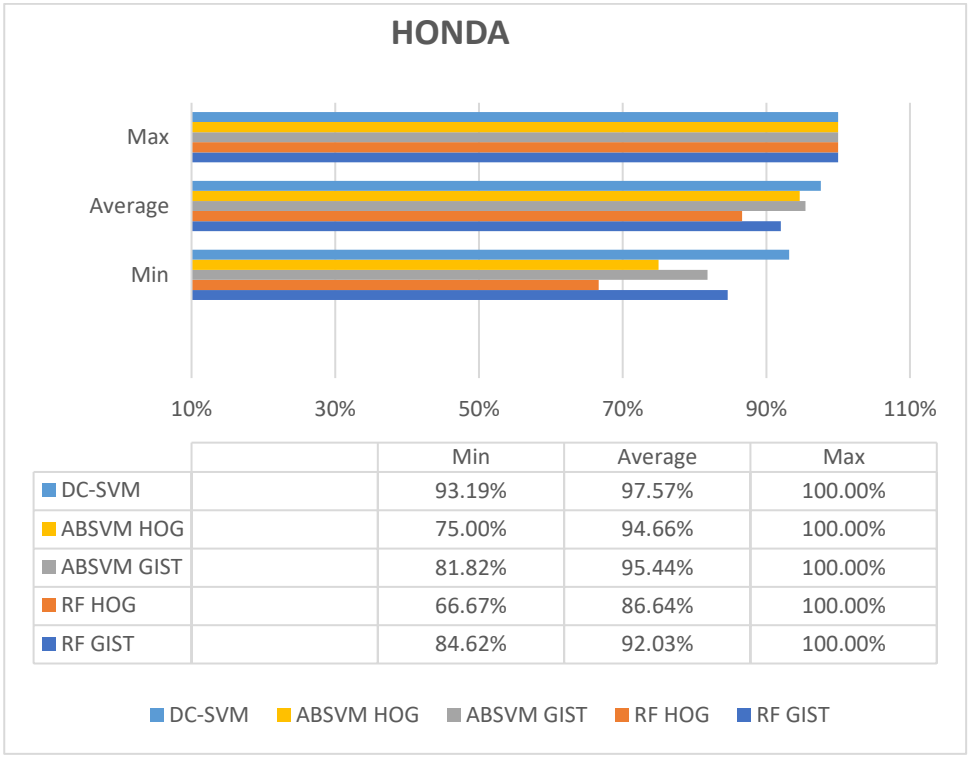
DCSVM-ODMS approaches were fruitful in foreseeing the model of car as obstacle in the abovementioned cases. In this manner, the DCSVM-ODMS give Accuracy and Speed to Prediction of class of vehicle gives a larger number of correctness's than ABSVM under when the preparation information is less as looked at testing information. Each of the 6 of the impediments as vehicle make model were perceived effectively.

Figure 5.12 Demonstrates the minimum, maximum and average recon ignition rate for the Ford different models.



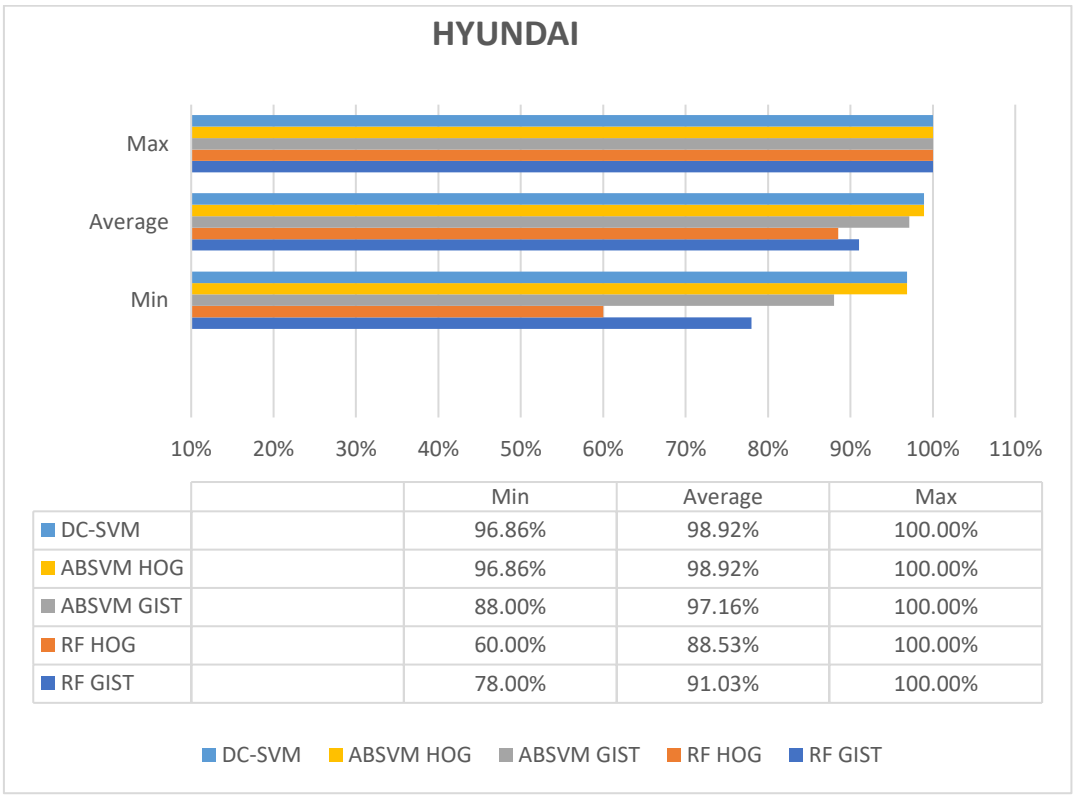
		Min	Average	Max
DC-SVM		97.83%	98.32%	98.97%
ABSVM HOG		97.83%	98.32%	98.97%
ABSVM GIST		96.43%	98.21%	99.22%
RF HOG		94.57%	97.22%	98.81%
RF GIST		73.81%	90.81%	96.74%

Figure 5.13 Demonstrates the minimum, maximum and average reconignition rate for the Honda different models.



