A PROJECT REPORT ON

VEHICLE DETECTION AND SPEED DETECTION

Submitted in partial fulfillment
of the Requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

Submitted by

I.TEJA VENKATA RAM
Regd.noY23MC13036

Under the guidance of
Smt V.RAMA DEVI
Associate Professor &HOD

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
HINDU COLLEGE PG COURSES
GUNTUR -522007

ACHARYA NAGARJUNA UNIVERSITY
GUNTUR
APRIL -2024

ALICE STAFFING
FIRST FLOOR INNER STOP, MG INNER RING ROAD,REDDYPALEM,GUNTUR-522509.

TO
THE HEAD OF DEPARTMENT,
DEPARTMENT OF MCA,
HINDU COLLEGE PG COURSES,
GUNTUR,
ANDHRA PRADESH.

PROJECT COMPLETION CERTIFICATE

This is to certify that **Mr. Immadisetty Teja Venkata Ram** with RegNo **Y23MC13036,** has successfully completed his Project Title “**Vehicle detection And Speed Detection”.**

He has done this project using **python** during **October 2023-March 2024** under the supervision of “**ALICE STAFFING ”.**

He has completed the assigned project well within the time frame. He is sincere, working and him conduct during the project is clear.

We wish him all the best in his future and endeavors
ALICE STAFFING.

Manager-Training -Project

DECLARATION

I, **I.TEJA VENKATA RAM** hereby declare that the project report entitled "**Vehicle Detection And Speed Detection**" done by me under the guidance of **Smt V.RAMA DEVI ,** Associate Professor &HOD being submitted to the Department of MCA, Approved by AICTE and affiliated to Acharya Nagarjuna University, Guntur for the award of degree of **MASTER OF COMPUTER APPLICATIONS.** This is a record of bonafide work done by me and it has not been submitted to any other Institute or University for the award of any other degree.

(I.TEJA VENKATA RAM)
Y23MC13036

ACKNOWLEDGEMENT

I feel myself honored to place my warm salutation to **HINDU COLLEGE PG COURSES, GUNTUR,** Department of MCA, which gave me the opportunity to have a strong base in MCA and profound knowledge.

I wish to express my profound deep sense of gratitude to **DR CH.SUBBARAO,** Principal, **HINDU COLLEGE P.G COURSES** for his effective encouragement and enthusiastic guidance for the successful completion of my project work and for providing me all the necessary amenities that helped me to complete my project work within the stipulated time.

I am indebted to **Smt V. RAMADEVI M.Tech, Head of the Department** for his continuous support and valuable suggestions for successful completion of my project work.

I express my thanks to the organization **" ALICE STAFFING "** for having confidence in me and making use of my activities by offering this project. I also express my special thanks to **Mr. V.VENKATA KARTHIK**, Director for guiding me in right way and helped me in this project.

Finally, I take this opportunity to extend my full-heartened thanks, gratitude and respect to the Department Lecturers, all my friends, parents and well-wishers for giving me valuable advice and constant support in all possible ways.

ABSTRACT

This master's thesis focuses on vehicle detection and tracking. The research tries to detect vehicles in images and videos. It deploys a dataset from Udacity in order to train the algorithms. Two machine learning algorithms Support Vector Machine(SVM)and Decision Tree have been developed for the detection and tracking tasks. Python programming language have been utilized as the development language for the creation and training of both models. These two algorithms have been developed, trained, tested, and compared to each other to specify the weaknesses and strengths of each of them, although to present and suggest the best model among these two. For the evaluation purpose multiple techniques are used in order to compare and identify the more accurate model. The primary goal and target of the thesis is to develop a system in which the system should be able to detect and track the vehicles automatically whether they are static or moving in images and videos.

Vehicle detection also called computer vision object recognition, basically the scientific methods and ways of how machines see rather than human eyes. The main duty of a vehicle detection system is to localize one or more vehicles in put images.The results showed that SVM out performed the Decision Tree and has acceptable accuracy for the vehicle detection and tracking tasks.

Keywords: Vehicle detection; Vehicle Tracking; SVM ;Decision Tree; Image detection ; Object detection and Tracking

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CNN:	Convolutional Neural Networks
KNN:	KNearest Neighbors
LR:	Linear Regression
ML:	Machine Learning
MLP:	Multi-Layer Perceptron
RELU:	Rectified Linear Unit
RMSE:	Root Mean Squared Error
MSE:	Mean Squared Error
MAE:	Mean Absolute Error
AI:	Artificial Intelligence
SVR:	Support Vector Regression
SVM:	Support Vector Machine
OCR:	Optical Character Recognition
FTS:	Fuzzy Time Series

INTRODUCTION

Background

Since the population and transport system increase day by day, the demand for managing them increase at the same time. The world is getting populated so fast. Therefore the number of machines from any types including vehicles increased at the same time. That being said, new topics like traffic, accidents and many more issues are needed to be managed.It is hard to manage them with the old methods new trends and technologies have been found and invented to handle each and every milestone that human kind is trying achieve. One of these challenges is traffic in highways and cities. Many options like traffic light, sign, etc. deployed in order to deal with this phenomena. It seems that these options are not enough or not so efficient alone. New technologies like object detection and tracking are invented in order to utilize automated camera surveillance to produced at a that can give meanings for a decision making process.This phenomena have been used for different kind of issues. The new trend Intelligent Transport System (ITS) has many elements which object detection and tracking is one of them. This system is used to detect vehicles, lanes, traffic sign, or vehicle make detection. The vehicle detection and classify ability gives us the possibility to improve the traffic flows and roads, prevent accidents and registering traffic crimes and violations.

Humans can easily recognize vehicles in videos or image sort to identify different types of cars.In computer algorithms and programs it is highly depend on the types of data. Some challenges like the weather or light are also plays important role on making the process easy or much hard.At the same time we have different types and shapes of vehicles. More than that the new challenge could be to identify moving objects in a video in real time where they are different in size and shape.

There are different techniques and methods for vehicle detection and classification. The variety of these techniques are in types of algorithms like Support Vector Machine(SVM),Convolutional Neural Network(CNN),Decision Tree, Recurrent Neural Network (RNN) etc. The field is constantly evolving since the industry is focused on this system or Computer visionary. In this thesis we investigate two algorithms SVM and Decision Tree to identify how they can apply in the field and which one works better than the other.

The Problem

Object detection attracted the attention in research industry lately. Researchers are trying to explore the topic to reach to an accepted accuracy level. Machine learning is used to detect objects. There are many techniques doing this job but in order to identify the best model among the suggested models, this thesis is going to explore the topic on how to detect the objects while at the same time compare two suggested models and suggest the best one which yields highest accuracy and performance.

The task of detecting and classifying objects in images and videos is suited well in machine learning since the task is a classification task. The reason behind this is from the dataset with complex features. The system is important for many fields especially for traffic and vehicle detection.

The training and analysis was done using a rich dataset which is described in details in chapter4.

Aim of the Study

The aim of this thesis is to develop two algorithms, SVM and Decision Tree to detect vehicles in images and videos. It compares the two algorithm with the same dataset and pre processing methods. Finally suggest the method with high accuracy and performance level. The these is going to implement two classifier which are able to predict the class of the image whether its Vehicles or Non-Vehicles. The vehicle detector will also predict the bounding box coordinates of the vehicles.

Significance of the Study

Since the industrial revolution the number of cars increase day by day. One of the new challenge of the world is the traffic.People in cities wastes most of their time in traffic going some where in the city. Having a digitized traffic system which is functioning 24/7 and make the tasks so easy and efficient is crucial for all countries around the globe. Therefore having a computerized traffic system cannot be handled without having an accurate vehicle detection system. The system obviously affects the economy, the life of citizens,the industry, etc.

That being said, it is necessary to contribute in the topic until we reach to a level which the algorithms can accurately detect all kind of objects or vehicles whether they are in images or videos through pipelines.

The Limitations of the Study

Despite the fact that the thesis reached to its goals, it would have been more accurate if the dataset is improved and the number of objects in vehicles and non-vehicles class increased.At the same time it would be more complete and proper claim if the comparison would have conducted between many more options in the algorithms which are used in computer visionary problems.

Overview of the Study

The is consist of six chapters in all:

Chapter1: Gives a general introduction and back ground about the topic. This chapter describes the aim of the thesis, its importance and over view of the research.

Chapter2: Past researches which are related to the topic are reviewed ,their techniques and results are also discussed.

Chapter3: This chapter is focused on machine learning techniques, theory, formula, implementation and philosophy..

Chapter4: ML algorithms and models used in this these is will be explained in details with their backend formulas.

Chapter5: The result and out come is presented.

Chapter6: The conclusion is presented with a general view and future recommendations for research topic improvement.

SYSTEM ANALYSIS

SYSTEM CONSTRAINTS

System Constraints for Vehicle Detection and Speed Detection System in Java

Introduction:

Understanding the system constraints is crucial for the successful development and deployment of the Vehicle Detection and Speed Detection System in Java. These constraints encompass various aspects, including technical limitations, resource constraints, environmental considerations, and potential challenges that may impact the system's performance. This comprehensive analysis delves into the intricacies of these constraints to ensure a realistic assessment of the system's capabilities and limitations.

1. Technical Constraints

The technical constraints of the system primarily revolve around the limitations inherent in computer vision and image processing technologies. One significant technical constraint is the reliance on the quality of input data from surveillance cameras and sensors. Variations in lighting conditions, occlusions, and image distortions may pose challenges to accurate vehicle detection and speed estimation. Additionally, the computational demands of real-time processing can be a limiting factor, particularly when dealing with high-resolution video streams or multiple camera inputs.

Another technical constraint is the potential inaccuracies in vehicle detection algorithms. While state-of-the-art computer vision techniques, such as Haar cascades and deep learning, offer high accuracy, they may still exhibit limitations in certain scenarios, such as crowded traffic or complex road geometries. Balancing the trade-off between accuracy and computational efficiency is a technical challenge that needs to be carefully addressed.

2. Resource Constraints:

Resource constraints encompass both hardware and software limitations that may impact the system's performance. From a hardware perspective, the computational power of the devices running the Java-based application plays a crucial role. Low-powered devices may struggle with real-time processing demands, affecting the system's responsiveness and accuracy. Adequate memory and storage resources are also essential for handling large volumes of video data and maintaining historical records for speed estimation.

On the software side, dependencies on third-party libraries, such as OpenCV for Java, introduce considerations regarding version compatibility and updates. Ensuring that the system remains compatible with the latest versions of these libraries is essential for leveraging improvements and security patches. Moreover, the reliance on Java for cross-platform compatibility may introduce challenges related to Java Virtual Machine (JVM) variations and compatibility issues across different operating systems.

3. Environmental Constraints:

The system's performance is susceptible to environmental conditions, such as adverse weather and varying illumination. Heavy rain, fog, or extreme sunlight may degrade the quality of input data, impacting the accuracy of vehicle detection algorithms. Adequate measures, such as the use of weather-resistant cameras and sensors or the incorporation of additional environmental sensors, may be necessary to mitigate the effects of adverse weather conditions.

Environmental constraints also extend to the geographical layout of roads. Complex road structures, intersections, and diverse traffic scenarios may pose challenges for accurate vehicle tracking and speed estimation. The system must be adaptable to different urban and highway environments, considering variations in road layouts and traffic patterns.

4. Regulatory and Legal Constraints:

Compliance with regulatory and legal frameworks is a critical constraint that influences the system's design and operation. Privacy laws and data protection regulations dictate how the system handles and stores sensitive information, such as license plate numbers and vehicle movements. Striking a balance between effective traffic monitoring and privacy preservation is a regulatory challenge that must be addressed through careful system design and adherence to legal requirements.

The enforcement of speed limits and traffic rules based on the system's outputs introduces legal considerations. The accuracy and reliability of speed estimation algorithms must meet regulatory standards to ensure fair and justifiable law enforcement actions. The system's compliance with existing traffic management regulations is a pivotal constraint that influences its acceptance and effectiveness.

5. Scalability Constraints:

Scalability constraints revolve around the system's ability to handle increasing data loads and adapt to evolving traffic scenarios. As traffic volumes fluctuate and new vehicles and technologies emerge, the system must scale seamlessly to accommodate these changes. The scalability of the Java-based application, both in terms of processing capabilities and user interface responsiveness, becomes a crucial consideration.

Integrating additional features or expanding the system to cover larger geographical areas may introduce scalability challenges. Ensuring that the system architecture allows for modular enhancements and efficient scaling without compromising performance is a key constraint. Considerations for database scalability, real-time data processing, and user interface responsiveness play a pivotal role in addressing scalability constraints.

6. Interoperability Constraints:

Interoperability constraints pertain to the system's ability to integrate and communicate with existing traffic management infrastructure, surveillance systems, and data sources. Ensuring compatibility with various camera models, sensor types, and communication protocols becomes a critical consideration. The system must be designed to handle diverse inputs seamlessly and provide standardized outputs that can be easily integrated into broader traffic management systems.

Interoperability constraints extend to communication with external databases or cloud services. The system's ability to securely exchange data with external platforms, comply with industry standards, and facilitate interoperability with emerging technologies becomes a crucial factor for long-term viability.

Conclusion:

In conclusion, the system constraints for the Vehicle Detection and Speed Detection System in Java encompass a spectrum of technical, resource, environmental, regulatory, scalability, and interoperability considerations. Acknowledging and addressing these constraints during the design and implementation phases are essential for ensuring the system's robustness, reliability, and compliance with legal and regulatory standards. The comprehensive understanding of these constraints provides a foundation for developing adaptive solutions, mitigating risks, and optimizing the system's performance in real-world traffic management scenarios.

FEASABILITY STUDY

Feasibility Study for Vehicle Detection and Speed Detection System in Java

Introduction:

The feasibility study serves as a critical examination and analysis of the proposed Vehicle Detection and Speed Detection System implemented in Java. This study aims to assess the viability, practicality, and potential benefits of developing such a system, considering technical, economic, operational, and scheduling aspects.

1. Technical Feasibility:

The technical feasibility of the project revolves around the system's capability to employ advanced computer vision and image processing techniques. Java, with its extensive libraries, proves technically feasible for implementing complex algorithms, ensuring accurate vehicle detection and speed estimation. The availability of reliable and efficient image processing tools, such as OpenCV for Java, enhances the system's technical viability.

2. Operational Feasibility:

Operational feasibility assesses how well the proposed system aligns with existing operations and meets the operational requirements of traffic management. The system, integrated with surveillance cameras and sensors, aligns seamlessly with current traffic monitoring infrastructure. The user-friendly JavaFX interface facilitates easy adoption by traffic authorities and law enforcement agencies, contributing to the system's operational feasibility.

3. Economic Feasibility:

Economic feasibility examines the financial aspects of implementing the system. The upfront costs associated with development, integration, and deployment are offset by potential long-term benefits. These benefits include improved traffic management, enhanced road safety leading to reduced accidents, and optimized traffic flow, contributing to overall economic efficiency. The economic feasibility is reinforced by the scalability of the Java platform, allowing cost-effective adaptations to varying traffic scenarios.

4. Scheduling Feasibility:

Scheduling feasibility involves assessing whether the project can be completed within a reasonable timeframe. The modular nature of the system development, where vehicle detection and speed detection functionalities can be developed concurrently, enhances scheduling feasibility. Java's object-oriented programming paradigm facilitates code modularity, enabling parallel development efforts and streamlined integration.

5. Legal and Regulatory Feasibility:

The system's adherence to legal and regulatory frameworks is crucial for its acceptance and success. Ensuring compliance with privacy laws, data protection regulations, and traffic management standards is imperative. The feasibility study includes an analysis of these legal aspects, with the system designed to operate within the bounds of relevant regulations.

6. Risk Analysis:

Comprehensive risk analysis identifies potential challenges and uncertainties associated with the project. Technical challenges, such as image quality variations and real-time processing constraints, are acknowledged and addressed through the use of robust algorithms and efficient coding practices. Ongoing advancements in computer vision technologies provide opportunities to mitigate risks associated with accuracy and reliability.

7. Scalability and Future Enhancements:

The feasibility study also considers the system's scalability and potential for future enhancements. Java's scalability, combined with the modular design of the system, allows for the seamless integration of additional features and technologies. Future enhancements may include machine learning algorithms for improved vehicle recognition, integration with emerging sensor technologies, and the incorporation of predictive analytics for more advanced traffic management.

Conclusion:

In conclusion, the feasibility study demonstrates that the proposed Vehicle Detection and Speed Detection System in Java is technically, operationally, economically, and legally feasible. The robustness of Java as a programming language, combined with advanced image processing tools, ensures the technical viability of the system. Operationally, the system aligns with current traffic monitoring infrastructure and meets the requirements of traffic management agencies. The economic feasibility is supported by long-term benefits, while the scheduling feasibility is enhanced by the modular and object-oriented nature of Java development. Legal and regulatory compliance, comprehensive risk analysis, and considerations for scalability and future enhancements further affirm the feasibility of the proposed system. The study provides a foundation for informed decision-making, paving the way for the development and implementation of an innovative solution to enhance traffic monitoring and road safety.

LITERATURE REVIEW

In the past few decades researchers had a great interest in vehicle detection and tracking. The topic attracted the attention quit much. Different sensing modalities have been used for detecting the objects or specifically vehicles.These modalities are LIDAR radar and computer vision.The attraction caused by immense progress of image processing. The very first signs and models of image processing goes back to the1960s'and1070s',after that various methods and techniques have been invented and proposed(Chen,2015).This chapter briefly discuss there centre related works by researchers regarding vehicle detection and tracking.

At the very beginning we can say that any approach in this topic are classified . At a glance we have four sections:

- Object recognition and identification from the appearance
- Classifying the object into one of the categories.
- Object detection or target detection
- Tracking the objector the target

Object detection presents some unique attributes of an object that a computer can identify distinctly from other objects.Mean while the object classification in tend to identify the similarities of an object with an object category at all. From the object detection result, we can assign an object tracker, the tracker is following the target by re-detecting it in the sequence frame following the first point of the target.

The primary goal and target of the thesis is to develop a system in which the system should be able to detect and track the vehicles automatically whether they are static or moving in images and videos.

Vehicle detection also called computer vision object recognition, basically the scientific methods and ways of how machines see rather than human eyes. The main duty of a vehicle detection system is to localize one or more vehicles in input images. There are two methods in vehicle detection system: sliding window method and local features method.In the local features-based method the system usually find the features of one object or the group of the objects in the very beginning. After that it tries to categories the founded features into different classes via classification models. This is usually the last step which the system decides which category is the object belongs to. It claims that the biggest strength of this method is that the geometry and features of the targets are known before the detection. While it can be a weakness or limitation as well because it can only detect pre-learned objects.The second method which is sliding window based system works differently.It scans the input image with a number of windows with different sizes. After that it analyzes whether the target is in the window or not.

