

Parking Detection System using

SVM-HOG

**Project Report**

***Team Members***

Reshmah C S (715520104040)

Swathy S (715520104052)

Mugilan S (715520104031)

***Project Guide***

Dr D Sivaganesan

Professor

Department of Computer Science and Engineering PSG Institute of Technology and Applied Research Neelambur, Coimbatore

### ACKNOWLEDGMENT

First and foremost, we express our deep sense of gratefulness to our respected Managing Trustee, **Shri. L. GOPALAKRISHNAN** for his provision to utilize all the necessary facilities in the institution.

We express our heartfelt gratitude to our beloved Principal of our institution, **Dr. G. CHANDRAMOHAN** for his overwhelming support and encouragement on this project.

We would also like to extend our heartfelt thanks to our honorable Secretary **Dr. P. V. MOHANRAM** for his moral support.

We are greatly indebted to Head of the Department of Computer Science and Engineering, **Dr. R. MANIMEGALAI** for her guidance and continuous support which was instrumental in the completion of this project.

We extend our thanks to our guide, **Dr. D. SIVAGANESAN** for his guidance and supervision, without which we could not completed this project study.

We would further like to thank our project coordinator **Dr. I. KALA** for carrying out reviews smoothly and the valuable feedback provided at each step of the project development.

Finally, I take this opportunity to extend my humble gratitude to Almighty without the blessings of whom we would not have been able to overcome the challenges posed by this project study.

RESHMAH C S

SWATHY S

MUGILAN S

# ANNA UNIVERSITY: CHENNAI - 600 025

## BONAFIDE CERTIFICATE

Certified that this project report “Parking Detection System using

SVM-HOG” is the bonafide work of “Reshmah C S (715520104040), Swathy S (715520104052), Mugilan S (715520104031)” who carried out the project work under my supervision.

### ------------------------- ------------------------

**SIGNATURE SIGNATURE**

Dr. R. Manimegalai Dr. D. Sivaganesan

### HEAD OF THE DEPARTMENT SUPERVISOR

Professor and Head Professor

Computer Science and Engineering Computer Science and Engineering PSG Institute of Technology and PSG Institute of Technology and Applied Research, Applied Research,

Coimbatore – 641 062 Coimbatore – 641 062

### Submitted for the project viva-voce Examination held on

**----------------------------------- ----------------------------------**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

## ABSTRACT

## 

With the increasing complexity of road environments and the growing number of vehicles on the road, ensuring road safety has become a paramount concern. Timely and precise detection of obstacles can prevent accidents, save lives, and reduce property damage. Finding a vacant parking space in a congested area or a large parking lot, specially, in peak hours, is always time consuming and frustrating to drivers. It is common for drivers to keep circling a parking lot and look for a vacant parking space. To minimize hassle and inconvenience to the drivers, many parking guidance systems have been developed over the past decade, where the system provides accurate, real-time car park space availability to the drivers looking for parking spaces and then guides them to the available spaces by dynamically updated guide signs.

***Keywords: Artificial Intelligence (AI), Machine learning, Computer Vision, Support Vector Machine(SVM), Histogram of Oriented Gradients (HOG)***

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **SNO** | **TITLE** | **PAGENO** |
|  | **ABSTRACT** | **4** |
| **1** | **INTRODUCTION** | **6** |
|  | 1.1 Working of framework | **7** |
|  | 1.2 Theoretical Background | 7 |
|  | 1.3 SVM  1.4 HOG  1.5 SVM-HOG | 8  9  10 |
| **2** | **SYSTEM DESCRIPTION** | **11** |
|  | 2.1 Problem Statement | 11 |
|  | 2.2 Project Description | 11 |
|  | 2.3 Project Goals | 12 |
| **3** | **PROPOSED SYSTEM** | **14** |
| **4** | 3.1 Description of the proposed systems  3.2 Algorithm  **LITERATURE SURVEY**  4.1 Analysis of existing systems | 14 |
| **5** | **RESULT AND ANALYIS**  5.1 Snapshots of feature extraction  5.2 Snapshots of SVM model  5.3 Result | **15** |
| **6** | **CONCLUSION AND FUTURE WORK** | **24** |
|  | 6.1 Conclusions | 24 |
|  | 6.2 Future Work | 24 |
|  | **APPENDIX** | **25** |
|  | **REFERENCES** | **31** |