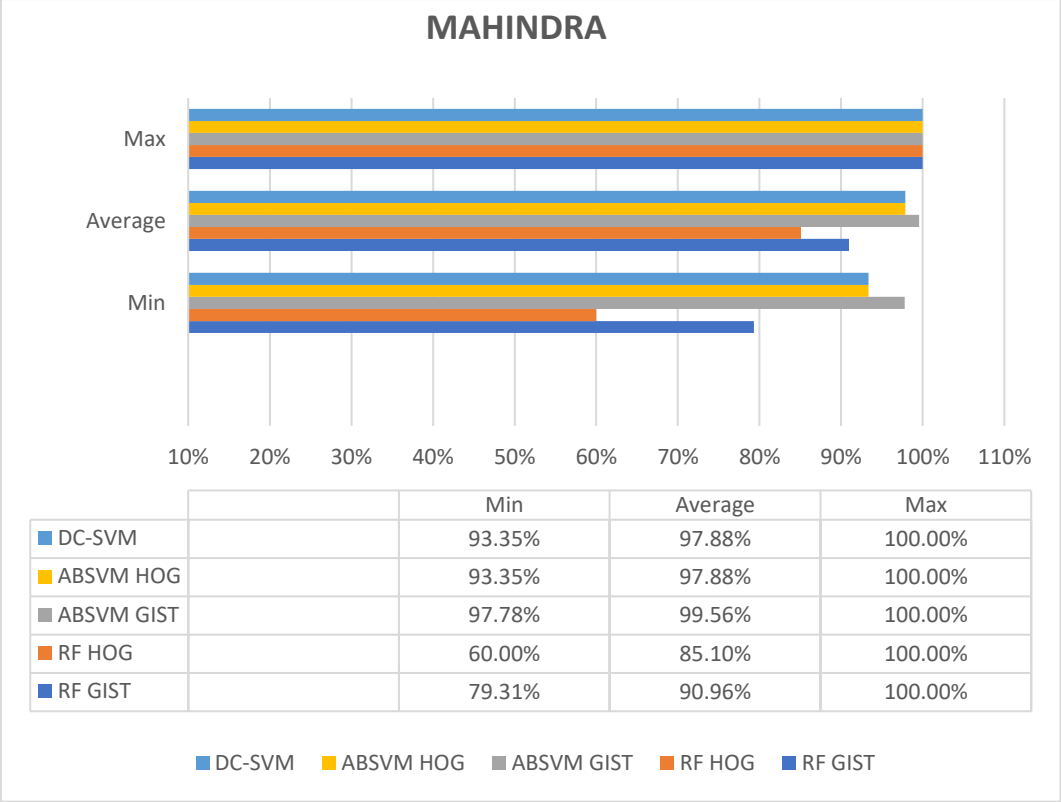
		Min	Average	Max
DC-SVM		93.19%	97.57%	100.00%
ABSVM HOG		75.00%	94.66%	100.00%
ABSVM GIST		81.82%	95.44%	100.00%
RF HOG		66.67%	86.64%	100.00%
RF GIST		84.62%	92.03%	100.00%

Figure 5.14 Demonstrates the minimum, maximum and average reconignition rate for the Hyundai different models.



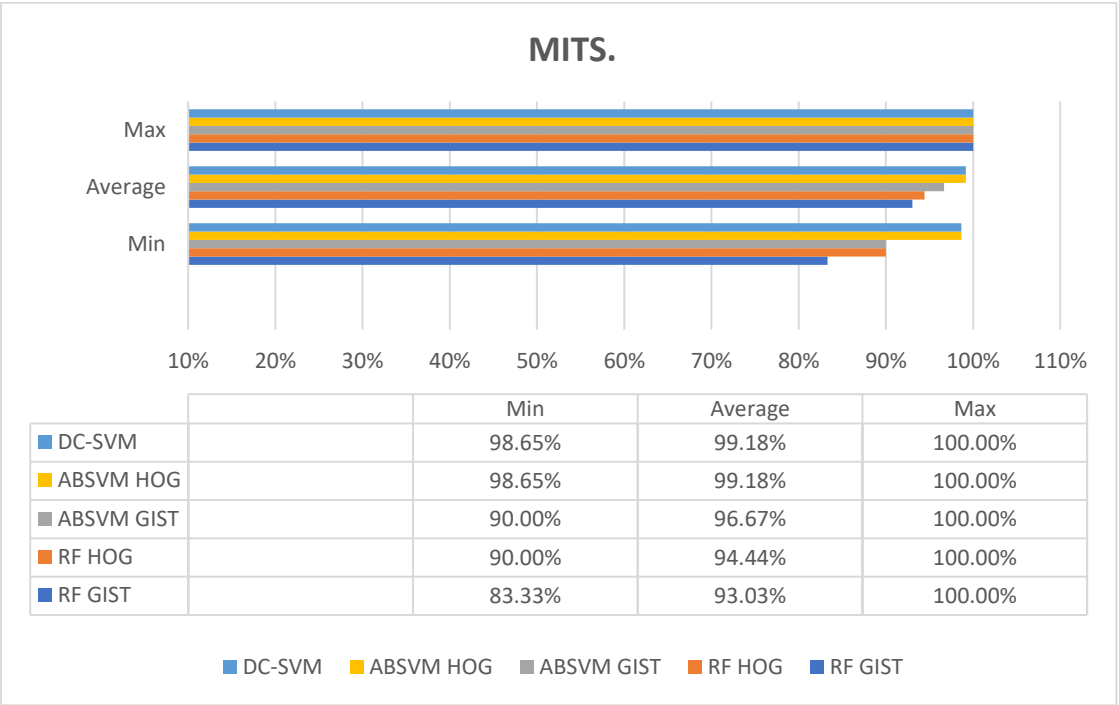
		Min	Average	Max
DC-SVM		96.86%	98.92%	100.00%
ABSVM HOG		96.86%	98.92%	100.00%
ABSVM GIST		88.00%	97.16%	100.00%
RF HOG		60.00%	88.53%	100.00%
RF GIST		78.00%	91.03%	100.00%

Figure 5.15 Demonstrates the minimum, maximum and average reconignition rate for the Mahindra different models.



	Min	Average	Max
DC-SVM	93.35%	97.88%	100.00%
ABSVM HOG	93.35%	97.88%	100.00%
ABSVM GIST	97.78%	99.56%	100.00%
RF HOG	60.00%	85.10%	100.00%
RF GIST	79.31%	90.96%	100.00%

Figure 5.16 Demonstrates the minimum, maximum and average reconignition rate for the Mits different models.