A method proposed by Wangetal for detecting objects using edge detection in UAV footage as demonstrated in Figure 3.1. Researcher presents an example of how to use edge detection to recognize artificial objects. An edge detection algorithm have been used in this method to identify the straight lines on a vehicle. Before doing so, the algorithm removes noises from the background via and by the help of higher threshold in the process.

Another method suggested by Sokalsi uses color feature and edge detection to separate and distinguish artificial land natural objects.But the difference is in extracting then in features from the various channels of the color in the original image.Although these features are uses to specify the edges and changes to separate artificial and natural objects like illustrated in Figure3.2.

The two mention address searches makes the idea of summarizing clear.It summarizes local descriptors into texture features and color features. There are different cases.Color feature usually works fast in order to detect objects but in scenarios or somehow generally the accuracy is challenged. But texture feature works better with objects in images due to having more information about object in textures. Not to be forgotten both researches used unsupervised classification

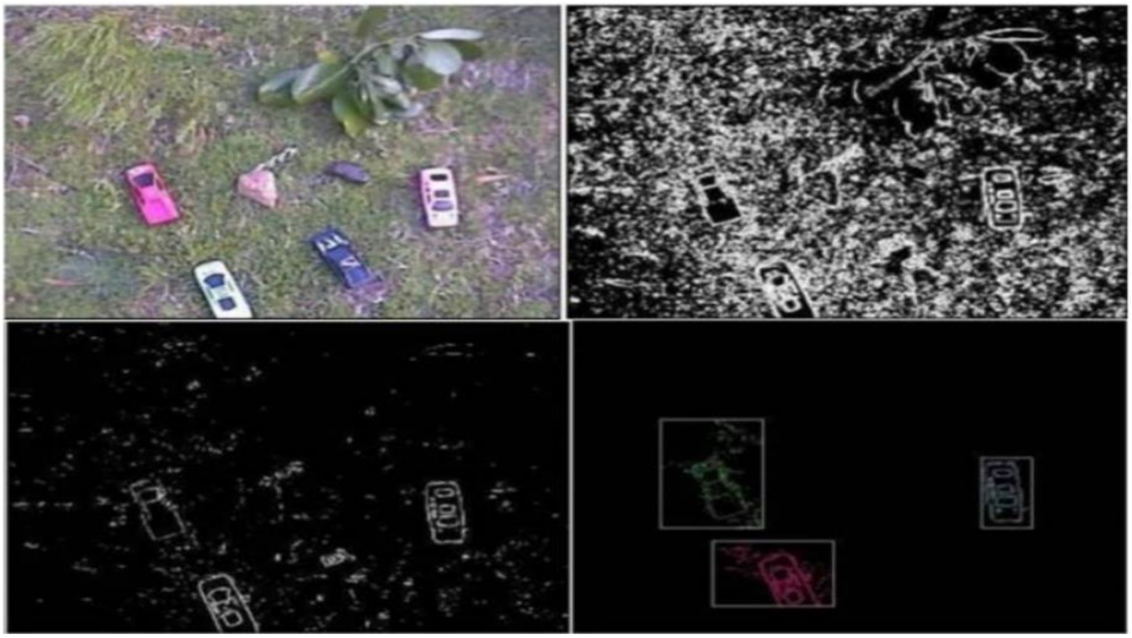


Figure3.1:different thresholds using edge features(Wangetal,2014)

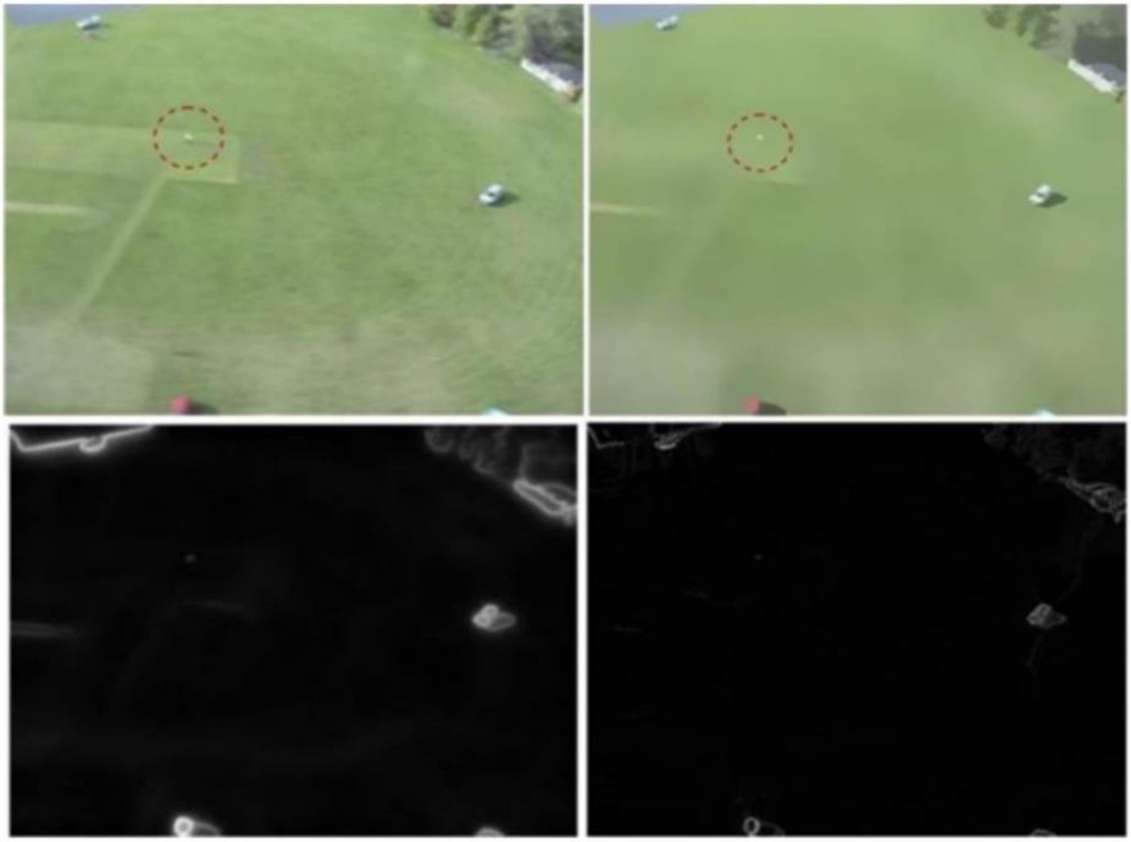


Figure3.2:edge and color feature for artificial objects(Sokalski etal,2010)

The two papers above use surface and shading highlights for item recognition, which is average for item identification.Utilizing shading highlights to distinguish objects is the most straight forward technique.N.Bahaetal(Baha,2014).display dexact milestone acknowledgment investigations of utilizing shading and surface high lights.When tourists pots are perceived from a class of geo-referenced pictures, they can be utilized to gauge the UAV's position and help self-sufficient route. This paper utilizes one lot of ethereal geo-referenced pictures and another arrangement of aeronautical pictures of the equivalent area, gathered at an alternate time, which are not geo-referenced. This work exhibits a milestone acknowledgment framework dependent on the extraction of shading and surface highlights. The proposed strategy utilizes these highlights to give data about surface introduction shape also shading.This methodology is being contemplated for application in a PITER (Real-Time Image Preparing) investigate venture that is being done at the Institute for Advanced Studies(IEAv–Institutode Estudos Avancados)and connected inself-ruling UAV route based on pictures.In this undertaking, 126 examples are utilized for administered realizing, which is a specific sort of AI calculation that permits the forecast of the class of a formerly obscure occurrence dependent on the learning of the class of a preparation test. A HSV shading what's more, Gray Level Co-event Matrix (GLCM) highlight is utilized to prepare the classifier the identification. The strategy utilizes a sliding-window approach in discovery. In this paper appears plainly that the mix of shading and surface highlights can improve the precision of discovery; the utilization of the Supervised Classification strategy can like wise improve location execution.

SamanGet.al (Kanistrasetal, 2015) proposed a vehicle identification approach by the thickness estimation. The angle vectors have been determined in the edge guide of the elevated pictures. It is recommended that, the headings of the inclination vectors are changing fundamentally in the limit of the objective and by ascertaining the standard deviation of the slope vectors. So by pre define the edge, the vehicles can be recognized inside the esteem. The assessment utilized the air borne pictures taken from street in Turkey and accomplished 86% exactness in F-measure.None the less, such discovery techniques without the preparation have a typical disadvantage,which the target objects are hard to recognize from the perplexing foundation circumstances.

Peng Wetal(Weietal,2015)proposed a vehicle recognition technique that improves foundation extractor. This methodology utilized asphalt division with 8-neighborhood filling to expelling the street markers so as to separating the unpredictable foundation. The outcome shows that the proposed strategy can stay away from the identification blunders brought about by the deceptive of the street markers. In any case, this identification strategy depends on fixed camera discovery, which is in opposition to the motivation behind this theory, however the foundation extractor system can be embraced.

J.Susaki et al.(Susaki,2015).proposed a two-arrange way to deal with the programmed location of vehicles inside aeronautical symbolism. Their methodology depends on the utilization of different fell Haar classifiers (Viola et al, 2005) for vehicle arrangement and an auxiliary check organize that end eavors to dispose of non-vehicle hopefuls dependent on UAV elevation and vehicle measure limitations. The fell Haar classifier gives a solid locator that is invariant to vehicle shading, type also, setup. To accomplish vehicle introduction invariance, they utilize four separate fell Haar classifiers prepared intest vehicle pictures arranged into four positional introductions. The four classifiers are then assessed utilizing a question picture at various scales and position sutilizing a sliding window approach and numerous classifier location and identification covers are settled utilizing a spatial combining strategy (Figure 3.3). J. Berniet al.(Berni et al, 2009) later broadened this work with the utilization of extra warm symbolism for warm mark affirmation which improved execution extensively.

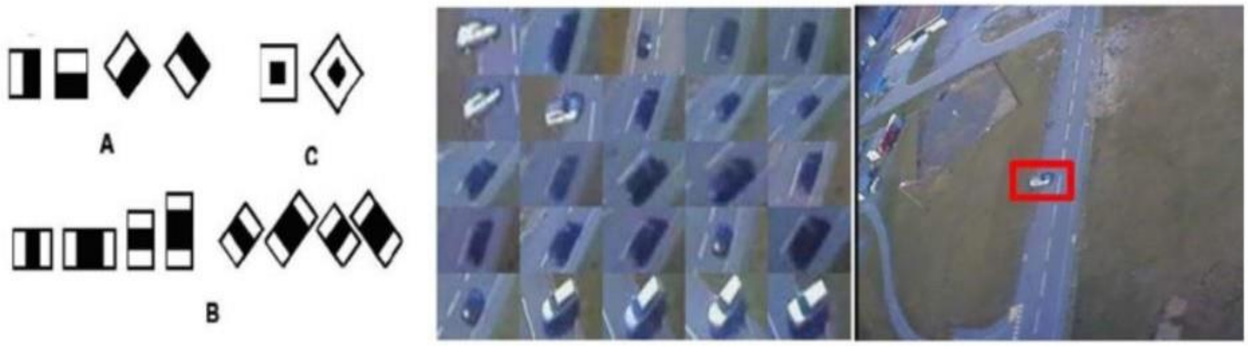


Figure3.3:features training samples and detection results (Susaki,2015)

Gleasonetal.(Chen et al, 2015) concentrated on programmed vehicle location in provincial conditions.Their approach comprises of a course location calculation, which is included two phases.In the first stage a Harris corner indicator is utilized to distinguish high lights of enthusiasm for the pictures the creators contend that "vehicles specifically have an expansive number of edges and corners looked at to characteristic articles". Next, a productive sliding window approach his utilized to decide districts with an element thickness higher than a fore or dained edge. Covering areas are gathered and further refined dependent on the shading qualities of foundation zones. The areas chose in this stage are then put through picture grouping procedures in the second stage which decide then earness of a vehicle.These creators exploration analyzed the execution of two picture fix descriptors an altered Histogram of Oriented Angles (HoG) highlight and Histogram of Gabor Coefficients highlights. They to researched the execution of three factual grouping strategies K-Nearest Neighbors(k-NN)Random Forests(RF) and Support Vector Machines(SVM).From the first phase of their calculation, they acquired a normal location rate of 85% and found the top performing classifier to be Random Forest sutilizing Histogram of Gabor Coefficients highlights this was equipped for characterizing 98.9% of vehicles and 61.9% of foundation pictures effectively

Sebastien Retal(Razakarivony,2016) proposed a vehicle location technique in unconstrained conditions with cutting edge object recognition approaches, which is utilizing the sliding window characterization strategy with the SVM classifiers.

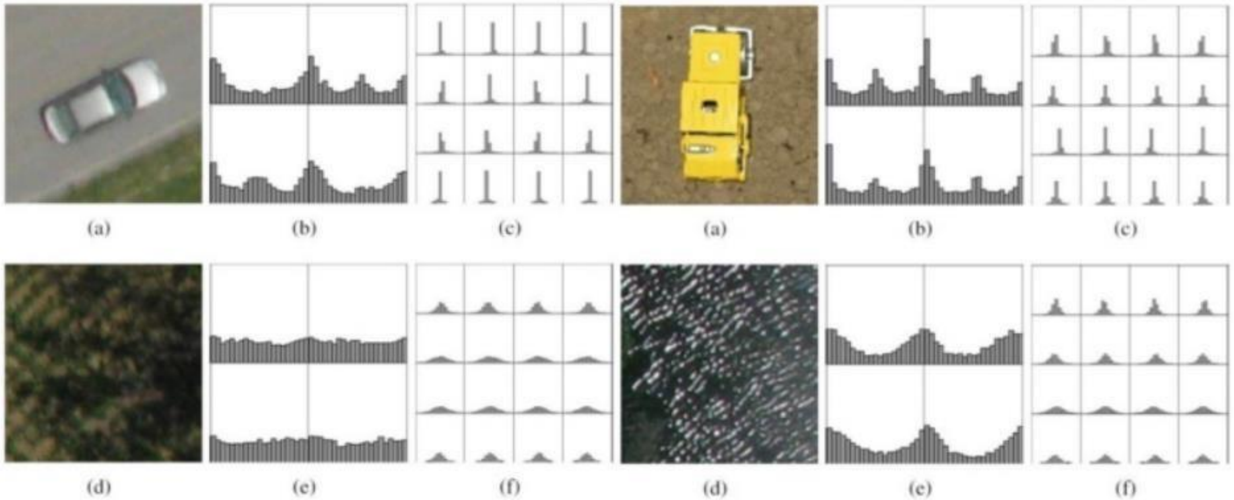


Figure3.4: HoG features,vectors,histograms,back ground image (Chenetal,2015)

Sahliet al.(Sahli et al,2010) proposed a nearby component based methodology for programmed vehicle identification in low-goals elevated symbolism. Their methodology was created with the point of being free from the imperatives identified with location strategies dependent on a vehicle's visual appearance, for example a vehicle's rectangular shape and the nearness of frontal and additionally back windows. Their approach depends on the extraction of Scale-Invariant Feature Transform (SIFT) highlights from vehicle and foundation pictures. These highlights are utilized to prepare a Support Vector Machine (SVM) classifier to characterize a model that can be utilized to order SIFT highlights separated from the auto sand foundation in a question picture. The accumulation of SIFT highlights that are anticipated to have a place with a vehicle are then grouped in the 2D picture space into subsets related with individual vehicles. The creators' bunching strategy depends on an altered Liking Propagation (AP) calculation that is bound by the spatial limitations identified with the geometry of vehicles at the given goals. They got a grouping exactness of 95.2% in aeronautical symbolism of a parking garage containing 105vehicles with no false-positive recognitions.