**CHAPTER 1**

**INTRODUCTION**

## 1.1 Introduction

Artificial Intelligence (AI) has emerged as a transformative technology in addressing this challenge. AI, particularly machine learning and computer vision techniques, can enable vehicles to "see" and interpret their surroundings, thereby improving their ability to detect and respond to obstacles effectively. In urban areas, the inefficiency of traditional vehicle identification systems, exacerbated by increasing vehicle numbers and limited parking infrastructure, results in congestion and wasted time. The core issue is the challenge of obstacle detection for vehicles. The limitations of traditional obstacle detection systems, such as reliance on simple sensors or lack of advanced decision-making capabilities. These limitations make it challenging for vehicles to detect and respond to complex obstacles effectively. The complexity of modern road environments, including factors like weather conditions, diverse road users (pedestrians, cyclists), and dynamic traffic situations. These factors amplify the difficulty of obstacle detection.

The current parking guidance systems obtain the availability of parking spaces using the sensors installed across the whole parking lot. However, deploying sensors in a large parking lot can be very expensive. Furthermore, the sensors can become inaccurate and would stop functioning easily when time passes. Therefore, it is highly desired to have a reliable and cost-effective way to track available parking spaces and guide drivers to the available parking spaces. Besides searching for available parking spaces by using the popular machine learning algorithm Support Vector Machine(SVM) with  Histogram of Oriented Gradients (HOG) for the feature extraction.

**1.4. Scope and Motivation:**

In congested urban areas, the quest for an available parking space is a pervasive challenge, causing frustration and time loss for drivers, especially during peak hours. The routine of circling parking lots in search of a spot not only intensifies individual stress but also contributes to overall traffic congestion. This issue is exacerbated during peak hours, amplifying the time and fuel consumption associated with finding parking. Recognizing these challenges, there is a growing motivation to explore technological interventions, such as smart parking systems, to streamline and optimize the parking experience. Embracing innovation holds the potential to alleviate frustrations, reduce congestion, and enhance overall urban mobility efficiency.

**1.5. Problem Statement:**

Congestion and lost time are caused by the inadequacy of conventional vehicle identification systems in metropolitan areas, which is made worse by an increase in the number of vehicles and a lack of adequate parking facilities. The difficulty of vehicle obstacle detection is the main problem. The shortcomings of conventional obstacle detection systems, include their reliance on basic sensors or their inability to make complex decisions. It is difficult for cars to recognize and react to complicated impediments efficiently because of these restrictions. the intricacy of contemporary road environments, which include elements like fluctuating weather, a variety of road users (bikers, pedestrians), and dynamic traffic circumstances. Obstacle detection is made more difficult by these variables. This innovative solution employs machine learning and real-time data to accurately manage parking space occupancy, improving urban mobility and sustainability.

**1.6. Objective:**

In response to the perennial challenges faced by drivers in urban environments, the development of advanced parking guidance systems has gained momentum over the past decade. The primary objective of these systems is to alleviate the considerable hassle and inconvenience experienced by drivers when searching for parking spaces.

By harnessing technology, these systems provide drivers with accurate, real-time information about the availability of parking spaces. The integration of dynamically updated guide signs further enhances the driving experience by guiding individuals seamlessly to the identified available spaces.

**1.7. Contribution of the project:**

While current parking guidance systems predominantly rely on sensor networks deployed throughout parking lots, our project introduces an innovative approach by integrating real-time traffic information. Unlike conventional systems, our solution considers not only parking availability but also the prevailing traffic conditions. Empowering drivers with up-to-date insights about free parking spaces in proximity to their destination represents a significant advancement. This multifaceted approach aims to revolutionize the user experience, offering drivers a more informed and efficient means of navigating parking challenges in urban settings.