	Min	Average	Max
DC-SVM	98.65%	99.18%	100.00%
ABSVM HOG	98.65%	99.18%	100.00%
ABSVM GIST	90.00%	96.67%	100.00%
RF HOG	90.00%	94.44%	100.00%
RF GIST	83.33%	93.03%	100.00%

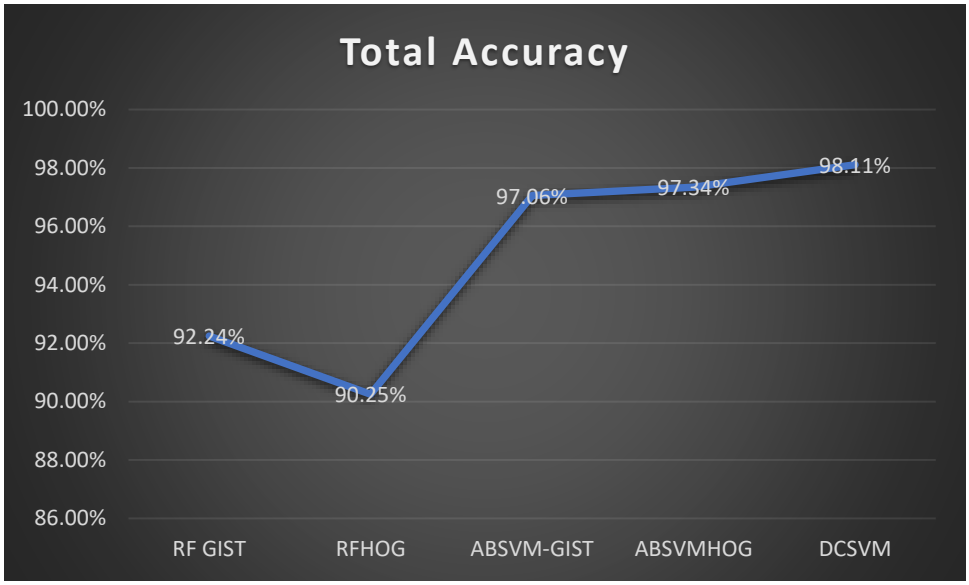
ABSVM has created result for ABSVM-HOG Total accuracies is 97.34%. HONDA CITY has lower acknowledgment rate with 22 training images while

SUZUKI IGNIS has great acknowledgment rate with 21 preparing pictures. At the point when DCSVM-ODMS utilized the Honda city with less preparing data a greater number of testing information still it creates the great acknowledgment rate. In the underneath figure show of how DCSVM give the more exactness when compared with ABSVM HOG, ABSVM GIST, RF HOG, RF GIST. Thus, the heterogenous characterization methods are utilized in obstacle detection model system utilizing for identifying the vehicle model of specific detected obstacle as car with Accuracy as boundary addressed in below table. The table show that the proposed DCSVM-ODMS give the highest accuracy when contrasted with different models and it additionally resolve the overfitting issue.

Table 5.1 Demonstrates the Total Accuracy rate of the Hybrid Classification Techniques.

Hybrid Classification Techniques	Total Accuracy
RF GIST	92.24%
RFHOG	90.25%
ABSVM-GIST	97.06%
ABSVMHOG	97.34%
DCSVMHOG	98.11%

Figure 5.17 Demonstrates the Total Accuracy rate of the Hybrid Classification Techniques



.

The proposed RF-ODMS and ABSVM-ODMS, DCSVM frameworks are contrasted and seven different methodologies in Below Table as for the acknowledgment rate and preparing speed. The wide range of various existing methodologies perceive the vehicles dependent on Model, But the Recognition Rate is High. Our ODMS frameworks best outcome than different frameworks, as far as both recognition rate and processing speed. The consequences of our proposed ODMS frameworks, are the best results among all the element extraction and machine learning algorithm calculation varieties. Our DCSVM-ODMS framework beats both of their frameworks as far as acknowledgment rate and preparing speed. Most previous approaches utilize key point-based highlights like SIFT, SURF, edges, corners, and so forth to address the pictures, which might be the explanation their less fortunate exhibition. We noticed more unfortunate execution for key point-based feature representations in our work.

Table 5.2 Comparation of Proposed models with Accuracy and Performance

Method	**Feature Extraction**	**Classification**	**Dataset**	**Recognition Rate**	**Processing Speed (fps)**
BoF-SD with SSVM-ODMS	SURF	Support Vector Machine	29 vehicles with 6 Classes	94.84	7.5
BOF-MD SSVM-ODMS	SURF	Support Vector Machine	29 vehicles with 6 Classes	93.70	6.99

RF-ODMS (HOG)	HOG	Random Forest	33 vehicles7 classes	90.25%	35.7
ABSVM-ODMS (HOG)	HOG	Support Vector Machine	33 vehicles 7 classes	97.34%	13.9
DCSVM-ODMS (HOG)	HOG	Support Vector Machine	33 vehicles 7 classes	98.11%	13.2

5.2 Limitations of the Study

There are some limitations to this study.

- High data necessity More the information for the preparation of the model, more accurate will the model be. In this way, one more constraint is the amount of dataset we use for training and testing purposes as there is a chance of false positives.
- In this framework consideration will obstacle as vehicle, in vehicle only car is considered.

5.3 Conclusion & Future work

The literature survey is done is to comprehend the past work done in the picked topic. Numerous papers were read and analysed to comprehend in which directions different Authors have moved in this field. Detailed review of vehicle as obstacle recognition and classification research utilizing image processing and machine learning / data mining techniques methods.

The mixed research approach is utilized in research study where the essential data source advising the study is quantitative in form of online surveys. Altogether, 361 responses were received frequent traveller. In data assortment process first examination for the exiting data is conveyed then the testing performed on exiting dataset whether it is adequate for our proposed model.

At last, proposed Heterogenous dataset because of variety and restrictions of existing datasets for the obstacle detection model system. Along these lines, the Proposed Heterogenous Dataset is gathered from on the web and images captures at heterogenous environments of obstacles as vehicle as car. At last, the proposed heterogenous contain

images of 141,941 and the classes is 7232. Quantitative Data utilized for the Research Study is 361, Qualitative Data utilized for the examination study is 141,941. total data utilized for the proposed model is 1,42,302.

After the study of the issue through the study the Obstacle detection model system is proposed which firstly Obstacle detection model system is to check the presence of obstacle as vehicle in the input image or video frame. In case of nonattendance of the obstacle as vehicle as car, no further preparing is required. However, the input image or video frame is processed further to recognize the vehicle in case of existence of the vehicle. Once the existence of vehicle is confirmed, next we try to remove the background and unwanted portion of the image and isolate the vehicle in the image.