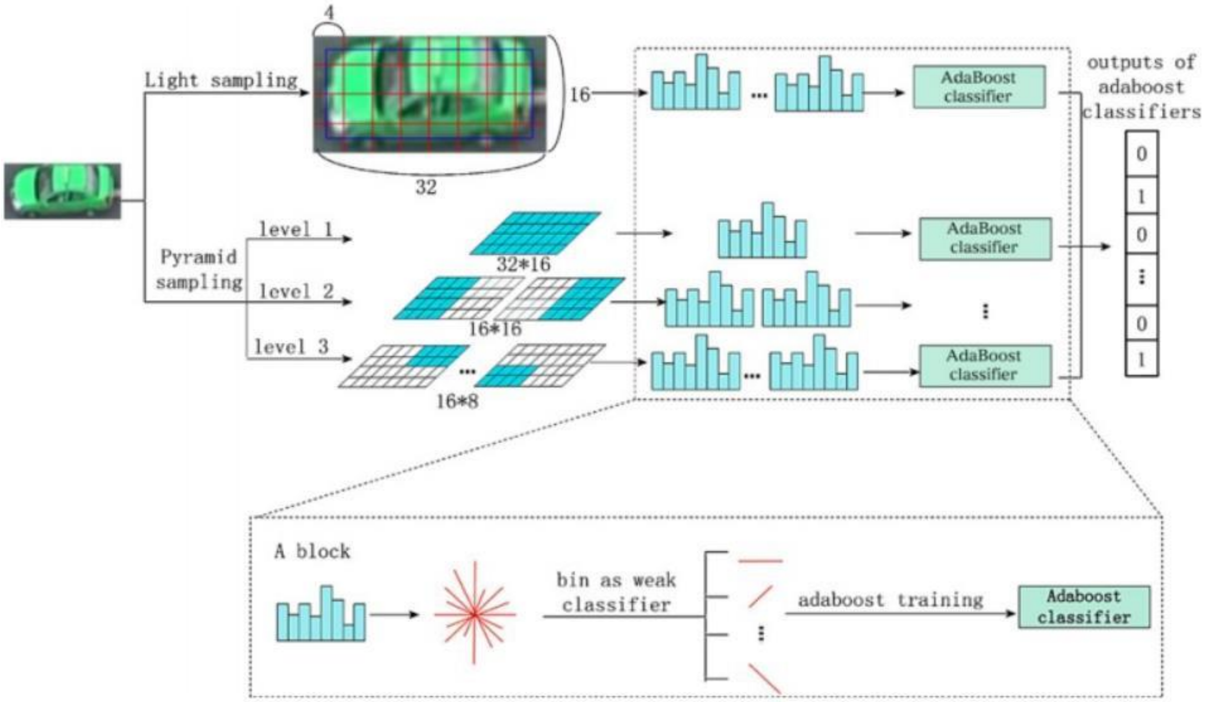


Figure3.5:extraction process work flow of the bLPS-HoG features(Xuetal,2011)

SYSTEM DESIGN

Architecture Diagram:

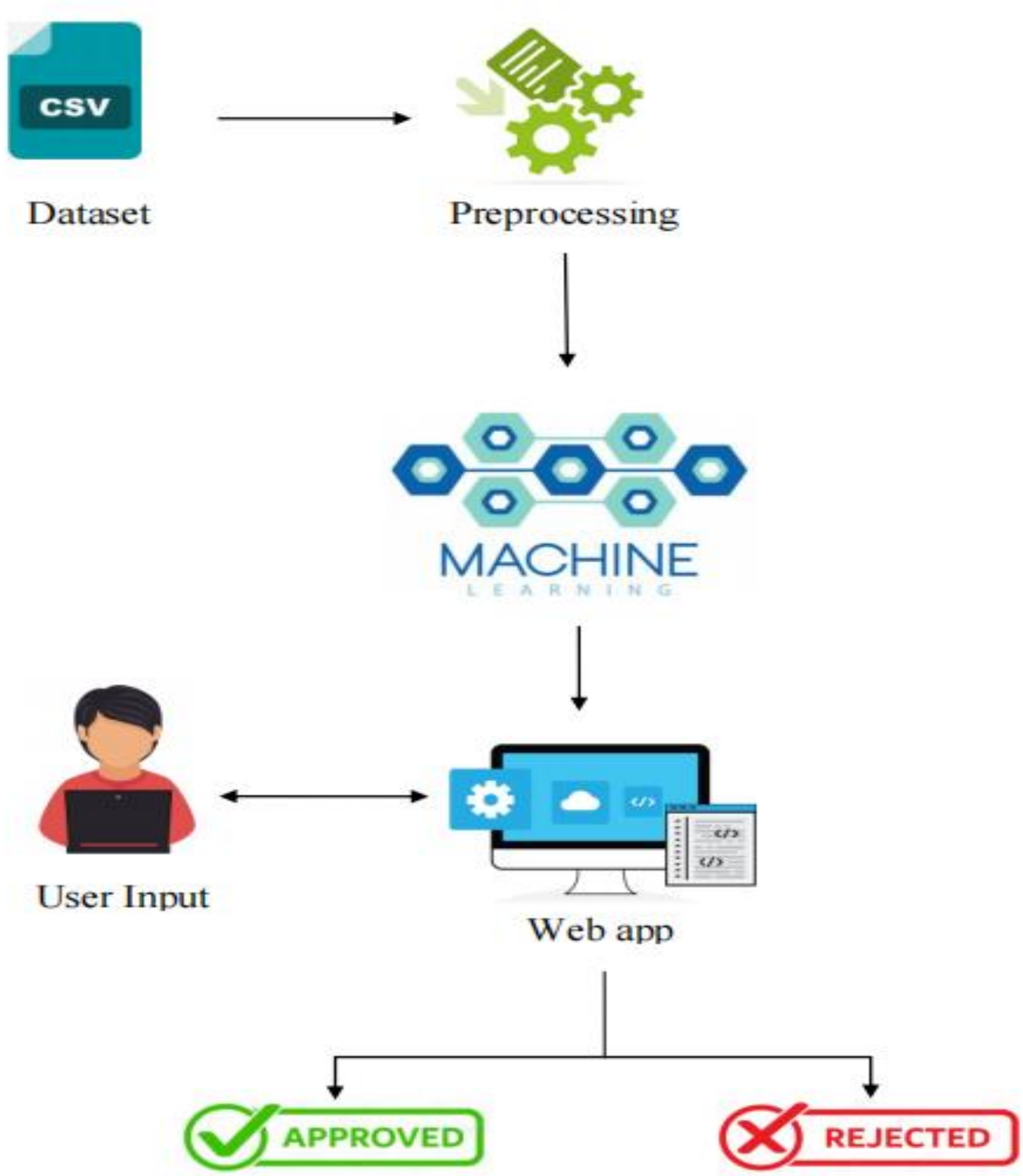
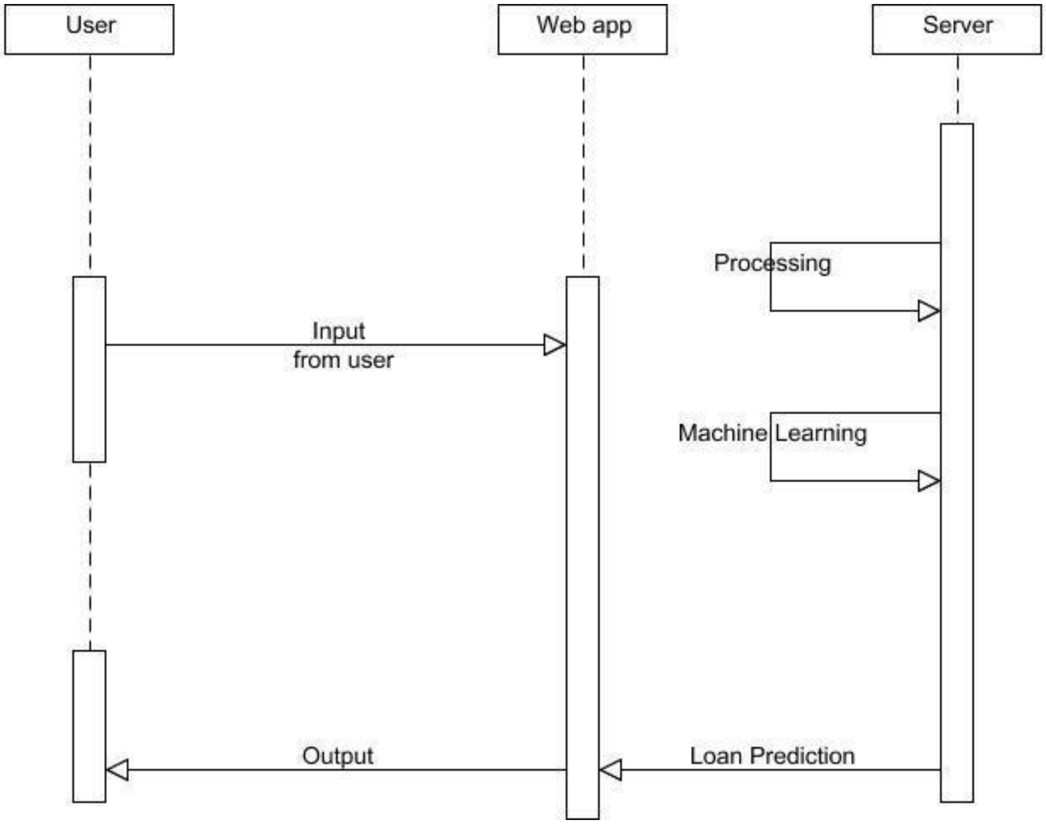


Fig: 5.1

Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order.It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group.UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize,modify,construct and document the art if acts of an object oriented software intensive system under development.

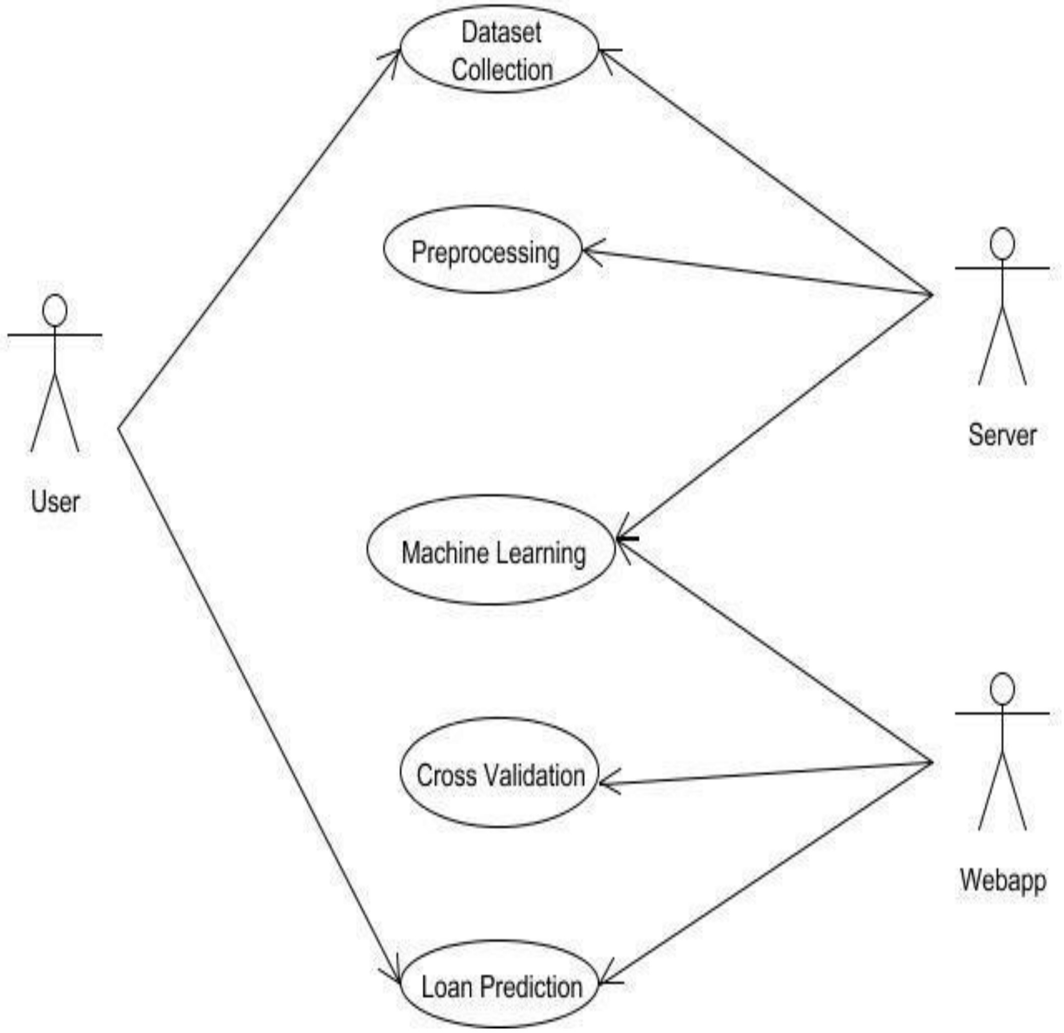
USE CASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

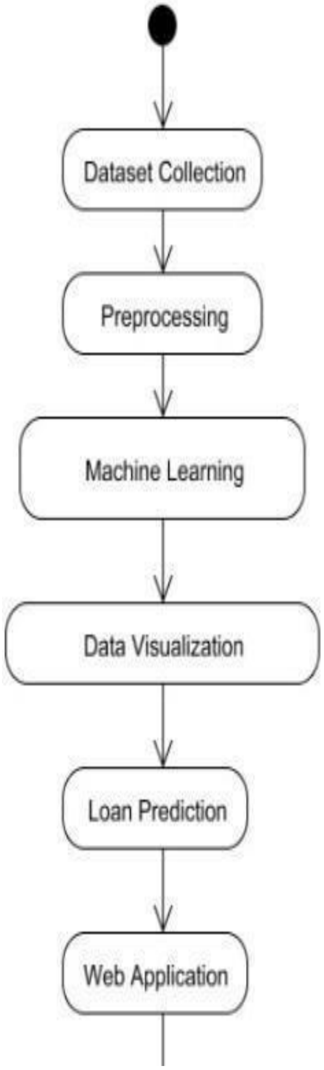


Activity Diagram:

Activity diagram is a graphical representation of work flows of step wise activities and actions with support for choice, iteration and concurrency.An activity diagram shows the overall flow of control.

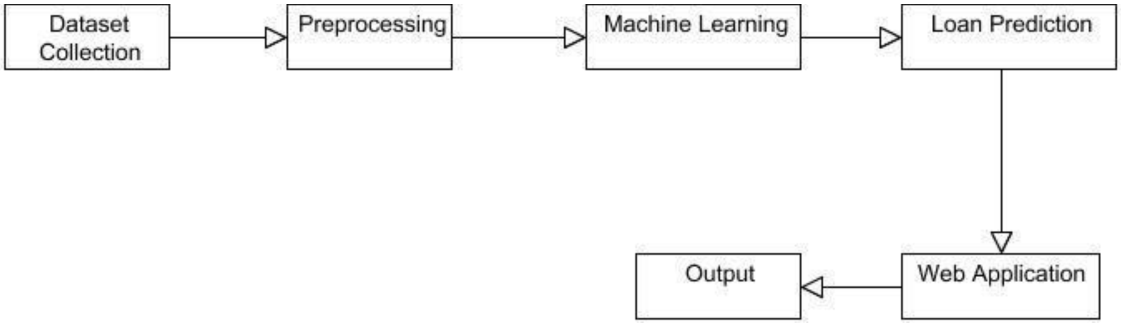
The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the work flow.



Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.


Dataset Collection
Preprocessing
Machine Learning
Loan Prediction
Web Application
Output

MACHINE LEARNING TECHNIQUES

Machine Learning

The digital revolution has expedited new issues confronted human beings.Quick tech development human interaction with electric devices and various technologies electronic records and notable development of people in the web set up hands together to create a gigantic measure of data each second. From the previous two decades associations, colleges, scientists and academicians are attempting to grow new patterns and innovations to use these information for various purposes for example for detections recognitions investigation distinguishing pieces of proof recomendations and so on. Practically all businesses these days use AI for the improvement and precision of their working procedures and methods like it is being utilized in Medical, engineering, finance, manufacturing and so forth one of these patterns is Machine learning in computer technology what's more, man-made consciousness which is considered and focused by tech specialists enormously.

The term Machine Learning (ML) is right off the bat utilized by Arthur Samuel in 1959 and being created and finished with gigantic different researchers till now. ML is a piece of Artificial Intelligence (AI) that uses statistic strategies to offer capacity to Computer applications and empower them to gain from data dynamically with out earlier directions and characterized programs (Koza et al, 1996).ML is doing that it makes the concealed data known by assessing and perceiving examples and relations between data and occasions. For these purposes ML utilizes computer algorithms, for the most part, models are worked to gain from the data also make predictions on a similar sort of information which is prepared by. Traditional applications are working in constrained and limited guidelines which are characterized by software engineers. Advancement and self-learning properties of algorithms influence them to conquer conventional applications since building information driven applications like computer visions or email sifting and so forth are nearly infeasible with traditional programming strategies. These algorithms help us to take better decisions and bring dependability.ML has 3 fundamental sorts they are as per the following:

- Supervised
- Unsupervised

- Reinforcement

Supervised Learning

In most of the cases, supervised learning is utilized for even minded AI problems. There are two factors in supervised learning, one for inputs of info and one for outputs.

The algorithms are attempting to master mapping between these factors through a mapping work(Russell and Norvig, 2016). The algorithms are prepared with information sources and thought about the result to accessible outputs. In the training stage as a coach or educator, we screen the learning.

At the point when the algorithms anticipate the output, it will check whether the appropriate response is correct or wrong or near output or a long way from the output. These procedures help the algorithms to learn and improve their performance. The learning procedure winds up when It comes to a worthy accuracy level. At the point when the new data comes, it endeavors to fore see the output dependent on the past learnings and estimated mapping among information inputs and outputs.

Supervised learning algorithms are gathered into regression and classification which are examined in the last section. There are a few concerns exist that these algorithms are as it were operational when the data is labeled. Since the using data as another oil on the planet, data isn't free any longer and social events information for learning may be costly. For the common sense part a ton of supervised algorithms have been tested. Every one of them had their own qualities and shortcomings. As a base idea in ML, there is no such algorithm that works best for all sort of tasks. Accordingly picking the algorithm is a significant topic that all ought to think about it while working in ML.probably the most utilized managed algorithms are as per the following:

- Support Vector Machines
- Naïve Bayes
- Logistic regression
- Linear regression
- Linear discriminant analysis
- Decision trees
- k-nearest neighbor

- Neural Networks

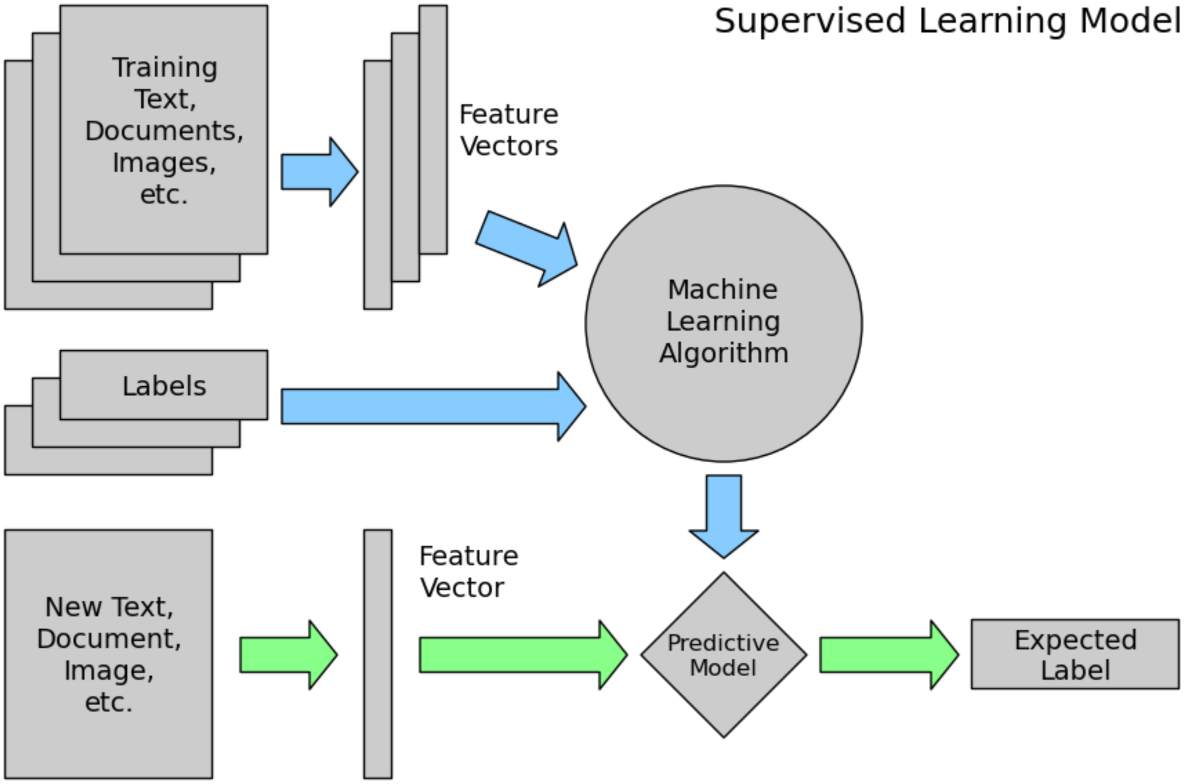


Figure5.1: Diagram of the Supervised Learning

Unsupervised Learning

Unsupervised algorithms not at all like supervised don't have right answers. As it were, there is no output variable and a guide or instructor to address botches. The algorithms are attempting to comprehend the information features. They search for covered up and concealed examples in the dataset to predict the output by simply having the input factors. There are no names for them to use so as to learn and improve their predictive capacity (OFOR, 2018). Unsupervised learnings are gathered in clustering and association problems.