**1.8. Organization of the thesis:**

Chapter 2 is Literature Survey which shows the various analyses and research made in the field of interest and the results already published, considering the various parameters of the project and the extent of the project and the drawbacks in the already existing systems. Chapter 3 is System Description where it discusses the platforms and the technologies worked on. Chapter 4 is System Design and implementation which has the modules that are implemented and the system flow. Also, the frameworks used in it, collection of data, setting up the environment and the module implementation. Chapter 5 deals with the experimental setup and results and the efficiency are calculated and what is saved in it. Chapter 6 is Conclusions and Future enhancements that will improve the efficiency.

**CHAPTER 2**

**WORKING OF THE FRAMEWORK**

## 2.1 Theoretical background

To get to know better about the algorithms used in different modules of this work,

Software programs can predict outcomes more correctly without explicit instructions by using machine learning, a type of artificial intelligence. Machine learning algorithms [2] use historical data as input to predict new output values. Recommendation engines commonly employ machine learning. Other frequent uses include fraud detection, spam filtering, malware threat identification, business process automation, and predictive maintenance [6]. Software programmers may forecast outcomes more accurately with the help of machine learning, an artificial intelligence technique, without needing to be explicitly told to do so. Machine learning algorithms use historical data as input to forecast new output values. Machine learning is vital since it helps with the creation of new products and gives organizations an overview of consumer behavior trends and operational business patterns. Machine learning is crucial to the operations of many of the leading companies of today, like Facebook, Google, and Uber. Machine learning has become a major point of competitive difference for many firms. A typical method to categorise conventional machine learning is the process by which a prediction-making algorithm learns to increase its accuracy. There are four basic learning methods: supervised learning, unsupervised learning, and reinforcement learning [2]. The kind of algorithm that data scientists use depends on the type of data that they want to forecast. The Support Vector Machine (SVM) fall within the category of supervised learning [5].

## 2.2 What is HOG?

Histogram of Oriented Gradients, or HOG, is a feature descriptor used for object detection in computer vision and image processing. It was first presented in the 2005 publication "Histograms of Oriented Gradients for Human Detection" by Navneet Dalal and Bill Triggs.

**Histograms of Local Gradient Orientation:**

HOG operates by segmenting a picture into tiny, overlapping areas known as cells.

The gradient directions and magnitudes are computed for every cell. The image's intensity variations are represented by the gradient, and the direction of the change is indicated by the orientation.

**Calculating a Gradient:**

The gradient is computed in both the horizontal and vertical axes by convolving the image using basic filters, such as Sobel filters. The purpose of doing this is to record data regarding the intensity gradients in various directions.

**Orientation Binning:**

Following that, each pixel's orientation is quantized into discrete bins (often nine bins ranging from 0 to 180 degrees), with the gradient magnitude contributing to the bin that corresponds to each bin.

**Block Standardization:**

The cells are arranged into blocks, which are bigger areas. Blocks are used to achieve invariance to variations in light and contrast by providing local normalization.

The process of normalization involves scaling the values according to the total energy in each block by taking into account the block histograms.

**Formation of Descriptors:**

The normalized block histograms are concatenated to create the final HOG description. The local image structure is therefore compactly represented.

When it comes to identifying objects, HOG is exceptionally well-liked, especially when it comes to identifying people and pedestrians. Its success is due to its capacity to record object shape and edge details while remaining mostly unaffected by variations in contrast and lighting. It has been extensively utilised to train classifiers for object detection in photos in conjunction with machine learning methods like support vector machines (SVMs). HOG is an essential part of many object identification systems, such as OpenCV's well-known pedestrian detection engine.

**2.3 Support Vector Machine**

**2.3.1 What is SVM?**

A supervised machine learning technique used for regression and classification problems is called Support Vector Machine (SVM). Its main objective is to locate the best potential hyperplane in an N-dimensional space (where N is the number of features) for classifying the data. The data points that are closest to the decision boundary (hyperplane) and are essential in establishing its location are referred to as "support vectors."

**Linear Separation:** SVM is used for binary classification in its most basic form, where the goal is to locate a hyperplane in the feature space that divides two classes.  A decision boundary known as a hyperplane separates the feature space into areas that correspond to various classifications.

**Margin:** SVM searches for the hyperplane with the largest margin in addition to a separating hyperplane. The distance between the closest data point from each class and the hyperplane is known as the margin. By increasing the margin, you can strengthen the model's ability to generalize and make it more resilient to fresh, untested data.