The Region of Interest (ROI) to represent the part of the vehicle in the image that provides the discriminative and prominent features. The discriminative and prominent features are easily distinguishable between different vehicles. The frontal vehicle images in our work to design ODMS and used bumper, front lights, and bonnet area as ROI bounding block. The ROI extraction module removes he background as well as part of vehicle from any given image that cannot be helpful during classification and can degrade the classification performance.

The next step is to extract image features and generate a global feature representation (image feature vector) that will be used by classifier to train the system and predict the new upcoming images. Region of Interest (ROI) Extraction and feature extraction are common for both training and testing tasks. The SIFT and SURF feature extraction techniques extract multiple local features from the image. Each feature is constructed using the local neighbourhood. Next step is to form a global representation of these local images and construct image feature vector that represents the entire image. The Image feature vector represents all dataset images in the same dimensionality and is used to train the classifier. We use Bag-of-Features (BoF) to compute image feature vector for SIFT and SURF. The HOG divides the image into blocks and computes the descriptor for each block, and then, concatenates the descriptors for all blocks. The simple concatenation is a global feature representation technique used for HOG. The concatenation is a part of HOG algorithm. Hence HOG descriptor uniformly represents all dataset images with the same dimensionality. Feature extraction module computes the discriminative features from the extracted ROI. The global feature representation module may generate a model for the encoding of images' features depending on the applied technique. Global Feature Representation module use the entire features set of

an image for a uniform representation, we may or may not require explicit global feature representation technique. Region of interest extraction is often considered as a part of vehicle detection process. Two distinct schemes for optimized Dictionary Building are proposed and investigated to address the multiplicity and ambiguity, The optimal dictionary sizes for both dictionaries are Single-SVM, Attribute Bagging based Ensemble of SVMs (AB-SVM) Performance of SSVM-based BoF-ODMS.

the computation times required for SIFT and SURF are much higher than those required for HOG and GIST, The RF-ODMS recognition rates shown in Figure 4-6 for four feature extraction techniques ((a) GIST, (b) HOG, (c) SIFT, and (d) SURF, Overall, GIST gives a slightly higher recognition rate than HOG in RF-ODMS but HOG has a faster processing speed, which makes HOG.

The proposed RF-ODMS and ABSVM-ODMS systems are compared with seven other ODMS approaches, with respect to the recognition rate and processing speed. The first element is the representation of dataset (image feature vectors) which includes the feature extraction techniques and global feature representation techniques. The second element is the machine learning classification algorithm. Both feature representation and machine learning algorithm affect the ODMS system. Observing the accuracies of ABSVM-ODMS and RF-ODMS

All the other existing ODMS approaches recognize the vehicles based on Model, thus we also computed vehicle classification results based on Model but under changeling scenarios the partial images obstacle as car model detection is carried out with consideration performance and accuracy parameter. Our ODMS systems outperform other vehicle detection systems, in terms of both recognition rate and processing speed. The results of our proposed ODMS systems are the best outcomes among all the feature extraction and machine learning algorithm variations. The ABSVM-ODMS method generally improves the recognition rate compared to SSVM-ODMS. The recognition rate for ABSVM-ODMS with SURF is lower than with other techniques. However, the ABSVM-ODMS method improves the recognition rate for the other three feature extraction techniques (GIST, HOG, and SIFT). The best recognition rate achieved with the ABSVM-ODMS method is 97.34% which is achieved with HOG features with a configuration of 33x9 blocks. The drawback of the ABSVM-ODMS method is increased computational time. In the case of GIST features, the processing speed is decreased by 10% and in the case of HOG features, the processing speed is decreased by around 30%. In an attempt to increase effectiveness, we propose a Proposed Fast

Multiclass Classification Method which solved Overfitting problem when Models trained on a small dataset are more likely to see patterns that do not exist, which results in high variance and very high error on a test set. So, the accuracy get the decrease at certain scenario where the data had less training sample in the previous result of ABSVM-HOG Total accuracies is 97.34. HONDA CITY has lower recognition rate with twenty-two training images while SUZUKI IGNIS has good recognition rate with twenty-one training images. When DCSVM-ODMS used the Honda city with less training data with testing data still it generates the good recognition rate. Thus, the DCSVM produces the better result in terms of Accuracy with less trained data 98.11%.

The major contributions of this work are:

1. The proposed obstacle detection system achieves better performance than previous vehicle detection systems in terms of the recognition rate and processing speed. Jabbar et al. achieved the best processing speed of 7.4 frames per second with a recognition rate of 94.84%. With our DCSVM-ODMS system using HOG features, we achieved the recognition rate of 98.11% with the processing speed of 13.2frames per second.

2. We evaluated Two existing machine learning classification algorithms for ODMS and proposed DCSVM-ODMS method, which improved the recognition rate.

3. we have proposed the DC SVM algorithm to improve the accuracy for the proposed model the obstacles. The proposed system works well in challenging situations where obstacles are partially occluded, partial out of the image frame or poorly visible due to low lighting. An Obstacle detection model system can provide great value in terms of obstacle As Vehicle detection monitoring and identification based on obstacle. Thus, our Obstacle detection model system recognizes a obstacle and provides information about obstacle type information.

4. Since the DC-SVM-ODMS provide better accuracy than remaining techniques but processing speed is low, to Improve Processing Speed DC-SVM-ODMS could be re-implemented in a procedural language, such as C++, to increase their speed.

REFERENCES

[1] H. He, Z. Shao, and J. Tan, “Recognition of Car Makes and Models from a Single Traffic Camera Image (2015),” IEEE Transactions on Intelligent Transportation Systems, vol. PP, no. 99, pp. 1–11.

[2] G. E. Moore, "Cramming more components onto integrated circuits", Electronics, vol. 38, no. 8, pp. 114-117, Apr. 1965

[3] D. Llorca, D. Colas, I. Daza, I. Parra, and M. Sotelo, “Vehicle model recognition using geometry and appearance of car emblems from rear view images,” in 17th IEEE International Conference on Intelligent Transportation Systems, pp. 3094–3099, Oct 2014

[4] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen, “Symmetrical SURF and Its Applications to B. Zhang, “Reliable classification of vehicle types based on cascade classifier ensembles,” IEEE transactions on Intelligent Transportation Systems, vol. 14, pp. 322–332, March 2013.