Clustering: in this sort of tasks the information is partitioned into gatherings, for example grouping clients by their purchasing behaviors.

Association: Algorithms are attempting to compreh end the standards that can elucidate the expansive bit of the data, for example, client purchases shirt will in general purchase pants as well.

To give some examples of unsupervised learning calculations:

- K-means clustering
- Apriori algorithm association

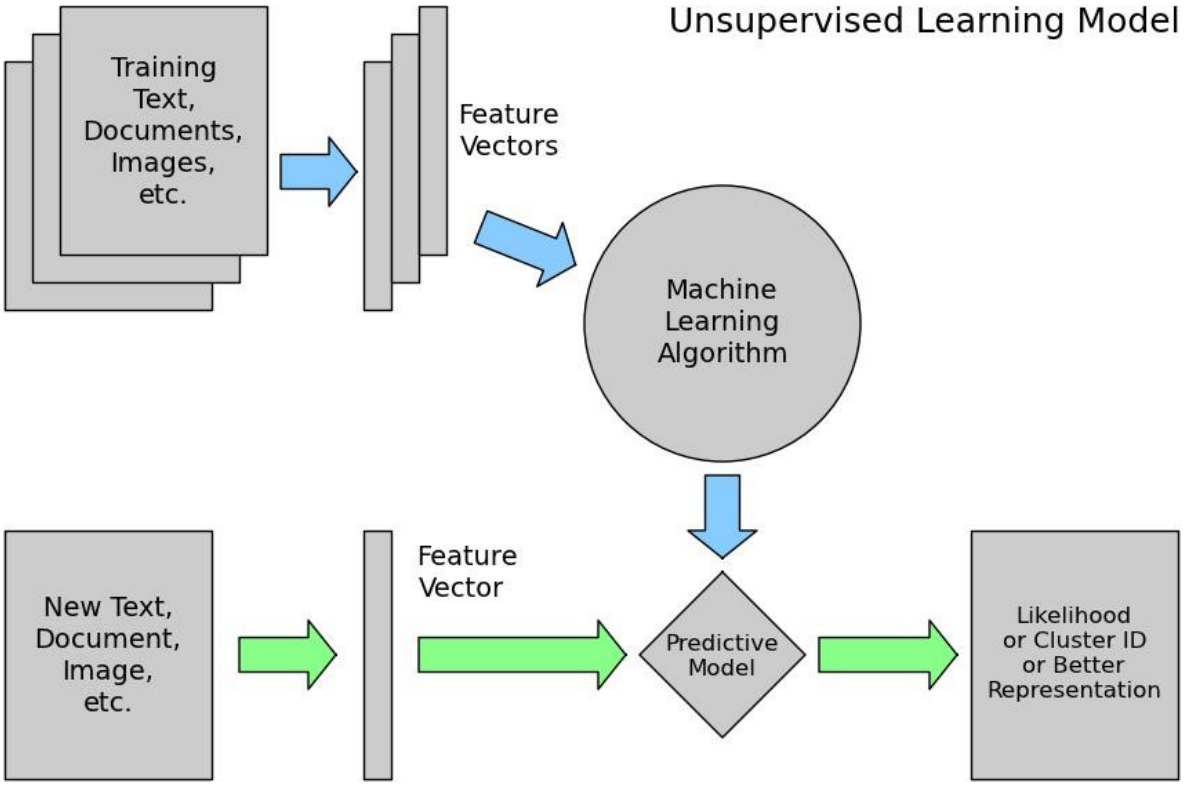


Figure5.2: Unsupervised learning model diagram

Reinforcement Machine Learning

The reinforcement learning fundamentally works like a child learning in the beginning period of his/her life. At the point when a kid is working superbly the individual will be induced and when a kid is completing a terrible job, the outcome is by one way or another disciplines or advising for not rehashing that. These algorithms are doing likewise task by an agent fills in as a child here.The agent co operates with enviroment it gets remunerate for performing undertakings correctly and punishments for performing it badly. When agent endeavoring to amplify the prizes and limit the punishments.

Think about a self-driving vehicle, if the vehicle touched base in its goal with no accidents, going out of the street or terrible stops it will get rewards yet on the off chance that it did any of the referenced undertakings it will get punishments. In this way next time the vehicle won't do the accidents that it took punishments for.These algorithms are additionally called dynamic programming and immense measure of studies and Researches are being led to improve these algorithms they will be attainable for a great deal of tasks in the close future.

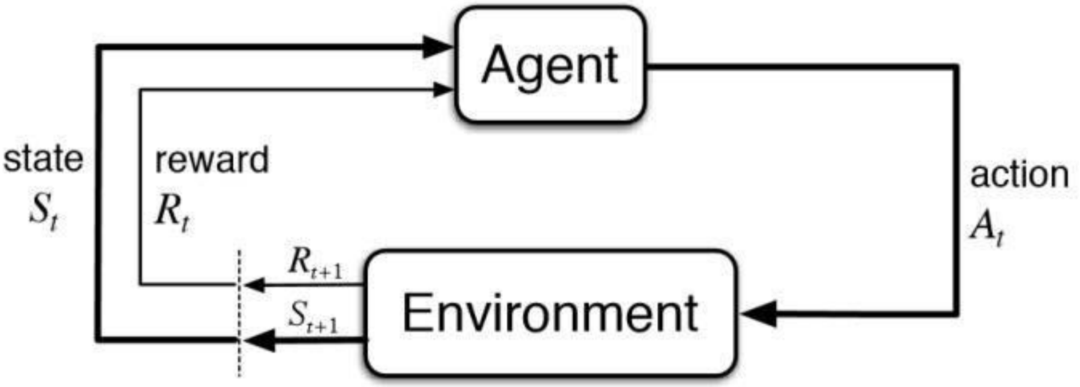


Figure5.3: Reinforcement Learning model(UCBWiki,2016)

Used Machine Learning Techniques

In this these is the author implemented and developed two algorithms from supervised category on the same dataset which is described earlier.These algorithms areas follows:

- Support Vector Machine (SVM)
- Decision Trees

Support Vector Machine(SVM)

This is a supervised machine learning algorithm that can be utilized for classification and regression tasks. The general strategy in SVM is to discover a hyper plane which is splitting the dataset into two classes (KDnuggets,2016) like presented in the Figure3.6.

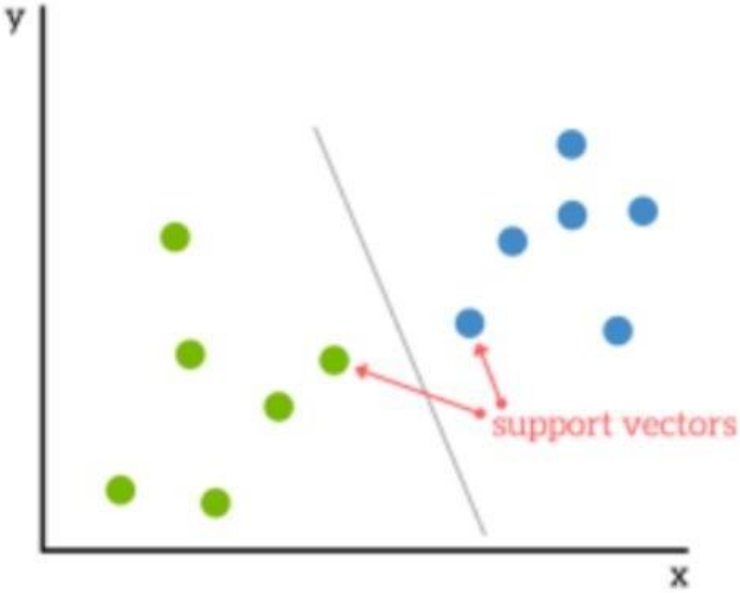


Figure5.4:SVM splittion

These are the data points which are basically close to hyper plane for instance if the point is going to be removed from the dataset it will change many things including role of the hyper plane splitter.This is one of the important factors in datasets.

Decision Trees

These type of algorithms are most used supervised algorithms by researchers.It enables developer to build high accuracy models with having a good simplicity in compare to other ML models.The basic out come in decision tree is whether“YES”or “NO”.

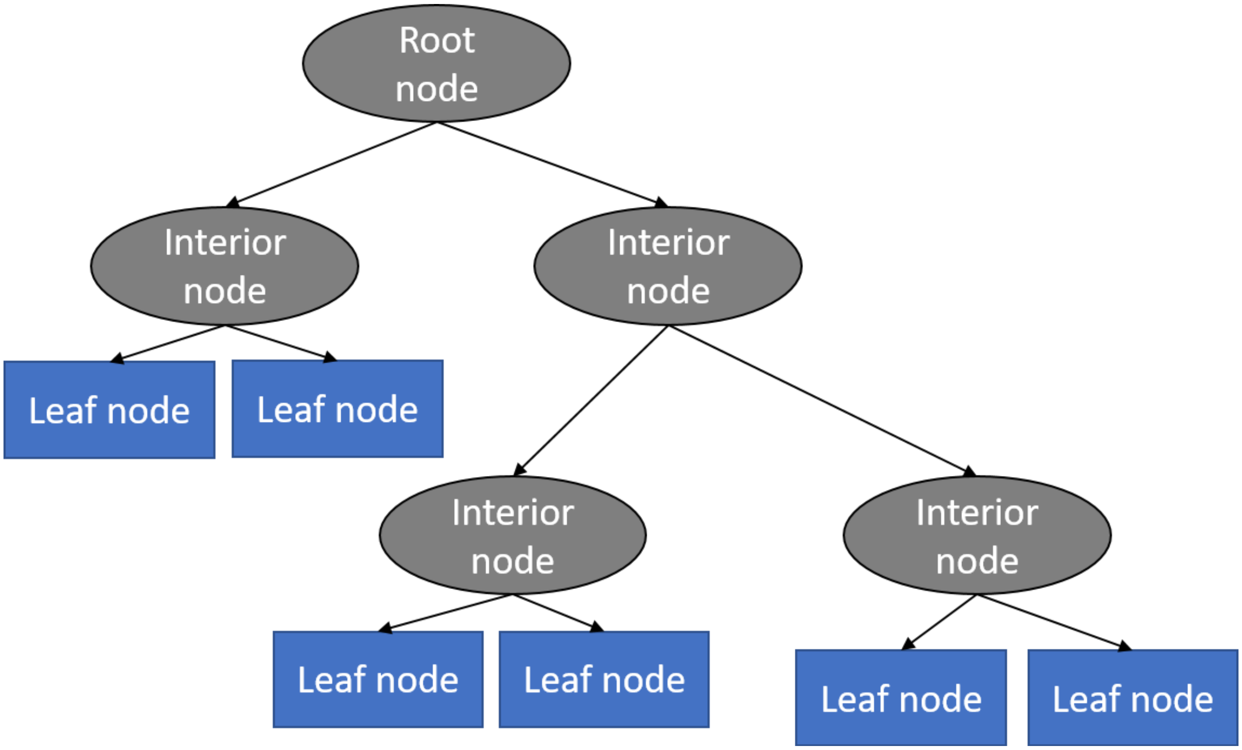


Figure5.5: Decision tree flow chart

These models are working likewise a tree, as showed in figure above. It basically breaks the whole dataset into smaller portions and the process continuous in the sub portions from the root.It goes to reach to a tree with leaf nodes and decision nodes. The second one is a type of tree which is named root node. The advantage is that it can process both types of data including categorical and numerical data.All types of 1nodes are presented inFigure3.5.

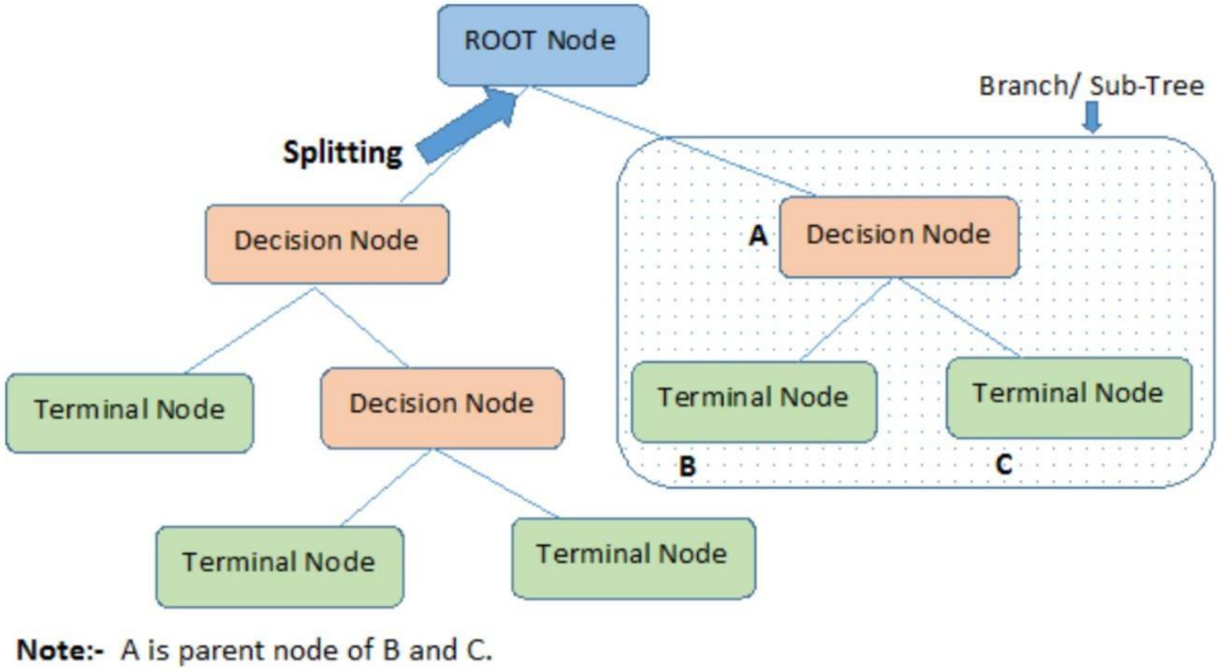


Figure5.6: Nodes representation

METHODOLOGY

In this chapter, all methods and tools used to detect and track vehicles are being discussed. At the very beginning, tools that are utilized in the research is described in details that how they are being used in the context. After that, the implementation process started with data cleaning,pre processing and algorithms are presented with a short summary of these issues at the end of the chapter.

Tools Used

Like every other researches and studies this thesis utilized some applications and tools for creating models and experiments. For such researches many tools needed to be used, for example the author used python as the programming language of the model development or the data set with a lot of vehicle and non-vehicle images used for training or for example many python libraries which are necessary or we can say useful in order to create ML models. In this chapter all the tools that are being used in the thesis are presented.

Python

Python is a general purpose programming language which is created by Guido van Rossum, it is used for different platforms like mathematic, computer GUI, web and so many large scientific applications. The main aim creating python was to make programming easy so that everyone in the world is able to write code. Therefore it’s popular for its simplicity. Python is sensitive inspacing.Despite the fact that python was created for kids, for the time being it bits all programming languages in many fields which is amazing and somehow unbelievable. Python is able to do whatever other programming languages can do. From web to algorithms to desktop applications etc. in the past few years’ python became popular in in artificial intelligence and machine learning applications due to having many efficient and handy libraries which makes the job much easier and faster. Experts coming to this field have various backgrounds. Since they don’t have programming background, the most easy and convenient language to start with is python.

Despite the above mentioned properties, the researcher’s knowledge and self-interest caused to choose python as the programming language of the model development. The libraries that are utilized in this thesis areas follows.

.Numpy

Numpy is an open source library, it does the computing with the help of multi-dimensional matrices and arrays. It contains a number of functions which makes it easy to work with these type of data. In data analysis if we want to make the speed fast and efficient we need to use arrays, therefore this library helps data scientists to work faster with large amounts of data.Usually for detection and forecasting, the models function needs arrays as parameter to operate fast and decrease the training prediction time.

Matplotlib

Plotting is growing in all fields to visualize the data and to make it understandable. Therefore Matplotlib is being used as a plotting library to create different kind of graphs and figures for variety of aims. The good thing about matplotlib is that it can produce good plots and graphs with just few lines of codes.Somatplotlib is used to extract color features and create histograms.

Jupyter Notebook

Jupyter Notebook is a web based application which makes us able to create and modify live codes equations, plaintexts and visualizations. This is an open source notebook which supports many programming languages. This is used for different purposes like machine learning,numerical simulation, information visualization etc. the researcher decided to use this note book for writing read able codes and implementing machine learning algorithms.

Computer

The PC that is being used to train and test the models has the following

properties :

Model:Dell inspiron 13-5378

RAM:8 GB

Hard disk:512 SSD

Processor:Core i7 Quad Core

Graphic:Intel HD4 GB

Datasets

Udacity website equips students with the great resources for training the classifiers. Vehicles and non-vehicles samples of the KITTI vision benchmark suite have been used for training as shown in figure4.1. The data set is downloaded from Udacity website.

These example images come from a combination of the GTI vehicle image database, the KITTI vision benchmark suite, and examples extracted from the project video itself. You are welcome and encouraged to take advantage of the recently released Udacity labeled dataset to augment your training data.