**Support Vectors:** The data points that are closest to the decision border are known as support vectors. They play a vital role in defining the margin and the hyperplane. The hyperplane's orientation and position are only determined by the support vectors. As long as the other data points remain outside the margin, they have no effect on the model.

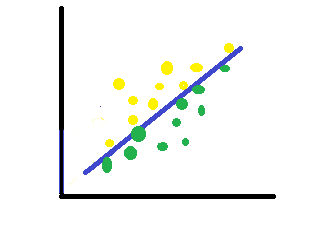
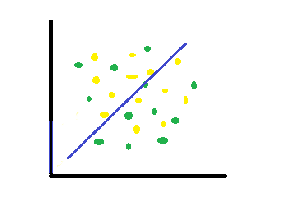
**Kernel Trick:** By utilizing the kernel trick, SVM can manage non-linear decision boundaries. The function of the kernel

**C Parameter:**  SVM has a regularization parameter, which is commonly represented by the letter C. It manages the trade-off between accurately identifying the training points and having a smooth decision boundary. While a greater C leads to a narrower margin with fewer misclassifications, a smaller C encourages a broader margin but may allow some misclassifications.

**Soft Margin SVM:** A soft-margin SVM permits some misclassifications when the data is not entirely separable. This is accomplished by adding a penalty term to the optimization aim for points that are incorrectly classified.

**Multi-Class SVM:** Using techniques like one-vs-one or one-vs-all, where multiple binary classifiers are trained and combined to create predictions for many classes, SVM can be extended for multi-class classification.

**SVM regression:** SVM may also be applied to regression tasks by minimizing deviations from the actual values and identifying the hyperplane that best fits the data.

Non - Linearly Separable dataset

Linearly Separable dataset

**2.3.2 Reasons why SVM is used?**

Because SVM works effectively in high-dimensional domains, it can be applied to applications involving a lot of features, such picture recognition and text categorization.

In order to reduce the likelihood of overfitting, particularly in high-dimensional spaces, SVM aims to maximise the margin. Regression and classification tasks are two applications for SVM. Fitting a hyperplane that forecasts a continuous output is the goal of regression.

SVM's versatility in capturing intricate patterns in data stems from its ability to handle non-linear decision boundaries, which is made possible by its kernel trick. Convex optimisation, on which SVM is based, guarantees that the algorithm finds the global minimum as opposed to becoming trapped in local minima.

## 2.4 The combination of SVM with HOG

This project is implemented using the machine learning algorithm Support Vector Machine (SVM) with the Histogram of Oriented Gradients (HOG) as the feature descriptor. The combination of these algorithms provides better results. A feature descriptor called HOG is used to describe the appearance and form of local objects in an image. It computes the gradient orientation in each of the small, connected regions called cells that make up an image. Histograms of directed gradients are then produced using the gradient information. The direction of intensity variations in various image regions is captured by these histograms. SVM is a supervised machine learning technique that may be applied to regression and classification problems. SVM is frequently used in the object detection context for binary classification, which separates an object's presence from absence in an image. SVM searches a high-dimensional feature space for a hyperplane that best divides the data points into several classes.

Integration of HOG with SVM for Object Detection: The process of combining HOG with SVM for object detection involves the following steps:

Phase of Training includes:

**Data Preparation:**

Positive samples are pictures that show the subject of interest, such faces or cars.

Negative Samples: Photos devoid of the subject.

**Calculate HOG Features:**

To extract feature vectors from both positive and negative samples, use the HOG method.

**Labelling:**

Give the feature vectors labels (e.g., green for positive, red for negative).

**SVM training:**

To use the SVM algorithm, feed the labelled feature vectors into it.

SVM learns to locate the best hyperplane in the high-dimensional HOG feature space for separating positive and negative data.

**Phase of Testing:**

Calculate the Test Image's HOG Features: To extract feature vectors from the test image, use the HOG technique.

**Sort utilising SVM:**

Classify the feature vector that was taken out of the test image using the SVM model that has been trained.