[5]] V. Varjas and A. Tanacs, “Car recognition from frontal images in mobile environment,” in Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium,pp. 819–823, Sept 2013.

[6] M. AbdelMaseeh, I. Badreldin, M. Abdelkader, and M. El Saban, “Car Make and Model recognition combining global and local cues,” in 21st International Conference on Pattern Recognition (ICPR), pp. 910–913, Nov 2012.

[7] G. Pearce and N. Pears, “Automatic make and model recognition from frontal images of cars,” in 8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), pp. 373–378, Aug 2011.

[8] A. Psyllos, C. Anagnostopoulos, and E. Kayafas, “Vehicle model recognition from frontal view image measurements,” Computer Standards & Interfaces, vol. 33, no. 2, pp. 142 – 151, 2011. XVI IMEKO TC4 Symposium Exploring New Frontiers of Instrumentation ¨ and Methods for Electrical and Electronic

Measurements and XIII International Workshop ¨ on ADC Modelling and Testing.

[9] A. Boukerche, J. Feng, and X. Fei, “V-Square: An Accurate Time Synchronization Protocol for Wireless Video Sensor Networks,” in Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE, pp. 1–5, Nov 2008.

[10] A. Boukerche, “Wireless multimedia sensor and actuator system: A necessary public security and safety testbed for an urban emergency preparedness class of applications,” in Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on, pp. 1–1, May 2010.

[11] C. Rezende, A. Mammeri, A. Boukerche, and A. A. Loureiro, “A receiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection,” Ad Hoc Networks, vol. 17, pp. 1 – 17, 2014.

[12] C. Rezende, A. Boukerche, M. Almulla, and A. A. Loureiro, “The selective use of redundancy for video streaming over vehicular ad hoc networks,” Computer Networks, vol. 81, pp. 43 – 62, 2015.

[13] C. Rezende, A. Boukerche, H. Ramos, and A. Loureiro, “A reactive and scalable unicast solution for video streaming over vanets,” IEEE Transactions on Computers, vol. 64, pp. 614–626, March 2015.

[14] Y. Li and A. Boukerche, “Qugu: A quality guaranteed video dissemination protocol over urban vehicular ad hoc networks,” ACM Trans. Multimedia Comput. Commun. Appl.,vol. 11, pp. 55:1–55:23, June 2015.

[15] R. Wang, M. Almulla, C. Rezende, and A. Boukerche, “Video streaming over vehicular networks by a multiple path solution with error correction,” in 2014 IEEE International Conference on Communications (ICC), pp. 580–585, June 2014.

[16] Y. Li, F. Naeimipoor, and A. Boukerche, “Video dissemination protocols in urban vehicular ad hoc network: A performance evaluation study,” in Wireless Communications and Networking Conference (WCNC), 2014 IEEE, pp. 2611–2616, April 2014.

[17] R. Wang, C. Rezende, H. Ramos, R. Pazzi, A. Boukerche, and A. Loureiro, "Liaithon: A location-aware multipath video streaming scheme for urban vehicular networks," in 2012 IEEE Symposium on Computers and Communications (ISCC), pp. 000436–000441, July 2012.

[18] F. Naeimipoor, C. Rezende, and A. Boukerche, "Performance evaluation of video dissemination protocols over vehicular networks," in Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on, pp. 694–701, Oct 2012.

[19] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, "Localization systems for wireless sensor networks," Wireless Communications, IEEE, vol. 14, pp. 6–12, December 2007.

[20] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, "Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems," Computer Communications , vol. 31, no. 12, pp. 2838 – 2849, 2008. Mobility Protocols for ITS/VANET.

[21] A. Boukerche, ed., Algorithms and protocols for wireless sensor networks. John Wiley Sons, 2008.

[22] H. de Oliveira, A. Boukerche, E. Freire Nakamura, and A. Loureiro, "An efficient directed localization recursion protocol for wireless sensor networks," IEEE Transactions on Computers, vol. 58, pp. 677–691, May 2009.

[23] H. A. Oliveira, A. Boukerche, E. F. Nakamura, and A. A. Loureiro, "Localization in time and space for wireless sensor networks: An efficient and lightweight algorithm," Performance Evaluation, vol. 66, no. 35, pp. 209 – 222, 2009. Modeling and Analysis of Wireless Networks: Selected Papers from {MSWiM} 2007.

[24] O. Abumansoor and A. Boukerche, "A secure cooperative approach for nonline-of-sight location verification in vanet," Vehicular Technology, IEEE Transactions on, vol. 61, pp. 275–285, Jan 2012.

[25] S. Samarah, M. Al-Hajri, and A. Boukerche, "A predictive energy-efficient technique to support object-tracking sensor networks," IEEE Transactions on Vehicular Technology, vol. 60, pp. 656–663, Feb 2011.

[26] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo, “A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications,” in Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless

[27] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo, “Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments,” Journal of Parallel and Distributed Computing, vol. 66, no. 4, pp. 586 – 599, 2006. Algorithms for Wireless and Ad-Hoc Networks.

[28] R. W. Pazzi and A. Boukerche, “Mobile data collector strategy for delay-sensitive applications over wireless sensor networks,” Computer Communications, vol. 31, no. 5, pp. 1028 – 1039, 2008. Mobility Management and Wireless Access.

[29] M. Elhadef, A. Boukerche, and H. Elkadiki, “A distributed fault identification protocol for wireless and mobile ad hoc networks,” Journal of Parallel and Distributed Computing, vol. 68, no. 3, pp. 321 – 335, 2008. Wireless Mesh Networks: Behavior, Artifacts and Solutions.

[30] A. Boukerche, A. Martirosyan, and R. Pazzi, “An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks,” Mob. Netw. Appl.,vol. 13, pp. 614–626, Dec. 2008.

[31] A. Boukerche, R. Pazzi, and R. Araujo, “Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol,” in Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on, pp. 560–567, Nov 2005.

[32] A. Prati and F. Qureshi, “Integrating consumer smart cameras into camera networks: Opportunities and obstacles,” Computer, vol. 47, pp. 45–51, May 2014.

[33] A. Boukerche, S. Hong, and T. Jacob, “An efficient synchronization scheme of multimedia streams in wireless and mobile systems,” Parallel and Distributed Systems, IEEE Transactions on, vol. 13, pp. 911–923, Sep 2002. 110.