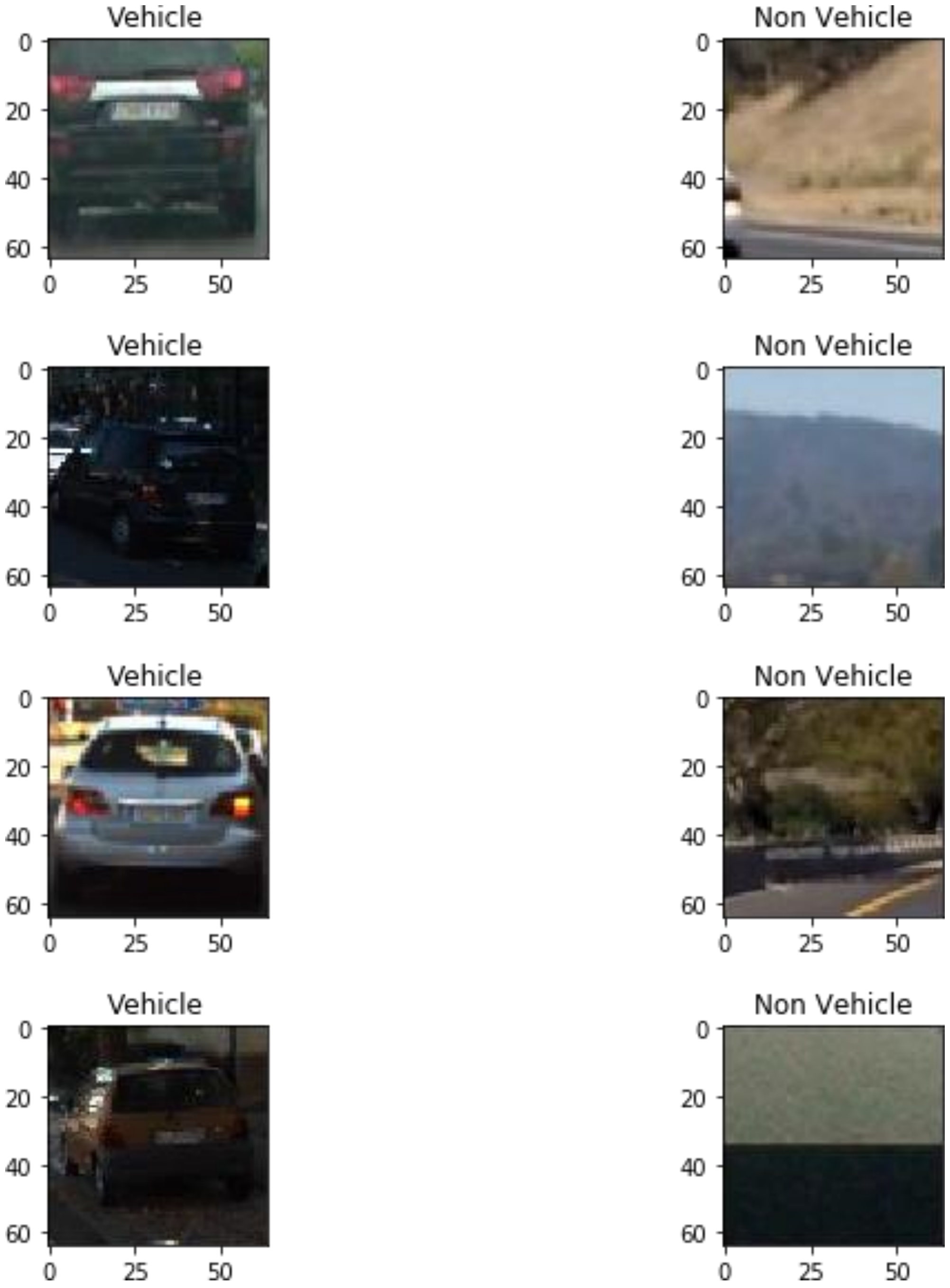


Figure6.1: Vehicle and Non-Vehicle images

Implementation

Vehicle detection and tracking is important in self-driving technologies to drive car safely.In this project, goal is to write a software pipeline to detect vehicles in a video.

It can be achieved by following the below tasks:

- Perform a Histogram of Oriented Gradients (HOG)feature extraction on a labeled training set of images and train a classifier Linear SVM classifier.
- Implement a sliding-window technique and use your trained classifier to search for vehicles in images.
- Run pipe line on a video stream and create a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles.
- Estimate a bounding box for vehicles detected.

Vehicle and non-vehicle images as Numpy array are loaded to the separate list using function.There are various feature extraction techniques has been used to train the classifier to detect the cars efficiently.

While it could be cumbersome to include three color channels of a full resolution image, you can perform spatial binning on an image and still retain enough information to help in finding vehicles.

As you can see in the code in appendix, even going all the way down to 32 x 32 pixel resolution,the car itself is still clearly identifiable by eye, and this means that the relevant features are still preserved at this resolution.

A convenient function for scaling down the resolution of an image is OpenCV’s cv2.resize(). If you then wanted to convert this to a one dimensional feature vector,numpy’sravel() function can be used.

Color histogram

In photography a histogram is simply a graphical representation of the number of pixels in the image that fall within a certian range, either luminance or color. For example for a normal luminance histogram the graph shows the number of pixels for each luminance or brightness level from black to white. The higher the peak on the graph the more pixels are at that luminance level. With a color histogram the principle is the same but instead of seeing the levels of black graphed you will now see the number of pixels for each of the three main colors.

A color histogram is a simply a histogram that shows the color level for each individual RGB color channel.

If we had to, we could differentiate the two images based on the differences in histograms alone.As expected the image of the red car has a greater intensity of total bin values in the R Histogram1 (Red Channel) compared to the blue car's R Histogram 2. In contrast the blue car has a greater intensity of total bin values in B Histogram 2 (Blue Channel) than the red car's B Histogram features. Differentiating images by the intensity and range of color they contain can be helpful for looking at car vs non-car images.

Histogram of Oriented Gradients(HOG)

A feature descriptor is are representation of an image or an image patch that simplifies the image by extracting useful information and throwing away extraneous information.The technique counts occurrences of gradient orientation in localized portions of an image.In the HOG feature descriptor, the distribution (histograms) of directions of gradients (oriented gradients) are used as features. Gradients (x and y derivatives) of an image are useful because the magnitude of gradients is large around edges and corners (regions of abrupt intensity changes) and we know that edges and corners pack in a lot more information about object shape than flat regions. Next one is to choose the right parameters to train the classifier to predict the image,I have defined he parameter class to define these parameters.

Classifiers

There are two classifiers that have been used in this thesis.

- Support vector machines (SVMs)
- Decision Tree

Support vector machines (SVMs) are a set of supervised learning methods used for classification regression and outlier's detection.I have decided to use Linear SVC as classifier in this project.

Decision Tree is also a supervised machine learning algorithm which is mostly used to classify the data. The details about classifiers are presented in chapter3.

Train and Test Split

The Standard Scaler() function assumes data is normally distributed within each feature and will scale them such that the distribution is now centred around 0, with a standard deviation of 1. ‘x’values are transformed using the function and get the output scaled_X.

There are few helping libraries to split the dataset. ‘train_test_split’ funtion from ‘sklearn’ is one of them which help to split the dataset into train and test data for the classifier.

Sliding Window

In the context of computer vision (and as the name suggests), a sliding window is rectangular region of fixed width and height that “slides” across an image. For each of these windows, we would normally take the window region and apply an image classifier to determine if the window has an object that interests us.

Here are three test images and we can see all the bounding boxes for where my classifier reported positive detections. You can see that overlapping detections exist for each of the two vehicles, and in two of the frames, there is a false positive detection on the middle of the road. In this exercise, you’ll build a heat-map from these detections in order to combine over lapping detections and remove false positives.

In order to combine overlapping detections and remove false positives, heat map and there a hold limit are used.

The hog sub-sampling is more efficient method for doing the sliding window approach. The code only has to extract hog features once and then can be sub-sampled to get all of its overlaying windows .Each window is defined by a scaling factor where a scale of 1 would result in a window that’s 8 x 8 cells then the overlap of each window is in terms of the cell distance. This means that a cells_per_step = 2 would result in a search window overlap of 75%. Its possible to run this same function multiple times for different scale values to generate multiple-scaled search windows. The hog sub-sampling helps to reduce calculation time for finding HOG features and thus provided higher throughput rate.

I have decided to choose stating position of the window search from 350 px to 656 px and cells_per_step reduced to one to get more accurate result.

As explained above, same heatmap and threshold with limit 1 technique is used to combine overlapping detections and remove false positives.

Pipeline video

Finally create the pipeline vide by processing the each frame of the image with above techniques and create the video out of the processed frames.

find_cars_hog_sub function extracts all the bounding boxes detected for the cars in the image.heat_threshold function is used to combine overlapping detections and remove false positives and produce the output with bounding box added to the image.

Model Development Summary

The entire process of vehicle detection and tracking is described in figure 6.2. Each step needed few actions and tasks to be completed. The flow diagram clearly describes the entire process.

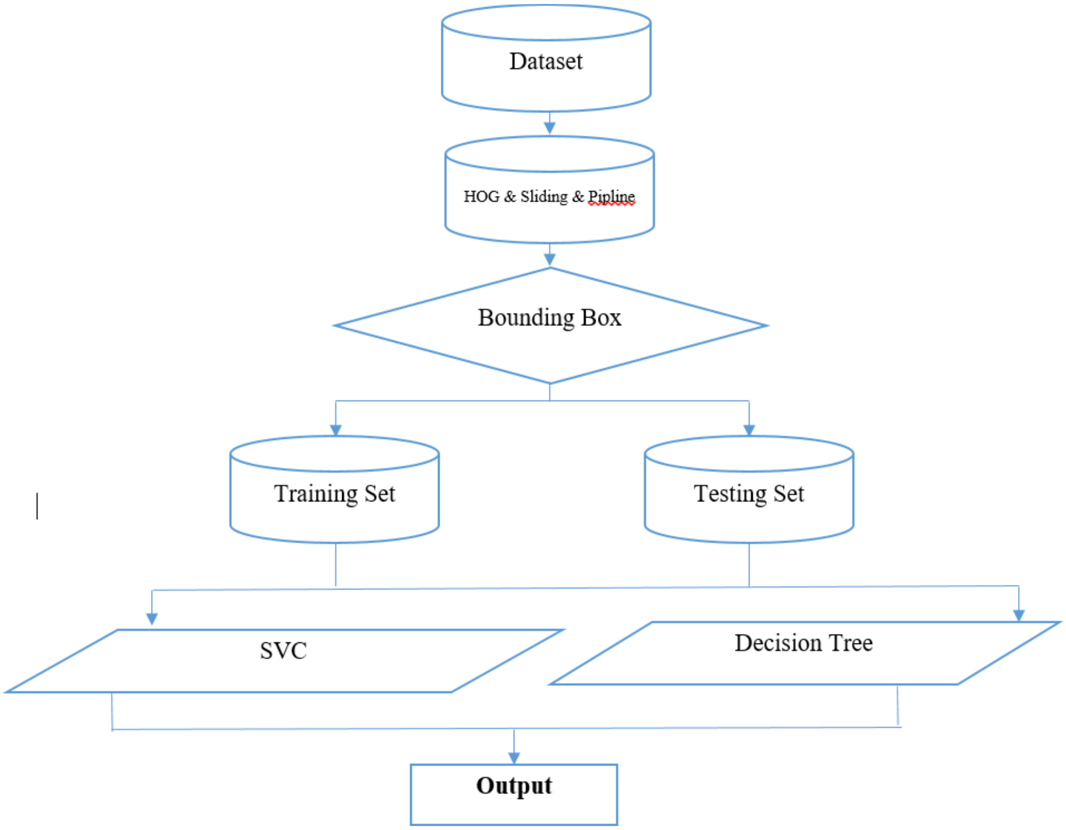


Figure6.2: Model development summary

CODING AND TESTING

CODING

Once the design aspect of the system is finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken when ever changes are required it easily screwed into the system.

CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program.They make the code Easier to read, understand and maintain. This phase of the system actually implements the blue print developed during the design phase.The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format Exception and

error handling

NAMING CONVENTIONS

Naming conventions of classes, data member, member functions,procedures etc., should be self-descriptive. One should even get the meaning and scope of the variable by its name. The conventions are adopted for easy under standing of the intended message by the user.So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lower case letter with each sub sequent letters of the new words in uppercase and the rest of letters in lower case.

VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time.

This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which in dentation is most important.Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

MESSAGE BOX FORMAT:

When something has to be prompted to the user, he must be able to under stand it properly.To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X –User has performed illegal operation.
- !–Information to the user.

TEST PROCEDURE

SYSTEM TESTING

Testing is performed to identify errors.It is used for quality assurance.Testing is an integral part of the entire development and maintenance process.The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself.For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walk through.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

TEST DATA AND OUTPUT

UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well asboundary values and special values, such as logically related inputs, files of identical elements,andempty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

STRESS TEST

Stress Test is those test designed to intentionally break the unit.A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths.The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition,I/Oerrors,buffer problems and textual errors in output information

INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to un cover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product.The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing,and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links.Errors are localized to the new module and its intercommunications.

The product development can be staged, and modules integrated in as they complete unit testing.Testing is completed when the last module is integrated and tested.

TESTING TECHNIQUES/TESTING STRATERGIES

TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet–undiscovered error.A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation,which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences.It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed.Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the

Review of specification design and coding.Testing is the process of executing the program with the intent of finding the error.A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error.Any engineering product can be tested in one of the two ways:

WHITE BOX TESTING

This testing is also called as Glass box testing.In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases.Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised.It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically.For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works“outward”toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules,which are highly prone to interface errors, should not be assumed to work instantly when we put them together.The problem of course,is“putting them together”-interfacing. There may be the chances of data lost across on another’s sub functions,when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels;global data structures can present problems.

PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written.An improperly defined field dimension or omitted keywords are common syntax error.These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler swill not deduct logical error, the programmer must examine the output.Condition testing exercises the logical conditions contained in a module.The possible types of elements in a condition include a Boolean operator,Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

SECURITY TESTING:

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins.Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted,one of two conditions exists.

* The function or performance characteristics confirm to

specifications and are accepted.

* A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system.The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required.This is done in regarding to the following points.

- Input screen design.
- Output screen design.

SYSTEM CODING

PSEUDOCODE

```
// Import necessary libraries
import java.util.List;
import java.util.Array List;
import java.util.concurrent.Executor Service;
import java.util.concurrent.Executors;

public class Vehicle Speed Detection System {
   // Define constants
   private static final int NUM_THREADS = 4;

   // Initialize data structures for vehicle tracking
   List<Vehicle> detected Vehicles;
   Executor Service thread Pool;

   // Initialize data structures for speed estimation
   List<Frame> frames;
   List<Vehicle Speed> vehicle Speeds;

   // Constructor
   public Vehicle Speed Detection System() {
      detected Vehicles = new Array List<>();
      frames = new Array List<>();
      vehicle Speeds = new Array List<>();
      thread Pool = Executors.new Fixed Thread Pool(NUM_THREADS);
   }

   // Main method to process incoming frames
   public void process Frame(Frame frame) {
      frames.add(frame);
```

```
        detect Vehicles(frame);
        estimate Speeds();
    }

    // Method to detect vehicles in a frame using computer vision techniques
    private void detect Vehicles(Frame frame) {
        List<Vehicle> vehicles In Frame = Computer Vision .detect Vehicles(frame);

        // Update the list of detected vehicles
        Detected Vehicles.add All(vehicles In Frame);
    }

    // Method to estimate speeds of detected vehicles in consecutive frames
    private void estimate Speeds() {
        if (frames.size() > 1) {
            // Create tasks for parallel speed estimation
            List<Speed Estimation Task> tasks = new ArrayList<>();
            for (int i = 1; i < frames.size(); i++) {
                Frame previous Frame = frames.get(i - 1);
                Frame current Frame = frames.get(i);
                List<Vehicle> vehicles In Previous Frame = get Vehicles In Frame(previous Frame);
                tasks.add(new Speed Estimation Task(vehicles In Previous Frame, current Frame,
vehicle Speeds));
            }

            // Submit tasks to the thread pool for parallel execution
            Thread Pool.invoke All(tasks);
        }
    }

    // Method to get vehicles present in a specific frame
    private List<Vehicle> get Vehicles In Frame(Frame frame) {
```

```
        // Implement logic to filter vehicles based on frame information
        // ...

        return filtered Vehicles;
    }

    // Task for parallel speed estimation
    private static class Speed Estimation Task implements Run able {
        private List<Vehicle> vehicles In Previous Frame;
        private Frame current Frame;
        private List<Vehicle Speed> vehicle Speeds;

        public Speed Estimation Task(List<Vehicle> vehicles In Previous Frame, Frame current
Frame, List<Vehicle Speed> vehicle Speeds) {
            this.vehicles In Previous Frame = vehicles In Previous Frame;
            this.current Frame = current Frame;
            this.vehicle Speeds = vehicle Speeds;
        }

        @Override
        public void run() {
            // Implement logic to estimate speeds of vehicles between consecutive frames
            // Update the list of vehicle speeds
            // ...
        }
    }

    // Additional methods for user interface, system initialization, etc.
    // ...
}
```

RESULTS AND DISCUSSION

Experimental Setup

While developing Machine learning algorithms there are many options to choose for instance MATLAB, Python, and R programming language. Each option has its own advantages and privileges. Because of that the researcher has decided to choose python for the development due to the easiness and rich libraries available for all kind of tasks. The tools and techniques which have been used in this thesis is discussed in details in chapter4. Many libraries have been utilized to perform various tasks like Numpy is used for importing and processing data,matplotlib is used for visualization of the extracted color features, Scikit-Learn for splitting data into train and test parts, classifier functions for creating the models and training them with available data in the datasets. In this chapter we are going to discuss the creation of the models along with focusing on the results. Like mentioned in previous chapters, there are two algorithms in the thesis which are used to classify the images and a pipeline to track images in the videos.The tow classifiers are SVM and Decision Tree. At the end of the chapter the result has been described in a comparative way which the best model or classifier is suggest among the two.

Support Vector Machine (SVM)Classification

SVMs algorithm are used mostly for classification tasks. These models are working based on discovering a hyper plane concept which actually perfectly divides the data into two classes(Bambrick,2016).

- SVM works perfect while dealing with unknown data.
- It is efficient when the data is semi-structured run structured like texts trees and images.
- SVM properly measure dimensional data
- In SVM usually there are no danger of over fitting due to having generalization in practice.

SVM implementation

Like mentioned earlier, there are bunch of libraries that can help us implement the models easily.First these libraries are needed to be imported in the note book.Then the data set needed to be read and load into arrays for further processing. Numpy have been used for these job. For the testing purpose and to understand whether the data is loaded properly or not, few examples of the data set showed in Figure 7.1.