Localization of Objects:

Additional steps (such non-maximum suppression) can be used to fine-tune and localise the detected object inside the image once the SVM predicts a positive detection.

**Principal Benefits of Using SVM with HOG in Object Detection:**

Robustness to Variations: HOG is resistant to changes in lighting, orientation, and scale since it records details about the shape and look of objects. Discriminative Power: SVM improves the HOG features' discriminative power for precise object detection by identifying the best decision boundary. Versatility: A lot of different items, including faces, pedestrians, and cars, can be detected in photos using this combination. To sum up, HOG and SVM work together to effectively recognise objects in images by utilising the advantages of feature extraction and machine learning.

**CHAPTER 3**

# SYSTEM DESCRIPTION

## 3.1 Problem Statement

Finding a vacant parking space in a congested area or a large parking lot, specially, in peak hours, is always time consuming and frustrating to drivers. It is common for drivers to keep circling a parking lot and look for a vacant parking space. To minimize hassle and inconvenience to the drivers, many parking guidance systems have been developed over the past decade, where the system provides accurate, real-time car park space availability to the drivers looking for parking spaces and then guides them to the available spaces by dynamically updated guide signs. The current parking guidance systems obtain the availability of parking spaces using the sensors installed across the whole parking lot. However, deploying sensors in a large parking lot can be very expensive. Furthermore, the sensors can become inaccurate and would stop functioning easily when time passes. Therefore, it is highly desired to have a reliable and cost effective way to track available parking spaces and guide drivers to the available parking spaces. Besides searching for available parking spaces. In urban areas, the inefficiency of traditional vehicle identification systems, exacerbated by increasing vehicle numbers and limited parking infrastructure, results in congestion and wasted time. The core issue is the challenge of obstacle detection for vehicles. The limitations of traditional obstacle detection systems, such as reliance on simple sensors or lack of advanced decision-making capabilities. These limitations make it challenging for vehicles to detect

and respond to complex obstacles effectively. The complexity of modern road environments, including factors like varying weather conditions, diverse road users (pedestrians, cyclists), and dynamic traffic situations. These factors amplify the difficulty of

obstacle detection.

## 

## 3.2 Project Description

Our AI-based parking system leverages the power of Support Vector Machines (SVM) and Histogram of Oriented Gradients (HOG) to create an intelligent and efficient solution for parking space management.

Support Vector Machine (SVM) and Histogram of Oriented Gradients (HOG) are both fundamental components in computer vision and image processing. SVM is a supervised machine learning algorithm used for classification and regression tasks. It finds a hyperplane that best separates different classes in a feature space. This hyperplane is chosen to maximize the margin between classes. SVM learns from labeled training data, and during prediction, it classifies new data points into different classes based on their features. In the context of an AI-based parking system, SVM can be trained to classify vehicles based on features extracted from images or video frames. HOG is a feature descriptor used for object detection in computer vision. Functionality: It captures the distribution of intensity gradients in localized regions of an image.

HOG divides an image into small, connected regions, computes the gradient magnitude and orientation in each region, and then represents this information as a histogram. In the context of an AI-based parking system, HOG can be employed for detecting vehicles by identifying the distinctive gradient patterns associated with different vehicle shapes.

SVM and HOG are often used together for object detection tasks, such as vehicle detection in images or video streams. In the parking system, HOG can extract relevant features from images, and SVM can classify these features to determine whether a given region contains a vehicle or not. For example, in an AI-based parking system, SVM-HOG model training involves feeding the SVM with features extracted by HOG from a dataset of images containing both vehicle and non-vehicle instances.

Employ HOG features to capture the shape and structure of vehicles effectively. Train an SVM classifier using labeled datasets to distinguish between different vehicle types.

Utilize the SVM-HOG model to intelligently allocate parking spaces based on the type and size of detected vehicles. Implement dynamic adjustments to optimize parking resource utilization. Develop an intuitive user interface, possibly through a mobile application or web portal, providing real-time updates on parking space availability. Enable users to easily locate and reserve parking spaces, enhancing overall user experience.

Conduct comprehensive testing under diverse scenarios, including different lighting and weather conditions, to validate the accuracy and efficiency of the SVM-HOG model. Iteratively refine the model based on testing results to enhance performance. Provide detailed documentation covering system architecture, SVM-HOG model training, algorithm implementation, and user instructions.