[34] A. Boukerche and D. Turgut, “Secure time synchronization protocols for wireless sensor networks,” Wireless Communications, IEEE, vol. 14, pp. 64–69, October 2007.

[35] Z. Zhang, R. W. Pazzi, and A. Boukerche, “A mobility management scheme for wireless mesh networks based on a hybrid routing protocol,” Computer Networks, vol. 54, no. 4, pp. 558 – 572, 2010. Advances in Wireless and Mobile Networks.

[36] Y. Ren, R. Pazzi, and A. Boukerche, “Monitoring patients via a secure and mobile healthcare system,” Wireless Communications, IEEE, vol. 17, pp. 59–65, February 2010.

[37] A. Boukerche and Y. Ren, “A secure mobile healthcare system using trust-based multicast scheme,” Selected Areas in Communications, IEEE Journal on, vol. 27, pp. 387–399, May 2009.

[38] Y. Ren and A. Boukerche, “Modeling and managing the trust for wireless and mobile adhoc networks,” in Communications, 2008. ICC ’08. IEEE International Conference on,pp. 2129–2133, May 2008.

[39] A. Boukerche and Y. Ren, “A trust-based security system for ubiquitous and pervasive computing environments,” Computer Communications, vol. 31, no. 18, pp. 4343 – 4351,2008. Secure Multi-Mode Systems and their Applications for Pervasive Computing.

[40] A. Boukerche and X. Li, “An agent-based trust and reputation management scheme for wireless sensor networks,” in Global Telecommunications Conference, 2005. GLOBECOM ’05. IEEE, vol. 3, pp. 5 pp.–, Nov 2005.

[41] X. Ma and W. Grimson, “Edge-based rich representation for vehicle classification,” in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2, pp. 1185–1192 Vol. 2, Oct 2005.

[42] L. Zhang, S. Li, X. Yuan, and S. Xiang, “Real-time object classification in video surveillance based on appearance learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2007.

[43] N. Buch, J. Orwell, and S. A. Velastin, “3d extended histogram of oriented gradients (3DHOG) for classification of road users in urban scenes,” in British

Machine Vision Conference, London, UK, September 7-10, pp. 1–11, 2009.

[44] Z. Chen, T. Ellis, and S. Velastin, "Vehicle type categorization: A comparison of classification schemes," in Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, pp. 74–79, Oct 2011.

[45] Z. Chen, T. Ellis, and S. Velastin, "Vehicle detection, tracking and classification in urban traffic," in 15th International IEEE Conference on Intelligent Transportation Systems,pp. 951–956, Sept 2012.

[46] R. Wang, L. Zhang, K. Xiao, R. Sun, and L. Cui, "Easisee: Real-time vehicle classification and counting via low-cost collaborative sensing," IEEE Transactions on Intelligent Transportation Systems, vol. 15, pp. 414–424, Feb 2014.

[47] Z. Dong, M. Pei, Y. He, T. Liu, Y. Dong, and Y. Jia, "Vehicle type classification using unsupervised convolutional neural network," in 22nd International Conference on Pattern Recognition (ICPR), pp. 172–177, Aug 2014.

[48] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semi supervised convolutional neural network," IEEE Transactions on Intelligent Transportation Systems, vol. PP, no. 99, pp. 1–10, 2015.

[49] A. Psyllos, C.-N. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a sift-based enhanced matching scheme," Intelligent Transportation Systems, IEEE Transactions on, vol. 11, pp. 322–328, June 2010.

[50] S. Yu, S. Zheng, H. Yang, and L. Liang, "Vehicle logo recognition based on bag-of-words,"in Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on, pp. 353–358, Aug 2013.

[51] D. Llorca, R. Arroyo, and M. Sotelo, "Vehicle logo recognition in traffic images using hog features and svm," in Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on, pp. 2229–2234, Oct 2013.

[52] Y. Ou, H. Zheng, S. Chen, and J. Chen, "Vehicle logo recognition based on a weighted spatial pyramid framework," in IEEE 17th International Conference

on Intelligent Transportation Systems (ITSC), pp. 1238–1244, Oct 2014.

[53] J. Xiao, W. Xiang, and Y. Liu, "Vehicle logo recognition by weighted multi-class support vector machine ensembles based on sharpness histogram features," Image Processing, IET, vol. 9, no. 7, pp. 527–534, 2015.

[54] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," Intelligent Transportation Systems, IEEE Transactions on, vol. 16, pp. 1951–1960, Aug 2015.

[55] H. Peng, X. Wang, H. Wang, and W. Yang, "Recognition of low-resolution logos in vehicle images based on statistical random sparse distribution," Intelligent Transportation Systems, IEEE Transactions on, vol. 16, pp. 681–691, April 2015.

[56] P. Badura and M. Skotnicka, "Automatic Car Make Recognition in Low-Quality Images," in Information Technologies in Biomedicine, Volume 3 (E. Pitka, J. Kawa, and W. Wieclawek, eds.), vol. 283 of Advances in Intelligent Systems and Computing, pp. 235–246, Springer International Publishing, 2014.

[57] L. Dlagnekov, "Video-based Car Surveillance: License Plate, Make, and Model recognition," Master's thesis, University of California, San Diego, 2005.

[58] D. Lowe, "Object recognition from local scale-invariant features," in 7th IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 vol.2, 1999.

[59] R. Baran, A. Glowacz, and A. Matiolanski, "The efficient real- and non-real-time make and model recognition of cars," Multimedia Tools and Applications, pp. 1–20, 2013.

[60] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF),"Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346 – 359, 2008.

[61] D. Jang and M. Turk, "Car-rec: A real time car recognition system," in IEEE Workshop on Applications of Computer Vision (WACV), pp. 599–605, Jan

2011.

[62] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 vol. 1, June 2005.

[63] V. Petrovic and T. Cootes, “Vehicle type recognition with match refinement,” in 17th IEEE International Conference on Pattern Recognition, vol. 3, pp. 95–98 Vol.3, Aug 2004.

[64] D. T. Munroe and M. G. Madden, “Multi-Class and Single-Class Classification Approaches to Vehicle Model Recognition from Images,” in 16th Irish Conference on Artificial Intelligence and Cognitive Science, pp. 93–104, Sep 2005.