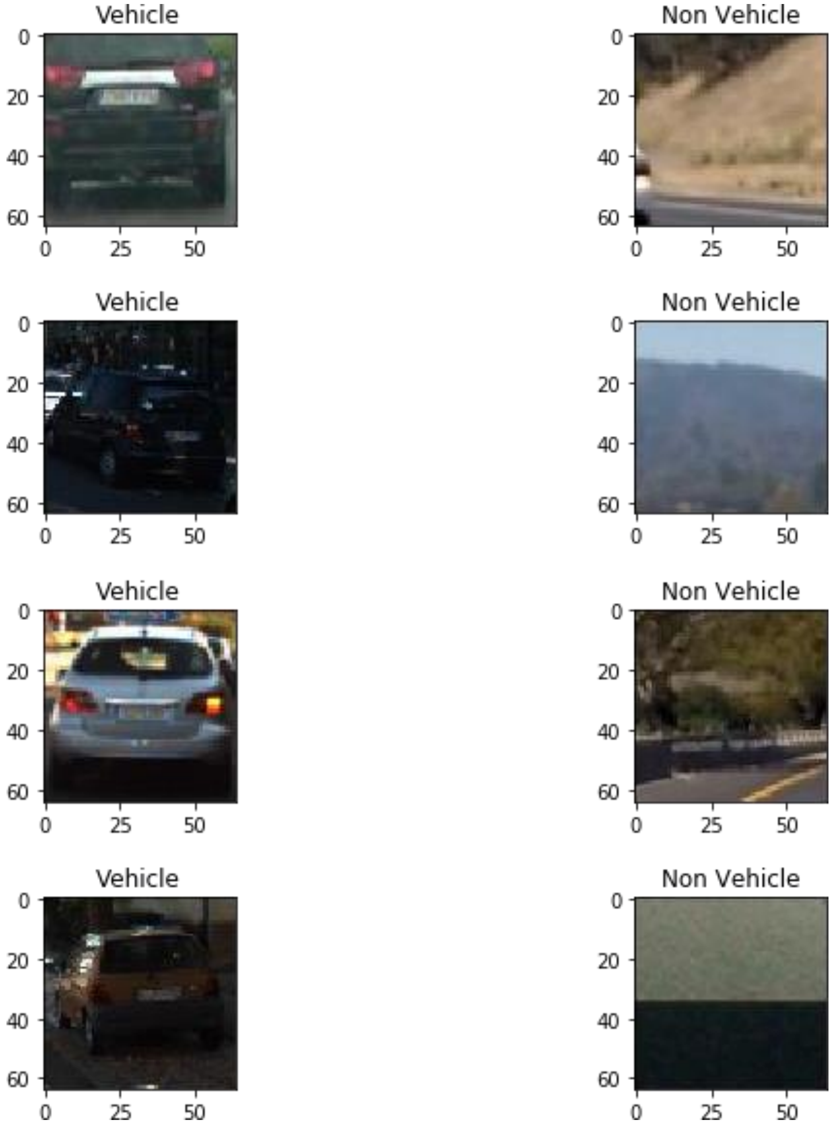


Figure7.1: Data set few instances

A histogram is an exact portrayal of the dissemination of numerical information. It is a gauge of the likelihood circulation of a continuous variable(CORAL)and was first presented by Karl

Pearson. It varies from a bar graph, as in a bar diagram relates two factors,yet a histogram relates just one. To develop a histogram, the initial step is to "bin" (or "bucket") the scope of qualities—that is, partition the whole scope of qualities into a progression of intervals—and after that check what number of qualities fall into every in term.There accept acles are typically determined as successive, non-covering intervals of a variable. The containers (intervals) must bead joining, and are frequently (yet are not required to be) of equivalent size. Color features and histogram for vehicles and non-vehicles are presented in Figure7.2 and7.3.

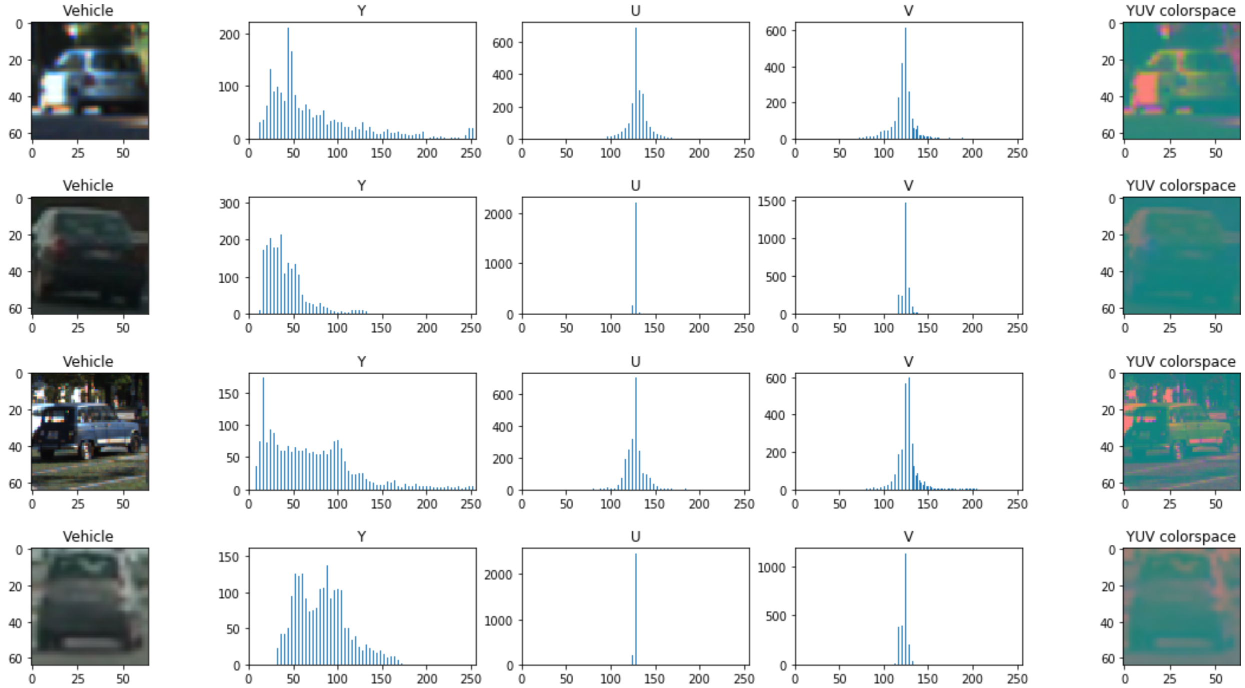


Figure7.2: Histogram and Color feature of Vehicles

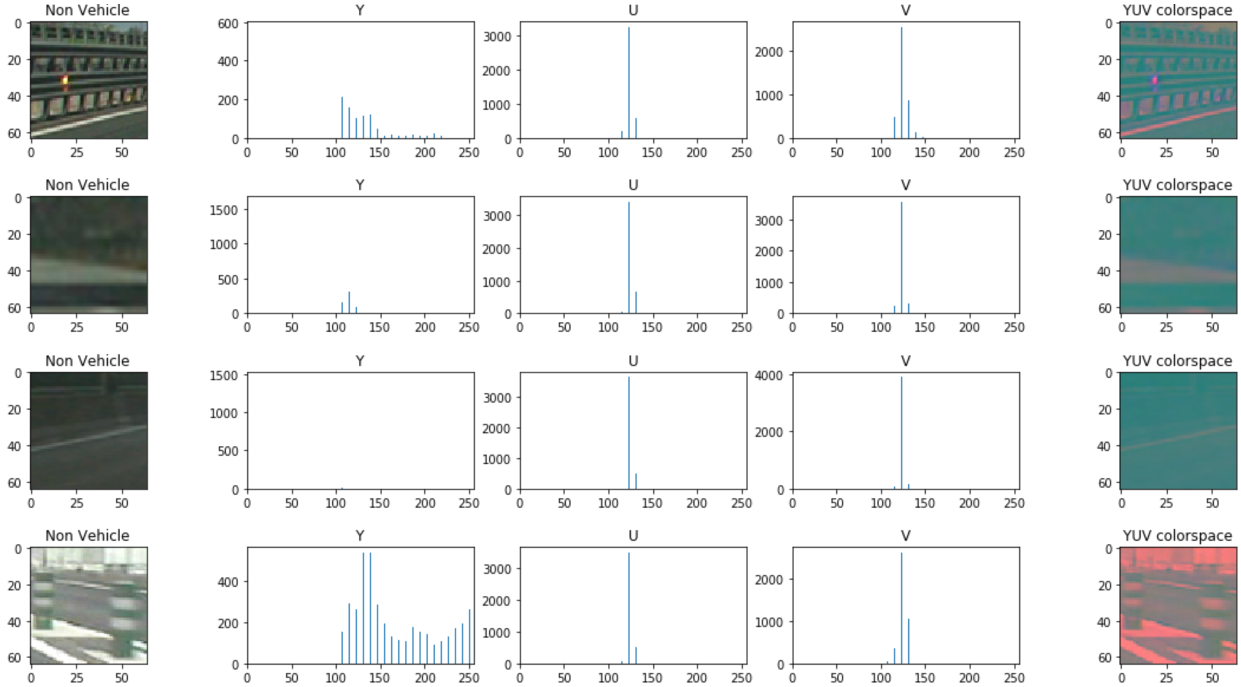


Figure7.3: Histogram and color feature of non-vehicles

After the extraction of the color feature,the data is going to bin.We have spatial binning which is “This modifier generates a 1-, 2- or 3-dimensional grid covering the simulation domain and assigns each particle into one of the uniformly sized bins. It then performs a reduction operation for a selected particle property, mapping the values of all particles contained in a cell to a single output value. This modifier can thus be used to project the per-particle data to a structured grid,for example to coarse-grain the atomistic data and generate a continuous field representation of a particle property. You can choose between different reduction operations, e.g. sum, average(mean), minimum or maximum. The bin grid can be one-, two- or three-dimensional, i.e. the simulation domain can be subdivided into equally sized bins along one, two or all three of its axes. The spatial bins are always aligned parallel to the simulation cell edges” (Chen, 2015). The action is conducted on the data set via binning functions and the results are as follows:

No of features before spatial binning 12288
No of features after spatial binning 768

Here after the histogram of oriented gradients are extracted. “The histogram of oriented gradients(HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique ecounts occurrences of gradient orientation in localized portion of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy".The result of the HOG extraction is showed inFigure5.4 and the following:

Feature Vector Length Returned is 324

No of features that can be extracted from image 4096

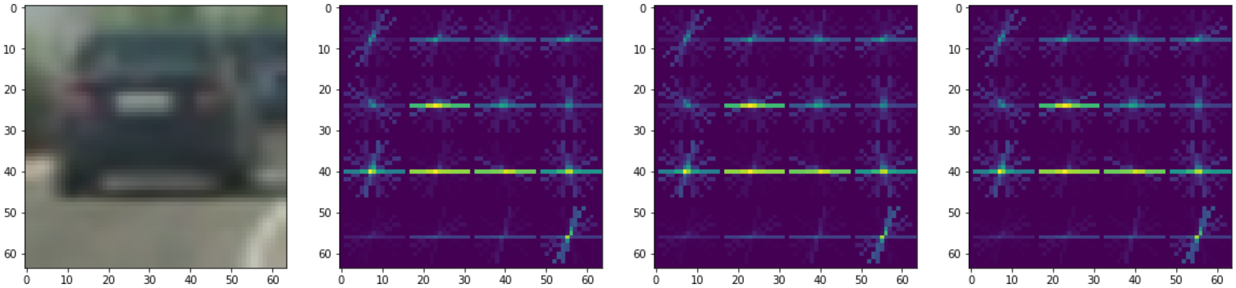


Figure7.4:HOG features extraction from one sample of the vehicles

The next step is data preprocessing, for training any algorithm we need to prepare data, do some cleaning, scaling etc. to feed the algorithm. Since the dataset here is an imagery dataset,there are no need for cleaning.Therefore, in this step the data splitting is needed to be done.

Data is split into two parts training and testing by the help of the train_test_split() function from the Scikit-Learn library. The percentage for both of them are considered like in general 80% for training and 20% for testing. Besides that the data has to be normalized. In the procedure of the SVC classifier the Standard Scaler() function has been utilized in order on normalize the data.

After all these processes the SVC classifier has been trained and the result that have been obtained from the algorithms is presented in table7.1

Table7.1:SVC classifier training results

Parameters	**Values**
Time	0.9second
Error Count	43
Accuracy	0.9879
Mean Squared Error	0.012106
Root Mean Squared Error	0.110027
Mean Absolute Error	0.012106

look in the table the algorithm is trained in 0.9 second and has 98% accuracy. With those values in evaluation matrices. When the model is trained with the training part of the dataset, the sliding window is the next task. There are functions to draw sliding windows. After running the sliding,the system needed to find the windows on which we are going to run the classifier. The function which is returning the refined windows in which the classifier predicts the output to be a car.Then the function which is drawing the main window around the identified cars is needed to be run.The sliding window and the identified car in drawn window around the car in the images are presented in Figure7.5.

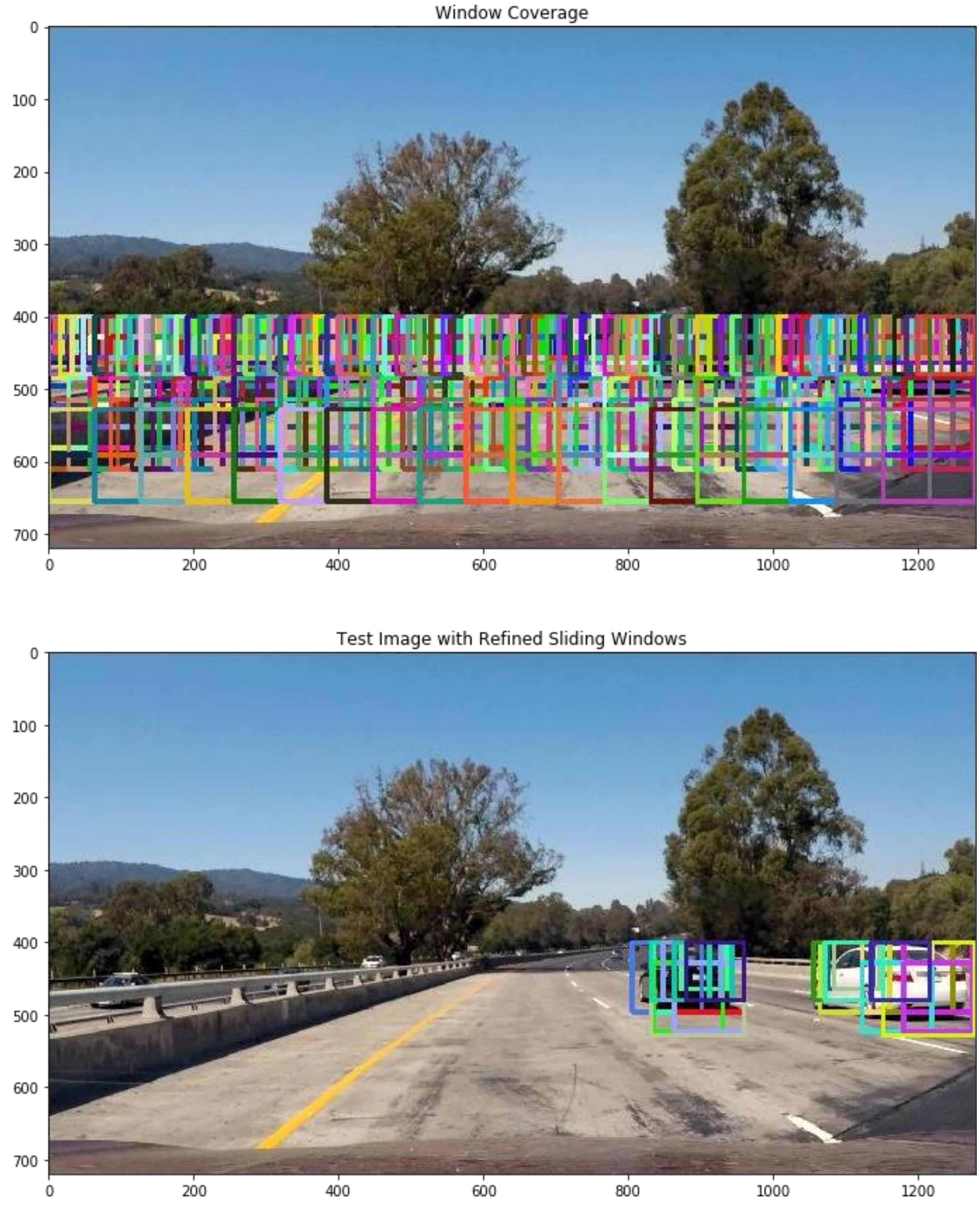


Figure7.5: Sliding window with the refined sliding windows

The process of drawing boxes around the cars are getting precise. Then after these all the heatmap is applied in order to increase the pixel by one inside each box. Then we need to apply a threshold value to the image to filter out low pixel cells and find pixels with each car number and draw the final bounding boxes.The testing heat map images are shown in Figure7.6.