## 3.3 Project Goals

Artificial Intelligence (AI) has emerged as a transformative technology in addressing this challenge. AI, particularly machine learning and computer vision techniques, can enable vehicles to "see" and interpret their surroundings, thereby improving their ability to detect and respond to obstacles effectively. If the driver have the up-to-date knowledge about the traffic situation like information about the free parking place near the destination area it would be then great benefit.

Design the system to handle varying parking demands, scaling seamlessly to cover larger parking areas without compromising performance. Ensure that computational resources are efficiently utilized to maintain responsiveness. Implement additional security measures, such as license plate recognition, to enhance overall safety and monitoring of the parking area. Integrate with existing security systems if applicable. Explore energy-efficient algorithms and hardware configurations to minimize power consumption while maintaining optimal system performance. Consider implementing sleep modes during low-demand periods to conserve energy.

Ensure that documentation is accessible and comprehensive for both development and maintenance purposes. Evaluate and minimize the environmental impact of the system, considering factors such as energy consumption, waste generation, and overall sustainability. By integrating these components, our AI-based parking system aims to deliver a smart, adaptive, and user-friendly solution that optimizes parking space utilization and enhances the overall parking experience.

## CHAPTER 4

## PROPOSED SYSTEM

## 4.1 Description of the proposed systems

## The suggested system uses Histogram of Oriented Gradients (HOG) as the feature descriptor and Support Vector Machines (SVM) as a machine learning technique to train the model. To guarantee the model's performance in practical situations, a custom dataset comprising photos with and without parking is created as part of the system's training process. The carefully chosen dataset adds to the model's resilience by representing a range of parking situations and environmental component fluctuations

## 4.2 Algorithm

It is important to employ HOG as a feature descriptor because it extracts gradient and spatial information from images, which helps the SVM identify patterns and generate precise predictions. By including SVM, a potent and popular classification method, the model is better able to distinguish between parking and non-parking situations with a high degree of accuracy.

The remarkable accuracy of 94% highlights the efficacy of the suggested strategy. This high accuracy rate shows that the algorithm can correctly recognise parking scenarios and generalise well to new data. In practical applications, such a dependable performance is essential, guaranteeing that the system may be confidently deployed in a variety of real-world scenarios.

Moreover, the accuracy rate of the system indicates that it can play a major role in resolving parking-related issues by providing a dependable and effective solution. This degree of precision is especially useful in situations where exact parking space identification is crucial for maximising space utilisation and improving overall efficiency, such as automated parking systems, traffic management, and urban planning.

In conclusion, a robust and precise parking recognition system has been produced by the integration of SVM, HOG feature descriptor, and the development of a carefully chosen dataset. The system's robustness and potential for useful application in a variety of contexts are demonstrated by the 94% accuracy attained, which positions it as a viable solution for parking-related problems.

**Steps involved in the process:**

**Data Loading and Preprocessing:** The script starts by importing the scikit-image, numpy, pandas, and scikit-learn libraries that are required. The categories ('yes' and 'no') are then defined, and empty arrays for the input data (flat\_data\_arr) and output labels (target\_arr) are initialised. The script then loops over each category, flattens the images, and appends them to the input array after reading and resizing the photos using scikit-image's imread and resize functions. To the output array, corresponding labels are appended (0 for "no" and 1 for "yes").

**Making a DataFrame:** A pandas DataFrame (df) is created from the input and output arrays. Each pixel in the flattened photos is represented by a column in the DataFrame, and an extra column called "Target" has the labels for each pixel. Additional processing and analysis are performed on this DataFrame.

**Data Splitting:** The DataFrame is used to extract the input data (x) and output data (y). Next, the train\_test\_split function from scikit-learn is used to divide the data into training and testing sets. To guarantee a proportionate representation of classes in both the training and testing sets, the split is stratified according to the output labels.

**Define the Model and Parameter Grid:** The script creates a Support Vector Classifier (svc) that can produce probabilities for each class. It uses GridSearchCV to set up a parameter grid (param\_grid) for hyperparameter tweaking. The regularisation parameter ('C'), kernel type ('kernel'), and kernel coefficient ('gamma') are among the parameters.