[65] X. Clady, P. Negri, M. Milgram, and R. Poulenard, “Multi-class vehicle type recognition system,” in Artificial Neural Networks in Pattern Recognition (L. Prevost, S. Marinai, and F. Schwenker, eds.), vol. 5064 of Lecture Notes in Computer Science, pp. 228–239,

[66] Springer Berlin Heidelberg, 2008. [74] P. Salembier and T. Sikora, Introduction to MPEG-7: Multimedia Content Description Interface. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[67] D. K. Park, Y. S. Jeon, and C. S. Won, “Efficient Use of Local Edge Histogram Descriptor,” in Proceedings of the 2000 ACM Workshops on Multimedia, MULTIMEDIA ’00, (New York, NY, USA), pp. 51–54, ACM, 2000.

[68] H. Lee, “Neural Network Approach to Identify Model of Vehicles,” in Advances in NeuralNetworks - ISNN 2006 (J. Wang, Z. Yi, J. Zurada, B.-L. Lu, and H. Yin, eds.), vol. 3973 of Lecture Notes in Computer Science, pp. 66–72, Springer Berlin Heidelberg, 2006.

[69] I. Zafar, E. A. Edirisinghe, S. Acar, and H. E. Bez, “Two-dimensional statistical linear discriminant analysis for real-time robust vehicle-type recognition,” Proc. SPIE, International Conference on Real-Time Image Processing, vol. 6496, pp. 649602–649602–8,2007.

[70] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,”

International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[71] U. Sinha, “SIFT: Generating a feature.” Last accessed on 15-August-2015.

[72] A. Mammeri, A. Boukerche, and M. Zhao, “Keypoint-based binocular distance measurement for pedestrian detection system,” in Proceedings of the Fourth ACM International Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, DIVANet ’14, (New York, NY, USA), pp. 9–15, ACM, 2014.

[73] A. Mammeri, D. Zhou, A. Boukerche, and M. Almulla, “An efficient animal detectionsystem for smart cars using cascaded classifiers,” in Communications (ICC), 2014 IEEE International Conference on, pp. 1854–1859, June 2014.

[74] A. Mammeri, T. Zuo, and A. Boukerche, “Extending the Detection Range of Vision-based Driver Assistance Systems Application to Pedestrian Protection System,” in 2014 IEEE Global Communications Conference (GLOBECOM), pp. 1358–1363, Dec 2014.

[75] A. Mammeri, A. Boukerche, J. Feng, and R. Wang, “North-american speed limit sign detection and recognition for smart cars,” in 2013 IEEE 38th Conference on Local Computer Networks Workshops (LCN Workshops), pp. 154–161, Oct 2013.

[76] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Li, “Learning multi-scale block local binary patterns for face recognition,” in Advances in Biometrics (S.-W. Lee and S. Li, eds.), vol. 4642 of Lecture Notes in Computer Science, pp. 828–837, Springer Berlin Heidelberg,2007.

[77] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 971–987, Jul 2002.

[78] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3360–3367, June 2010.

[79] H. Jegou, M. Douze, C. Schmid, and P. Perez, “Aggregating local descriptors into a compact image representation,” in Computer Vision and Pattern

Recognition (CVPR), 2010 IEEE Conference on, pp. 3304–3311, June 2010.

[80] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 34, pp. 1704–1716, Sept 2012.

[81] X. Zhou, K. Yu, T. Zhang, and T. Huang, "Image classification using super-vector coding of local image descriptors," in Computer Vision ECCV 2010 (K. Daniilidis, P. Maragos, and N. Paragios, eds.), vol. 6315 of Lecture Notes in Computer Science, pp. 141–154, Springer Berlin Heidelberg, 2010.

[82] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II, (Cambridge, MA, USA), pp. 487–493, MIT Press, 1999.

[83] F. Perronnin, J. Snchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in Computer Vision ECCV 2010 (K. Daniilidis, P. Maragos, and N. Paragios, eds.), vol. 6314 of Lecture Notes in Computer Science, pp. 143–156, Springer Berlin Heidelberg, 2010.

[84] Y. Ke and R. Sukthankar, "Pca-sift: a more distinctive representation for local image descriptors," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II–506–II–513 Vol.2, June 2004.

[85] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," in Proceedings of the Ninth IEEE International Conference on Computer Vision, pp. 1470–1477 vol.2, Oct 2003.

[86] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual Categorization with Bags of Keypoints," in Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22, 2004.

[87] L. Nanni and A. Lumini, "Heterogeneous Bag-of-Features for Object/Scene Recognition,"Applied Soft Computing, vol. 13, no. 4, pp. 2171 – 2178, 2013.

[88] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in Proceedings of the IEEE

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 923–930, June 2013.

[89] F. Shi, E. Petriu, and R. Laganiere, “Sampling strategies for real-time action recognition,” in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on,pp. 2595–2602, June 2013.

[90] P. Pinto, A. Tome, and V. Santos, “Visual detection of vehicles using a bag-of-features approach,” in 13th International Conference on Autonomous Robot Systems (Robotica),pp. 1–4, April 2013.

[91] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2169–2178, 2006.

[92] S. Singh, S. Choudhury, K. Vishal, and C. Jawahar, “Currency Recognition on Mobile Phones,” in 22nd International Conference on Pattern Recognition (ICPR), pp. 2661–2666, Aug 2014.

[93] D. Arthur and S. Vassilvitskii, “K-means++: The Advantages of Careful Seeding,” in 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1027–1035, 2007.

[94] L. Hazelhoff, I. Creusen, and P. de With, “Optimal Performance-Efficiency Trade-off for Bag of Words Classification of Road Signs,” in 22nd International Conference on Pattern Recognition (ICPR), pp. 2996–3001, Aug 2014.

[95] Y.-G. Jiang, C.-W. Ngo, and J. Yang, “Towards Optimal Bag-of-features for Object Categorization and Semantic Video Retrieval,” in 6th ACM International Conference on Image and Video Retrieval (CIVR), pp. 494–501, ACM, 2007.

[96] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” Data Mining and Knowledge Discovery, vol. 2, pp. 121–167, June 1998.

[97] V. N. Vapnik, The Nature of Statistical Learning Theory. New York, NY, USA: SpringerVerlag New York, Inc., 1995.

[98] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, pp. 273–297, Sept. 1995.

[99] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT ’92, (New York, NY, USA), pp. 144–152, ACM, 1992.

[100] B. Scholkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, “New support vector ¨ algorithms,” Neural Comput., vol. 12, pp. 1207–1245, May 2000.

[101] M. G. Genton, “Classes of kernels for machine learning: A statistics perspective,” J. Mach.Learn. Res., vol. 2, pp. 299–312, Mar. 2002.