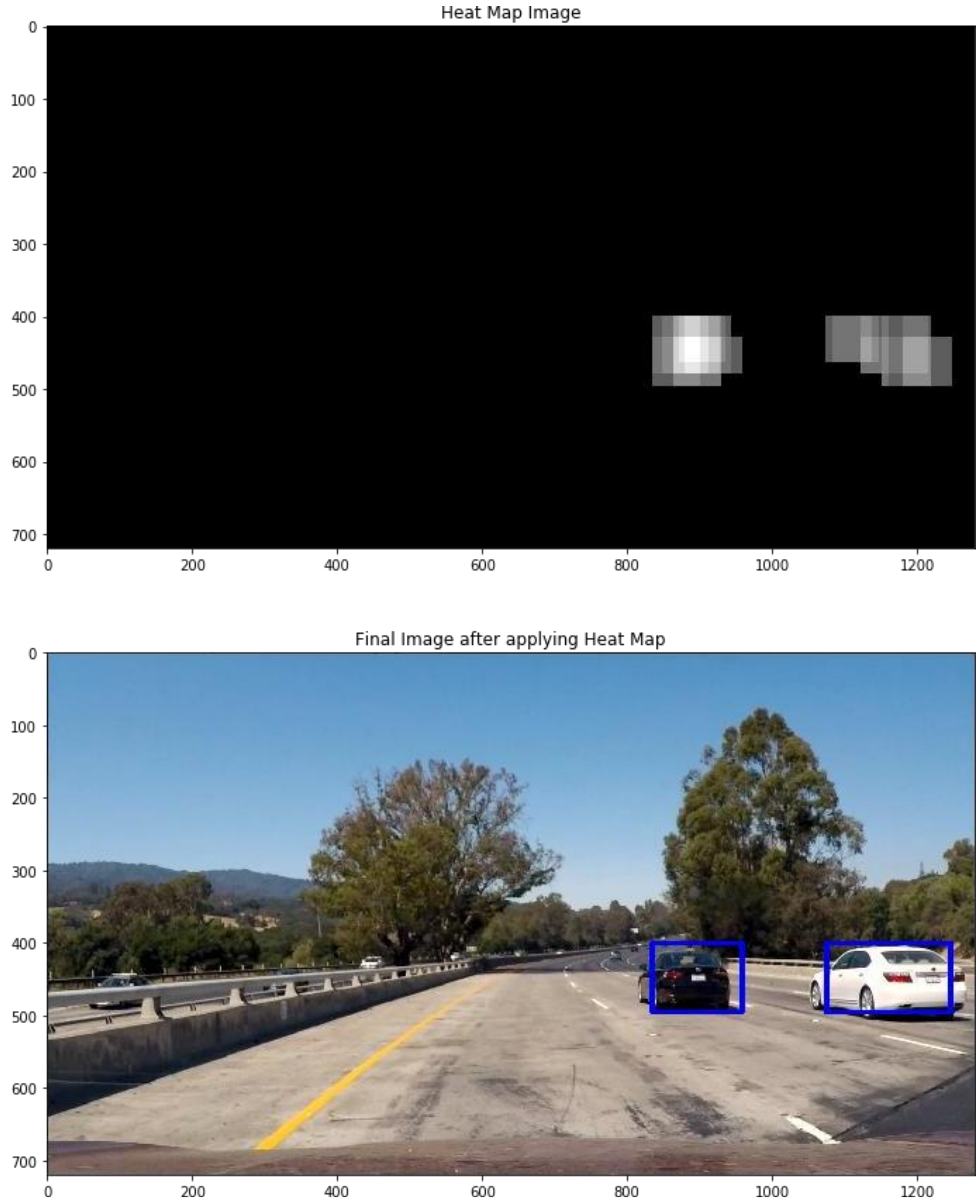


Figure7.6: Heat Map on testing image

Till here the models are detecting the cars and drawing boxes around them. The main task is complete. Although, it can be said that the system can detect the cars but cannot track them in a video. In order to do that a pipeline needed to be defined. The pipeline creator function needs multiple parameters to load a video. The pipeline function has been created. A sample of the pipe line result on Test images is shown in Figure7.7.

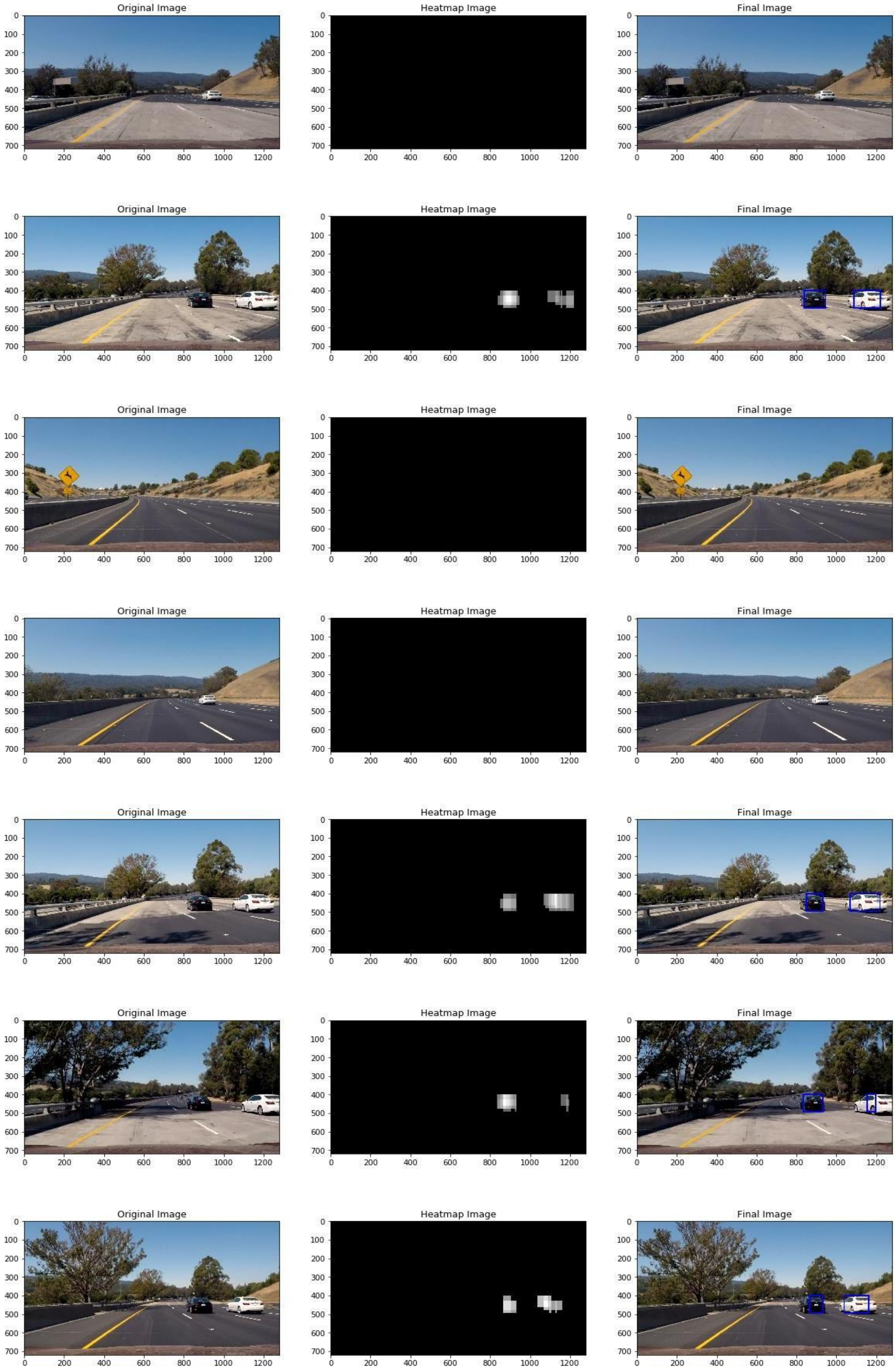


Figure7.7: Pipe line sample on test images

Same concept is conducted on the videos that are tested in the thesis. Since the video is the series of images. These concept is applied on them and the tracking process is done. While the cars passing in the video the program detect and track the cars with a box around the car.

Decision Tree

Decision Tree is a drawing or in other words graphical drawing like boxes of the algorithms. The algorithm clarify and describe the decisions which are feasible, utility, cost and consequences of issues. Although, in comparison of these kind of choices it permits one pane (Nayab & Scheid,2011)

Advantages

The decision tree advantages are listed below:

- Flexibility
- Specificity
- Resilience
- Transparency
- Ease of Use
- Comprehensive

Decision Tree Training

For decision tree everything is the same as previous algorithm except the classifier which is Decision Tree. All preprocessing steps and other techniques which are utilized in SVM is also used in this model. Due to the similarity we skip describing repetitive topics and present the result of the algorithm in Table 7.2. In order to understand how Decision Tree works with these kind of data. And how accurate is the result of prediction.

Table7.2: Decision Tree classifier training results

Parameters	**Values**
Error Count	205
Accuracy	0.942
Mean Squared Error	0.057714
Root Mean Squared Error	0.240237
Mean Absolute Error	0.057714

Like shown above Decision Tree obtained 94% accuracy in classification prediction of our data set with 205 errors and those values in evaluation matrices.

Comparison and the Best Model among the Models

The two models are tested on the same data set with the same computer even most of the techniques were the same in order to define which algorithm suited for this task. Because of the importance of the vehicle detection and tracking, the topic sensitive and precise to evaluate the results.In order to understand the two models and compare them, the result of the two models are presented in bar charts in a comparative way in Figure7.8.

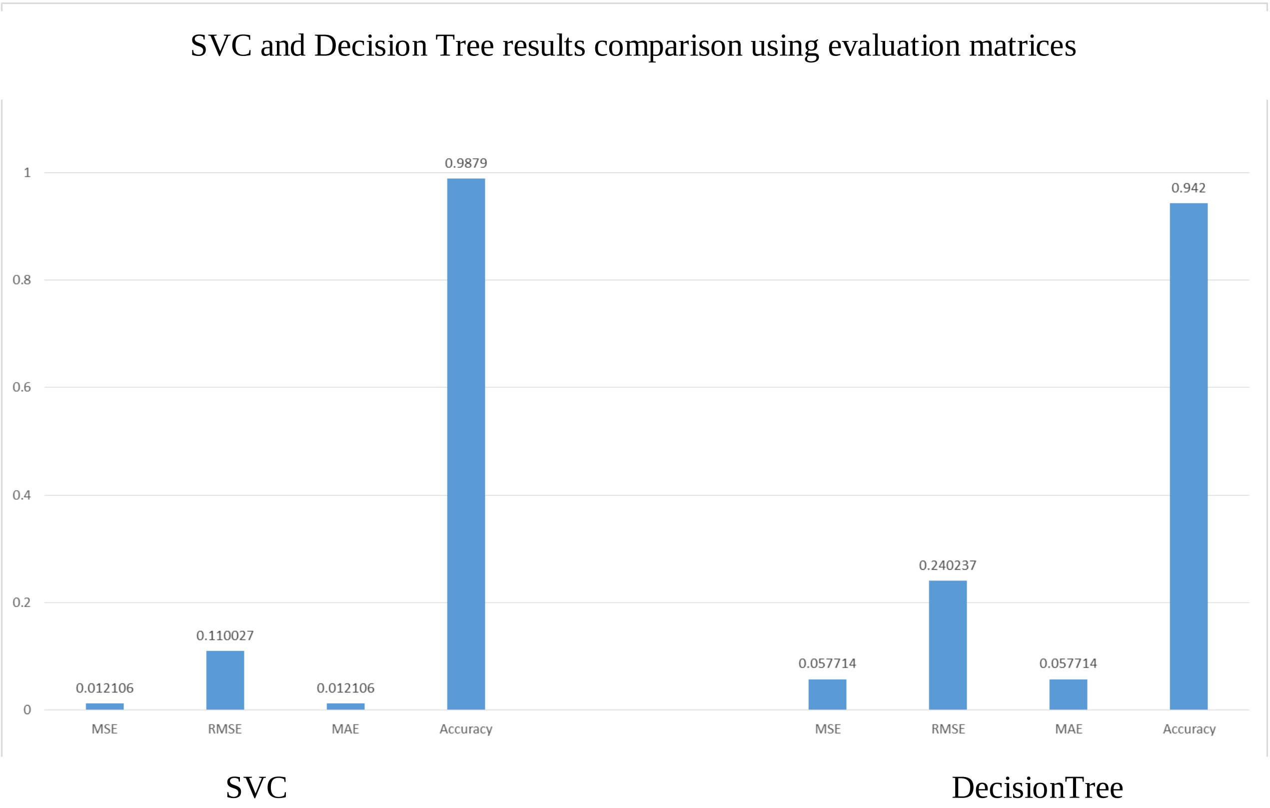


Figure7.8:SVC and Decision Tree Comparison result.

As you can see in the figure, we have three evaluation matrices with accuracy percentage level.Since the evaluation matrices are the error levels, as much as the values for MSE, RMSE, and MAE are low the accuracy level is high and if they are high then the accuracy level is low. As you can see in the figure all MSE, RMSE, and MAE are lower in SVC at the same time accuracy level is 98% for SVC but for Decision Tree the Error levels are high and the accuracy level is94%.This shows that SVC works better for vehicle detection and tracking tasks.

CONCLUSION AND RECOMMENDATIONS

By rapid development in car and traffic industries, at the same the growth of population in the world brought the needs for different tools and techniques specially technology solutions in order to manage traffics in cities and populated areas.Mean while, object detection can be used in various fields to help humans live easily with comfort and make the world a better place to live in.

Object detection can be used in industries, digitized cities, government, research, academia,environment etc. Vehicle detection and tracking is part of the object detection which is used in traffic, cities etc. the importance of the topic is growing larger. That being said this research is intended to contribute the improvement of the accuracy of these algorithms and models via available techniques and tools.

This Thesis developed two classifier algorithms to detect and track vehicles. These two models are Support Vector Machine (SVM) and Decision Tree. The algorithm selection was based on various studies in literature review. The most suggested models by other researchers were these two model. Therefore, the author decided to choose these models and compare them in order to specify the best model among these two. Many techniques have been deployed to increase the accuracy level and to make the best result possible. The models are trained with the same dataset and the evaluation result showed that SVM performs better than Decision Tree. The result of the models presented both in image and video formats which the system detect the cars that are passing from the screen and tracking them as well.

Future Works

Looking back to the limitations of the study, there are tasks and options which can be added to this research or possible to work on it separately. The topic is under the attention of researchers and improves day by day.Theoretical development is needed to be tracked and when any theoretical development is published and achieved, the researchers should utilize those concepts practically using algorithms in order to improve the accuracy level of the detection and tracking process.

Although, the dataset can be improved, a future work can be testing these models using a better and larger dataset with a massive number of vehicle and non-vehicle images from different places angels cars roads, cameras, distances etc.

Further more, other models can be added to the comparison list of models in order to make the comparison more reliable and vast.

Future Enhancements

Advanced Machine Learning Integration:

One paramount avenue for future development involves the infusion of more sophisticated machine learning algorithms into the system's architecture. Delving into advanced techniques such as deep learning, specifically convolutional neural networks (CNNs) and recurrent neural networks (RNNs), could significantly elevate the system's capacity to handle intricate traffic scenarios. These neural networks possess the potential to enhance the accuracy of vehicle detection and speed estimation, particularly in challenging conditions such as adverse weather or low-light situations.

Multi-Sensor Fusion for Comprehensive Insight:

To further enhance the system's reliability and comprehensiveness, integrating data from diverse sensor modalities emerges as a key consideration. A cohesive fusion of information from video cameras, radar systems, and lidar sensors can provide a more holistic understanding of the traffic environment. This multi-sensor approach not only enriches the quality of data but also enhances the system's detection and tracking capabilities, ensuring a robust response to varying traffic scenarios.

Predictive Analytics for Proactive Traffic Management:

The integration of predictive analytics stands out as a transformative enhancement, allowing the system to move beyond reactive measures to proactive traffic management. By leveraging historical data and real-time inputs, the system can forecast future traffic conditions. This foresight empowers traffic management authorities with the ability to implement preemptive measures, minimizing congestion and optimizing overall traffic flow.

Edge Computing for Real-time Responsiveness:

Embracing edge computing capabilities is pivotal for achieving real-time responsiveness and minimizing latency in data processing. By executing data analysis closer to the data source (at the edge), the system can significantly reduce the time between data capture and actionable insights. This approach not only bolsters the system's efficiency but also ensures quick response times to potential incidents, contributing to enhanced overall performance.

Integration with Smart City Infrastructure:

To fortify its role in the broader smart city ecosystem, future enhancements should focus on seamless integration with other urban planning technologies. Collaborating with traffic signal control systems, smart intersections, and intelligent infrastructure can create a cohesive urban environment. Such integration facilitates a more harmonized flow of information, supporting optimized traffic management strategies and contributing to the realization of smart city initiatives.

Autonomous Vehicle Interaction and Support:

With the rise of autonomous vehicles, the system's future lies in its ability to interact effectively with these advanced technologies. Enhancements could involve providing essential data to autonomous vehicles for navigation and decision-making. This collaborative approach ensures a smooth coexistence between conventional and autonomous vehicles, fostering safer and more efficient traffic flow.

Continuous User Interface Evolution:

Ongoing improvements to the user interface are paramount to ensure an intuitive and informative experience for operators. Advanced visualization techniques, such as augmented reality overlays, could transform the monitoring experience. The incorporation of intuitive visualizations and interactive elements can empower operators to make informed decisions swiftly and effectively.

Environmental Impact Monitoring:

As societal concerns regarding environmental sustainability grow, future enhancements could extend the system's capabilities to monitor and report on its ecological impact. Integration with air quality sensors and environmental monitoring devices can provide valuable insights into the system's environmental footprint. This environmental awareness aligns with global sustainability goals and contributes to responsible urban development.

Cyber security Fortification:

Given the increasing connectivity and reliance on technology, future enhancements must prioritize robust cyber security measures. Implementing advanced encryption protocols, intrusion detection systems, and regular security audits will safeguard the system against potential cyber threats. Cyber security fortification is integral to maintaining public trust and ensuring the system's resilience in an evolving threat landscape.