**Training of the GridSearchCV Model:**

To determine the ideal hyperparameters for the SVM model, the script makes use of GridSearchCV. This entails varying the hyperparameter combinations listed in the parameter grid in order to train the model. On the basis of cross-validated performance, the optimal combination is chosen.

**Testing and Accuracy Calculation:** Next, using the testing set as a test set, the trained model is tested and predictions are generated via the predict technique. Using the scikit-learn accuracy\_score function, the model's accuracy is determined and shown on the console.

**Image Prediction:** Lastly, the script loads an example image from a path, uses matplotlib to show it, resizes it to fit the input size of the model, and then makes a prediction. Each class's anticipated probability are printed, and the anticipated class is shown.

**CHAPTER 4**

**LITERATURE SURVEY**

**4.1 Analysis of existing systems**

* Zichen Fan, Detecting and Classifying Phishing Websites by Machine Learning in 3rd International Conference on Applied Machine Learning (ICAML). Depicts a method of detecting phishing websites with joint features. SVM and bayes methods are mixed in training classifiers. After going through training, the classifier can detect 1000 websites per second.
* Lu, Rongxing and Lin, Xiaodong and Zhu, Haojin and Shen, Xuemin SPARK: A new VANET-based smart parking scheme for large parking lots Proceedings - IEEE INFOCOM}, doi ={10.1109/INFCOM.2009.5062057} Searching for a vacant parking

space in a congested area or a large parking lot and preventing auto theft are major concerns to our daily lives. In this paper, we propose a new smart parking scheme for large parking lots through vehicular communication.

* Abir Mchergui and Tarek Moulahi and Sherali Zeadally, Survey on Artificial Intelligence (AI) techniques for Vehicular Ad-hoc Networks (VANETs) Vehicular Communications, Advances in communications, smart transportation systems, and computer systems have recently opened up vast possibilities of intelligent solutions for traffic safety, convenience, and effectiveness. Artificial Intelligence (AI) is currently being used in various application domains because of its strong potential to help enhance conventional data-driven methods. In the area of Vehicular Ad hoc NETworks (VANETs) data is frequently collected from various sources.
* {Ma, Chunmei and Zhu, Jinqi and Liu, Ming and Zhao, Hui and Liu, Nianbo and Zou, Xinyu Parking Edge Computing: Parked-Vehicle-Assiste d Task Offloading for Urban VANETs IEEE Internet of Things Journal, Vehicular edge computing has been a

promising paradigm to offer low latency and high reliability vehicular services for users.

Nevertheless, for compute-intensive vehicle applications, most previous research cannot perform them efficiently due to both the inadequate of infrastructure construction and the computing resource bottleneck of the edge server.

* Shweta Singh, M.P. Singh, Ramprakash Pandey, Phishing Detection from URLs Using Deep Learning Approach in 5th International Conference on Computing Communication and Security (ICCCS), Proposed a phishing detection system using deep learning techniques to prevent phishing attacks. The system works on URLs by applying a convolutional neural network (CNN) to detect the phishing webpage.
* Rehman, Mujeeb Ur and Shah, Munam Ali and Khan, Muhammad and Ahmad, Shaheed, A VANET based Smart Car Parking System to Minimize Searching Time, Fuel Consumption and CO2 Emission in 2018 24th International Conference on Automation and Computing (ICAC). In this paper, a Vehicular Ad-hoc Network(VANET) based routing algorithm for SPS is proposed called Updating Block Route Algorithm (UBRA).

**4.2. Drawbacks in the existing system:**

While VANETs offer promising solutions for parking management, there are several drawbacks and challenges associated with existing parking systems based on VANET technology: Achieving widespread adoption and ensuring coverage across diverse urban and rural areas remains a challenge. Ensuring the reliability of communication in real-time parking updates is crucial for accurate information dissemination to drivers.

Striking a balance between providing valuable information to users and protecting individual privacy is a challenge that needs careful consideration. Implementing robust security measures to protect the integrity and confidentiality of parking-related data is essential. The cost and effort associated with installing and maintaining this infrastructure may pose challenges, particularly in less developed or remote areas.