[102] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[103] T. K. Ho, “Random decision forests,” in Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on, vol. 1, pp. 278–282 vol.1, Aug 1995.

[104] A. Bosch, A. Zisserman, and X. Muoz, “Image classification using random forests and ferns,” in IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007.,pp. 1–8, Oct 2007.

[105] F. Zaklouta and B. Stanciulescu, “Real-time traffic-sign recognition using tree classifiers,”IEEE Transactions on Intelligent Transportation Systems, vol. 13, pp. 1507–1514, Dec2012.

[106] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, “Sparse representation for computer vision and pattern recognition,” Proceedings of the IEEE, vol. 98, pp. 1031– 1044, June 2010.

[107] L. Wang, H. Wu, and C. Pan, “Manifold regularized local sparse representation for facerecognition,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 25,pp. 651–659, April 2015.

[108] J. Chen and Z. Yi, “Sparse representation for face recognition by discriminative low-rank matrix recovery ,” Journal of Visual Communication and Image Representation , vol. 25,no. 5, pp. 763 – 773, 2014.

[109] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, “Robust face recognition via adaptive sparse representation,” IEEE Transactions on Cybernetics, vol. 44, pp. 2368–2378, Dec 2014.

[110] S. Shekhar, V. Patel, N. Nasrabadi, and R. Chellappa, “Joint sparse representation for robust multimodal biometrics recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, pp. 113–126, Jan 2014.

[111] V. M. Patel and R. Chellappa, “Sparse Representation-based Object Recognition,” in Sparse Representations and Compressive Sensing for Imaging and Vision, Springer Briefs in Electrical and Computer Engineering, pp. 63–84, Springer New York, 2013.

[112] H.-F. Chiang, J.-W. Hsieh, C.-H. Chuang, K.-T. Chuang, and Y. Yan, “Modeling and recognizing action contexts in persons using sparse representation,” Journal of Visual Communication and Image Representation, vol. 30, pp. 252 – 265, 2015.

[113] X. Zhu, J. Liu, J. Wang, C. Li, and H. Lu, “Sparse representation for robust abnormality detection in crowded scenes,” Pattern Recognition, vol. 47, no. 5, pp. 1791 – 1799, 2014.

[114] S. H. Lee, K. Kostas Plataniotis, and Y. M. Ro, “Intra-Class Variation Reduction Using Training Expression Images for Sparse Representation Based Facial Expression Recognition,” IEEE Transactions on Affective Computing, vol. 5, pp. 340–351, July 2014.

[115] K. B. Raja, R. Raghavendra, V. K. Vemuri, and C. Busch, “Smartphone based visible iris recognition using deep sparse filtering,” Pattern Recognition Letters, vol. 57, pp. 33 – 42, 2015.

[116] M. Aharon, M. Elad, and A. Bruckstein, “ K -SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation,” IEEE Transactions on Signal Processing, vol. 54, pp. 4311–4322, Nov 2006.

[117] S. Li and A. Jain, eds., Handbook of Face Recognition. Springer-Verlag New York, 2005.

[118] R. Bryll, R. Gutierrez-Osuna, and F. Quek, “Attribute Bagging: Improving

Accuracy of Classifier Ensembles by Using Random Feature Subsets," Pattern Recognition, vol. 36, no. 6, pp. 1291 – 1302, 2003.

[119] N. Chawla and K. Bowyer, "Random subspaces and subsampling for 2-d face recognition," in IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 582–589 vol.2, June 2005.

[120] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, pp. 1088–1099, July 2006.

[121] L. Hazelhoff, I. Creusen, D. van de Wouw, and P. H. N. de With, "Large-scale classification of traffic signs under real-world conditions," in Proc. SPIE 8304, Multimedia on Mobile Devices 2012; and Multimedia Content Access: Algorithms and Systems VI, 83040W, vol. 8304, pp. 83040W–83040W–10, 2012.

[122] Dash, Dipak K. "This Is What Makes Indian Road Deadliest." TOI, TIMESOFINDIA,29Oct.2022,09:13PM,timesofindia.indiatimes.com/india/this-is-what-makes-indian-roads-deadliest/articleshow/78870630.cms.

[123] Ravindra Joshi," Speeding car rams into truck near new Katraj tunnel, four dead"HindustanTimes,Pune,DEC12,201711:07PM,"https://www.hindustantimes.com/pune-news/four-including-3-of-a-family-die-in-mishap-in-pune/story-B2mHRMzbv8jXnq8qWnk5TP.html".

[124] Accident Research Cell, "Roads Accidents in Maharashtra "Dec 2019 15.30 pm"https://highwaypolice.maharashtra.gov.in/wpcontent/uploads/2020/09/highway-traffic-book-2019.pdf"

[125] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 27:1–27:27, 2011.

[126] S. Sivaraman and M. Trivedi, "Looking at vehicles on the road: A survey of visionbased vehicle detection, tracking, and behavior analysis," IEEE Transactions on Intelligent Transportation Systems, vol. 14, pp. 1773–1795, Dec 2013.

[127] M. Cheon, W. Lee, C. Yoon, and M. Park, “Vision-based vehicle detection system with consideration of the detecting location,” Sept 2012.

[128] A. Mammeri, E.-H. Khiari, and A. Boukerche, “Road-Sign Text Recognition Architecture for Intelligent Transportation Systems,” in 80th IEEE Vehicular Technology Conference (VTC Fall), pp. 1–5, Sept 2014.

[129] A. Al-Ghaili, S. Mashohor, A. Ramli, and A. Ismail, “Vertical-edge-based car-license plate detection method,” IEEE Transactions on Vehicular Technology, vol. 62, pp. 26–38, Jan 2013.

[130] B. Tian, Y. Li, B. Li, and D. Wen, “Rear-View Vehicle Detection and Tracking by Combining Multiple Parts for Complex Urban Surveillance,” IEEE Transactions on Intelligent Transportation Systems, vol. 15, pp. 597–606, April 2014.

[131] G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, “Application-oriented license plate recognition,” IEEE Transactions on Vehicular Technology, vol. 62, pp. 552–561, Feb 2013.

[132] M. Sarfraz, A. Shahzad, M. Elahi, M. Fraz, I. Zafar, and E. Edirisinghe, “Real-time automatic license plate recognition for cctv forensic applications,” Journal of Real-Time Image Processing, vol. 8, no. 3, pp. 285–295, 2013.