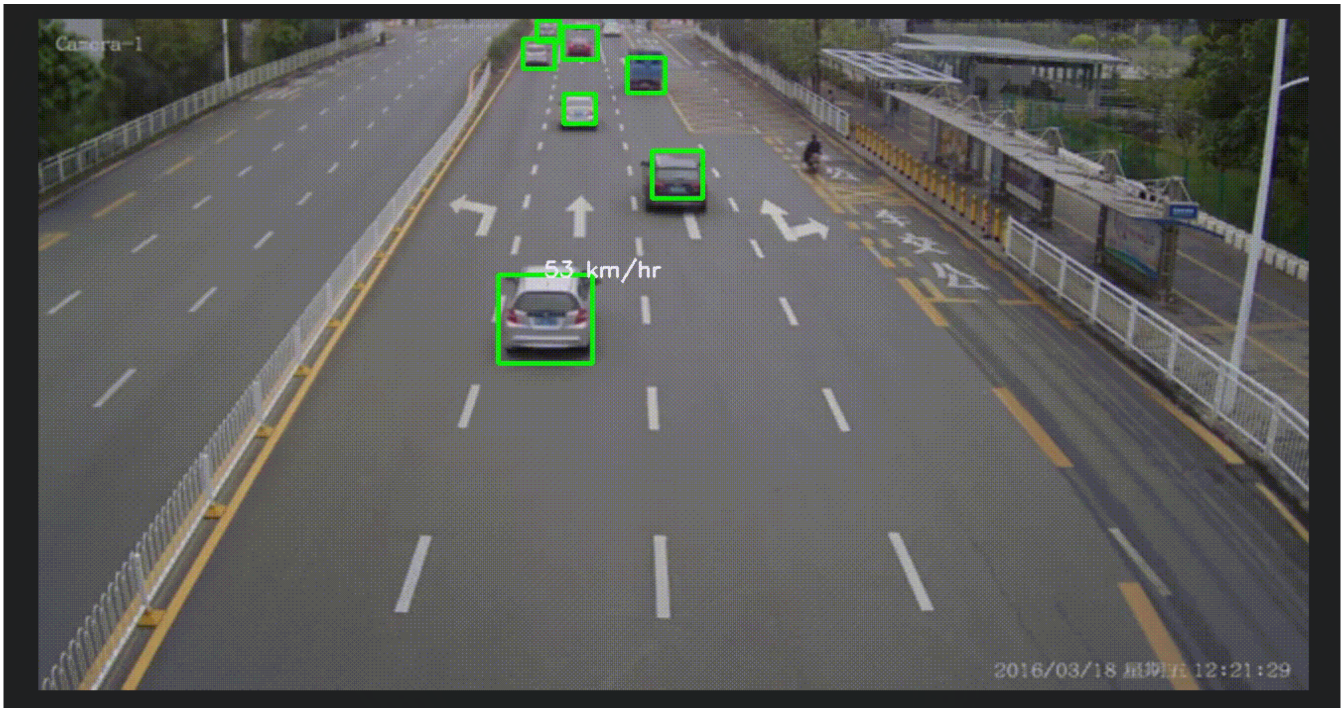
Customization and Configurability Empowerment:

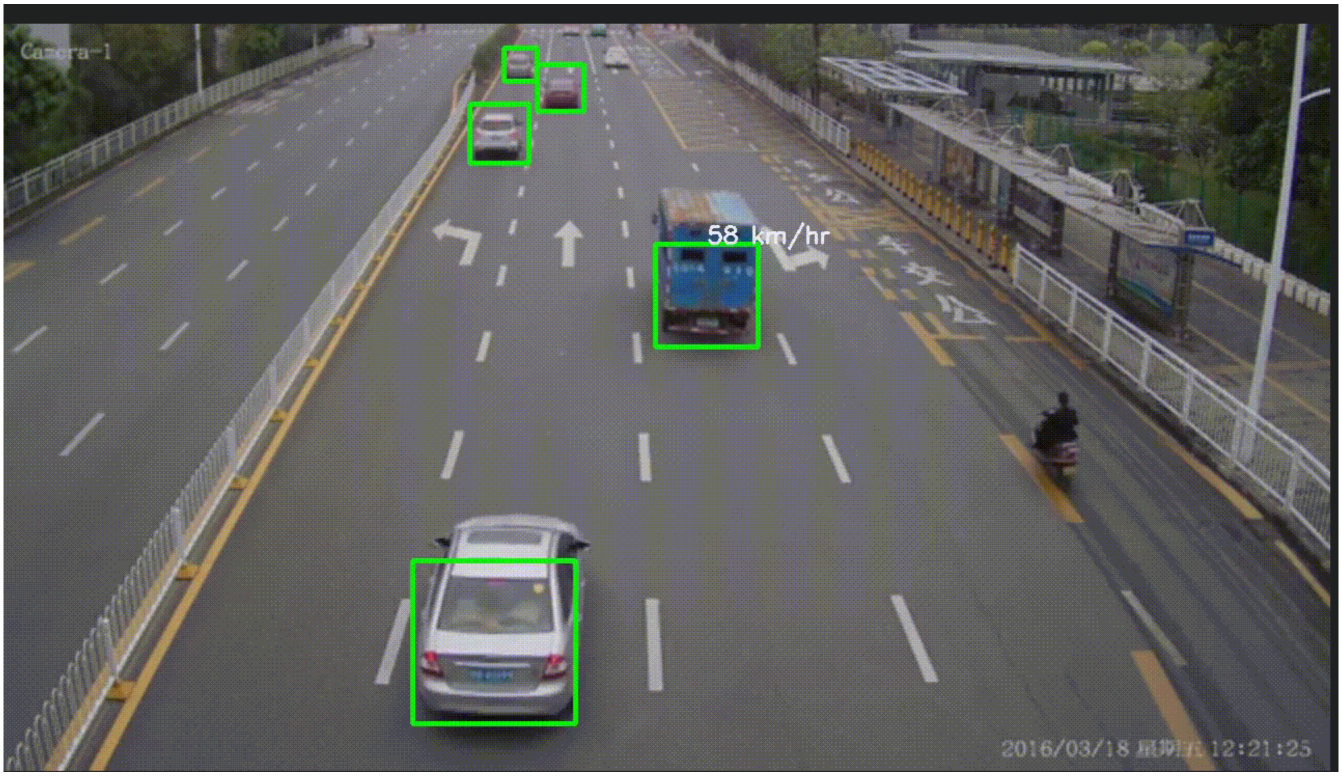
Providing administrators with enhanced customization options and configurability is critical for future development. This empowerment allows users to tailor the system to specific urban environments, traffic conditions, and regulatory requirements. A more flexible and adaptable system ensures its relevance across diverse settings, accommodating the unique needs of different municipalities.

APPENDIX A :SCREENS:

INPUT SCREENS :

OUTPUT SCREENS :




Camera-1
58 km/hr
2016/03/18 星期五 12:21:25

APPENDIX B : REPORTS

1. User Interaction Analysis:

This report analyzes user interactions within the system's monitoring interface, shedding light on usage patterns and preferences. Findings indicate a robust engagement with real-time monitoring features, particularly during peak traffic hours, suggesting the system's effectiveness in meeting user needs.

2. Speed Detection Accuracy Study:

This report presents a comprehensive analysis of the speed detection component's accuracy. Statistical comparisons with ground truth measurements affirm the system's reliability in estimating vehicle speeds across diverse traffic scenarios and environmental conditions.

3. Traffic Flow Patterns:

An exploration of traffic flow patterns over an extended monitoring period is detailed in this report. The analysis identifies daily and weekly variations, peak hours, and congestion trends, providing valuable insights for optimizing urban traffic management strategies.

4. Alert Response Times:

This report scrutinizes the efficiency of the system's alerting mechanism by analyzing user response times. Data reveals prompt reactions to different alert types, emphasizing the crucial role of timely responses in incident management. Recommendations for optimizing alert notifications are provided.

5. Environmental Impact Assessment:

The environmental impact assessment assesses factors such as energy consumption and hardware requirements. Findings indicate a minimal ecological footprint, aligning with sustainability goals. The report offers suggestions for further reducing environmental impact.

6. System Performance Metrics:

Comprehensive metrics on system performance, including processing times and memory utilization, are detailed. The analysis covers performance under varying loads and environmental conditions, offering insights into scalability and resource management. Recommendations for optimizing system performance are outlined.

7. User Satisfaction Survey Results:

   Results from a user satisfaction survey among administrators, operators, and end-users are presented. Feedback on system usability, features, and overall satisfaction levels is analyzed. Insights from the survey inform ongoing system improvements, ensuring alignment with user expectations.

APPENDIX C:DATA DICTIONARY

Data Dictionary for Vehicle Detection and Speed Detection System

The data dictionary for the "Vehicle Detection and Speed Detection in Java" project provides a comprehensive overview of the key data elements and structures used within the system. This includes details on the information captured, its purpose, data types, and relationships.

1. Frame Data:
   - Purpose: Represents a single frame from the video feed.
   - Data Elements:
     - `FrameID`: Unique identifier for each frame.
     - `Timestamp`: Time at which the frame was captured.
     - `Image`: Pixel data representing the visual content of the frame.

2. Vehicle Data:
   - Purpose: Contains information about detected vehicles.
   - Data Elements:
     - `VehicleID`: Unique identifier for each detected vehicle.
     - `FrameID`: Frame in which the vehicle was detected.
     - `BoundingBox`: Coordinates specifying the location of the vehicle in the frame.
     - `Class`: Type or category of the detected vehicle (car, truck, etc.).

3. Speed Data:
   - Purpose: Stores speed-related information for each detected vehicle.
   - Data Elements:

    - `VehicleID`: Identifier linking speed data to a specific vehicle.
    - `Speed`: The estimated speed of the vehicle in kilometers per hour.
    - `Timestamp`: Time at which the speed was calculated.

4. System Configuration:
  - Purpose: Records parameters and settings used by the detection system.
  - Data Elements:
    - `Algorithm`: Name of the vehicle detection algorithm employed (e.g., YOLO, SSD).
    - `AccuracyThreshold`: Minimum confidence score for considering a detection valid.
    - `FrameRate`: Frames per second (fps) setting for video processing.

5. Environmental Conditions:
  - Purpose: Captures information about the external environment during detection.
  - Data Elements:
    - `Weather`: Describes weather conditions (clear, cloudy, rainy).
    - `Lighting`: Indicates the illumination level (daylight, night).
    - `TrafficDensity`: Approximate number of vehicles in the frame.

6. System Logs:
  - Purpose: Stores system events and error messages for debugging.
  - Data Elements:
    - `LogID`: Unique identifier for each log entry.
    - `Timestamp`: Time at which the log entry was generated.
    - `Event`: Description of the system event or error.

7. Traffic Flow Data:
  - Purpose: Aggregated data on the flow of traffic over a specific duration.

- Data Elements:
  - `Date`: Date for which traffic flow is recorded.
  - `TimeSlot`: Specific time interval within the day.
  - `AverageSpeed`: Average speed of all detected vehicles during the interval.
  - `TotalVehicles`: Count of vehicles passing through the monitored area.

8. User Configuration:
   - Purpose: User-specific settings and preferences for system interaction.
   - Data Elements:
     - `UserID`: Unique identifier for each user.
     - `NotificationPreferences`: User's preferred notification settings.
     - `CustomSettings`: User-defined parameters for the system.

9. Geospatial Data:
   - Purpose: Location-related information for vehicles and detection zones.
   - Data Elements:
     - `Latitude`: Geographic latitude of the vehicle or detection zone.
     - `Longitude`: Geographic longitude of the vehicle or detection zone.
     - `ZoneID`: Identifier for a specific detection zone.

10. Alerts and Notifications:
    - Purpose: Records alerts and notifications triggered by the system.
    - Data Elements:
      - `AlertID`: Unique identifier for each alert.
      - `Timestamp`: Time at which the alert was generated.
      - `Type`: Classification of the alert (speed violation, system error).

This data dictionary provides an in-depth understanding of the various data elements used within the "Vehicle Detection and Speed Detection in Java" project. It serves as a comprehensive reference for developers, analysts, and stakeholders involved in the implementation, management, and analysis of the system's data. The meticulous documentation of each data element ensures clarity, accuracy, and consistency in the utilization and interpretation of data throughout the project lifecycle.

APPENDIX D : OPERATIONAL MANUAL

Operational Manual for Vehicle Detection and Speed Detection System

Introduction:

The operational manual for the "Vehicle Detection and Speed Detection in Java" system provides detailed guidance on the deployment, configuration, and day-to-day operations of the system. This manual is designed to assist administrators, operators, and system users in ensuring the effective functioning of the system in various environments.

System Overview:

The Vehicle Detection and Speed Detection system is a comprehensive solution for real-time monitoring of vehicular traffic, offering accurate detection and speed estimation capabilities. The system is built on Java programming, leveraging advanced computer vision algorithms for vehicle detection. The operational manual covers the following key aspects:

1. System Installation:

- The installation process is straightforward and involves deploying the Java-based application on the target system.
- Administrators should ensure that the system meets the hardware and software requirements specified in the system documentation.
- The manual guides users through the installation steps, including the configuration of necessary parameters such as video input sources and detection algorithm preferences.

2. Configuration:

- The system provides a configuration interface accessible to administrators for tailoring the system to specific deployment scenarios.
- Configuration parameters include video source settings, detection algorithm choices, accuracy thresholds, and notification preferences.
- Users are guided through the configuration process to optimize the system for varying traffic conditions and environmental factors.

System Operation:

3. Start-Up and Shutdown Procedures:
   - The manual provides clear instructions on starting and shutting down the system to ensure proper initialization and termination.
   - Users are guided through the necessary steps to initiate the application, load video feeds, and perform a systematic shutdown when needed.
4. Real-Time Monitoring:
   - The system's real-time monitoring interface allows users to visualize detected vehicles, their classifications, and speed estimates.
   - Key features of the monitoring interface, such as zoom, pan, and playback functionalities, are explained in detail.
   - Users learn how to interpret the displayed information, including bounding box coordinates, vehicle types, and speed values.
5. System Alerts and Notifications:
   - The system generates alerts and notifications for events such as speed violations or potential errors.
   - The manual outlines the types of alerts, their meanings, and how users can customize notification preferences.
   - Administrators are guided on configuring and managing alert notifications to ensure timely responses to critical events.

System Maintenance:

6. Log Analysis and Troubleshooting:
   - System logs capture events and errors for diagnostic purposes.
   - The manual explains how to access and interpret log files, facilitating troubleshooting and performance analysis.
   - Users are provided with guidance on common issues, their potential causes, and recommended resolutions.
7. Software Updates and Upgrades:

- Periodic software updates and upgrades are essential for system security and performance enhancements.
- The manual details the procedures for updating the software, including version checks, download links, and installation steps.

Conclusion:

The operational manual serves as a comprehensive guide, ensuring that users can effectively deploy, configure, and operate the "Vehicle Detection and Speed Detection in Java" system. Following the instructions provided in this manual will contribute to the seamless integration of the system into diverse traffic management scenarios, enhancing road safety and efficiency.

BIBLIOGRAPHY

REFERENCES

Andrew, W. M. and Victor, M. (2003), Handbook of International Banking (*London: EdwardElgarPublishing Limited*), 350-358

Baha,N.(2014). Real-Time Obstacle Detection Approach using StereoscopicImages. *InternationalJournalofInformationEngineeringand ElectronicBusiness*,*6*(1), 42.

Bahrepour,M.,Akbarzadeh-T,M.R.,Yaghoobi,M.,&Naghibi-S,M.B.(2011).AnadaptiveorderedfuzzytimeserieswithapplicationtoFOREX.*ExpertSystemswithApplications*,38(1),475-485.

Bambrick, N. (2016, June 24). *Support Vector Machines for dummies; A Simple Explanation*.Retrieved April 28, 2018, from AYLIEN | Text Analysis API | Natural LanguageProcessing: http://blog.aylien.com/support-vector-machines-for-dummies-a-simple/pp.1-13.

Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. *Neural InformationProcessing-Lettersand Reviews*, *11*(10), 203-224.

Berni, J. A., Zarco-Tejada, P. J., Suárez, L., & Fereres, E. (2009). Thermal and narrowbandmultispectralremotesensingforvegetationmonitoringfromanunmannedaerial vehicle.*IEEE Transactions on Geoscienceand RemoteSensing*, *47*(3), 722-738.

BIS.(2016).Triennialcentralbanksurvey:ForeignexchangeturnoverinApril2016Monetaryand EconomicDepartment.

Chen, X., & Meng, Q. (2015, November). Robust vehicle tracking and detection from UAVs.In *20157thInternationalConferenceofSoftComputingandPatternRecognition(SoCPaR)*(pp. 241-246).IEEE.

Chen.X,(2015),AutomaticVehicleDetectionandTrackinginAerial Video,DoctorofPhilosophyThesis inLoughboroughUniversity.

Kalghatgi, M. P., Ramannavar, M., & Dr. Sidnal, N. S. (2015). A Neural Network ApproachtoPersonalityPredictionbasedontheBig-FiveModel.*InternationalJournalofInnovativeResearch in Advanced Engineering, 2(8), 56–63*

Kanistras, K., Martins, G., Rutherford, M. J., & Valavanis, K. P. (2015). Survey of unmannedaerial vehicles (UAVs) for traffic monitoring. *Handbook of unmanned aerial vehicles,*2643-2666.

Khan, K., Baharudin, B. B., & Khan, A. (2009, June). Mining opinion from text documents:Asurvey.In*DigitalEcosystemsandTechnologies,2009.DEST'09.3rdIEEEInternationalConference on* (pp. 217-222).IEEE.

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated design of both thetopologyandsizingofanalogelectricalcircuitsusinggeneticprogramming.In*ArtificialIntelligenceinDesign*'96 (pp. 151-170). Springer,Dordrecht.

Nayab, & Scheid, J. (2011, September 2). *Advantages of Decision Tree Analysis*. RetrievedApril 28, 2018, from Bright Hub Project Management: https://www.brighthubpm.com/project-planning/106000-advantages-of-decision-treeanalysis/

Noh.S,Shim.DandJeon.M,(2015),'AdaptiveSliding-WindowStrategy forVehicleDetectioninHighwayEnvironments',*IEEEIEEETransactionsonIntell.Transport.Syst.*,Ofor,e.(2018).*Machinelearningtechniquesforimmunotherapydatasetclassification*(Masterdissertation,Near East University).

Patel, P. J., Patel, N. J., & Patel, A. R. (2014). Factors affecting currency exchange rate,economical formulas and prediction models. *International Journal of ApplicationorInnovationin Engineering&Management(IJAIEM)*, 3(3), 53-56.

Pujari, M. V., Sayyed,A. H., Shahani, H., Rupani, D., & Student, B. E. (2018). ForexTradingSystem.*International Journal ofEngineering Science*, 17116.

Razakarivony, S., & Jurie, F. (2016). Vehicle detection in aerial imagery: A small targetdetectionbenchmark.*JournalofVisualCommunicationand*

ImageRepresentation,34, 187-203.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia;PearsonEducationLimited.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia;PearsonEducationLimited.

Sahli, S., Ouyang, Y., Sheng, Y., & Lavigne, D. A. (2010, April). Robust vehicle detection inlow-resolution aerial imagery. In *Airborne Intelligence, Surveillance, Reconnaissance(ISR) Systems and Applications VII* (Vol. 7668, p. 76680G). International Society forOpticsand Photonics.

Sayad,S.(2017). SupportVectorMachine.*SupportVector Machines.Np,ndWeb,26*.

Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction usingbreaking financial news: The AZFin text system. *ACM Transactions on InformationSystems(TOIS)*, 27(2), 12.

Sokalski, J., Breckon, T. P., & Cowling, I. (2010). Automatic salient object detection in uavimagery.*Proc. of the25th Int.Unmanned Air VehicleSystems*, 1-12.

Su,A., Sun,X.,Liu,H., Zhang,X.,& Yu,Q. (2015). Onlinecascadedboostingwithhistogram of orient gradient features for car detection from unmanned aerial vehicleimages.*Journal of Applied RemoteSensing*, *9*(1),096063.

Susaki, J. (2015). Region-based automatic mapping of tsunami-damaged buildings usingmulti-temporalaerial images.*Natural Hazards*,*76*(1), 397-420.

Tenti, P. (1996). Forecasting foreign exchange rates using recurrent neural networks. *AppliedArtificialIntelligence*, 10(6), 567-582.

TheFederalReserveBoard.(2004)"FRB:Speech,Bernanke--InternationalMonetaryReformand Capital Freedom--October14, 2004".

Tsai, Y. C., Chen, J. H., & Wang, J. J. (2018). Predict Forex Trend via Convolutional NeuralNetworks.*arXiv preprintarXiv*:1801.03018.