Designing systems that can efficiently handle a large number of vehicles and ensure timely updates in high-density scenarios requires careful planning and optimization. Coordinating the machine learning algorithm SVM with HOG with the traditional parking systems, local regulations, and urban planning requires collaboration between various stakeholders. Designing energy-efficient communication protocols and optimizing data transmission to minimize the impact on vehicle energy resources is a key challenge.

**CHAPTER 5**

**RESULT AND ANALYSIS**

**5.1 Snapshots of the output for the HOG feature descriptor:**

These are the results:

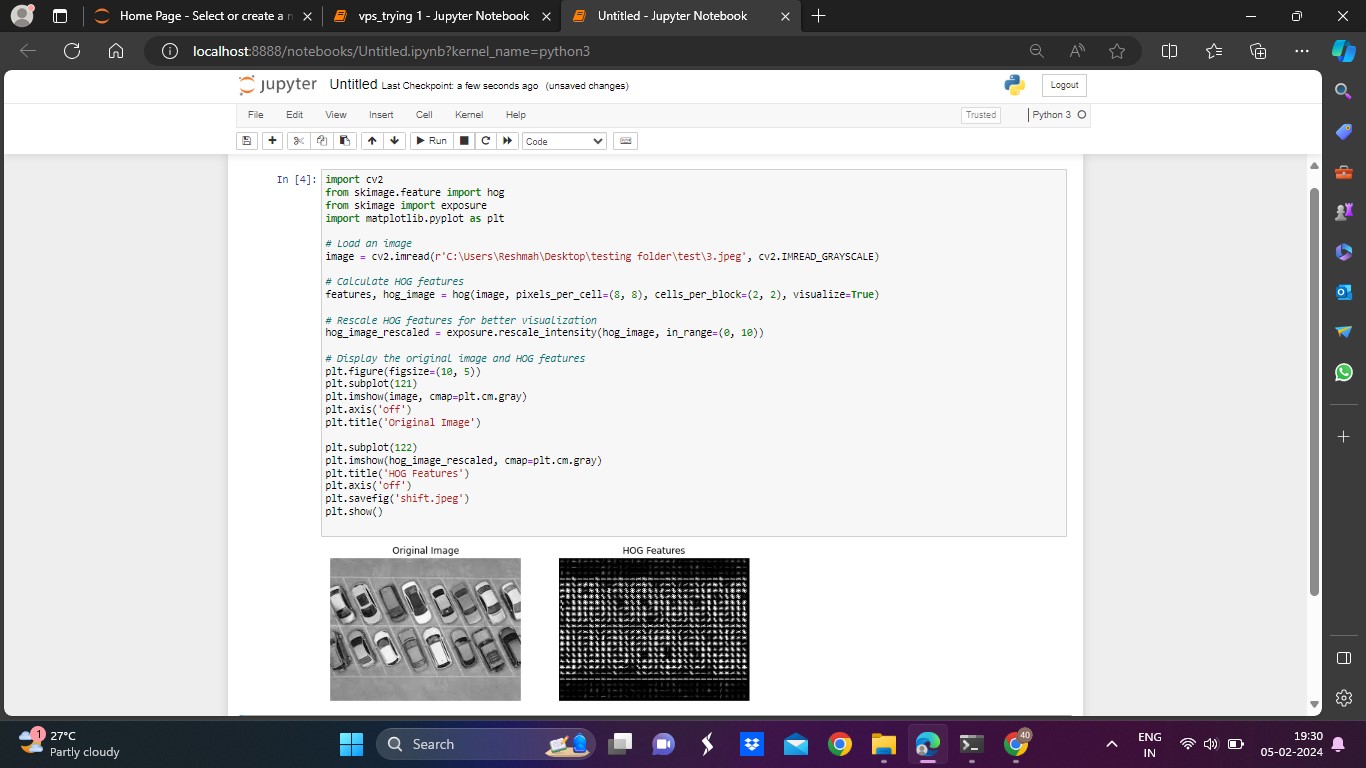


Figure 5.1 HOG for no parking image

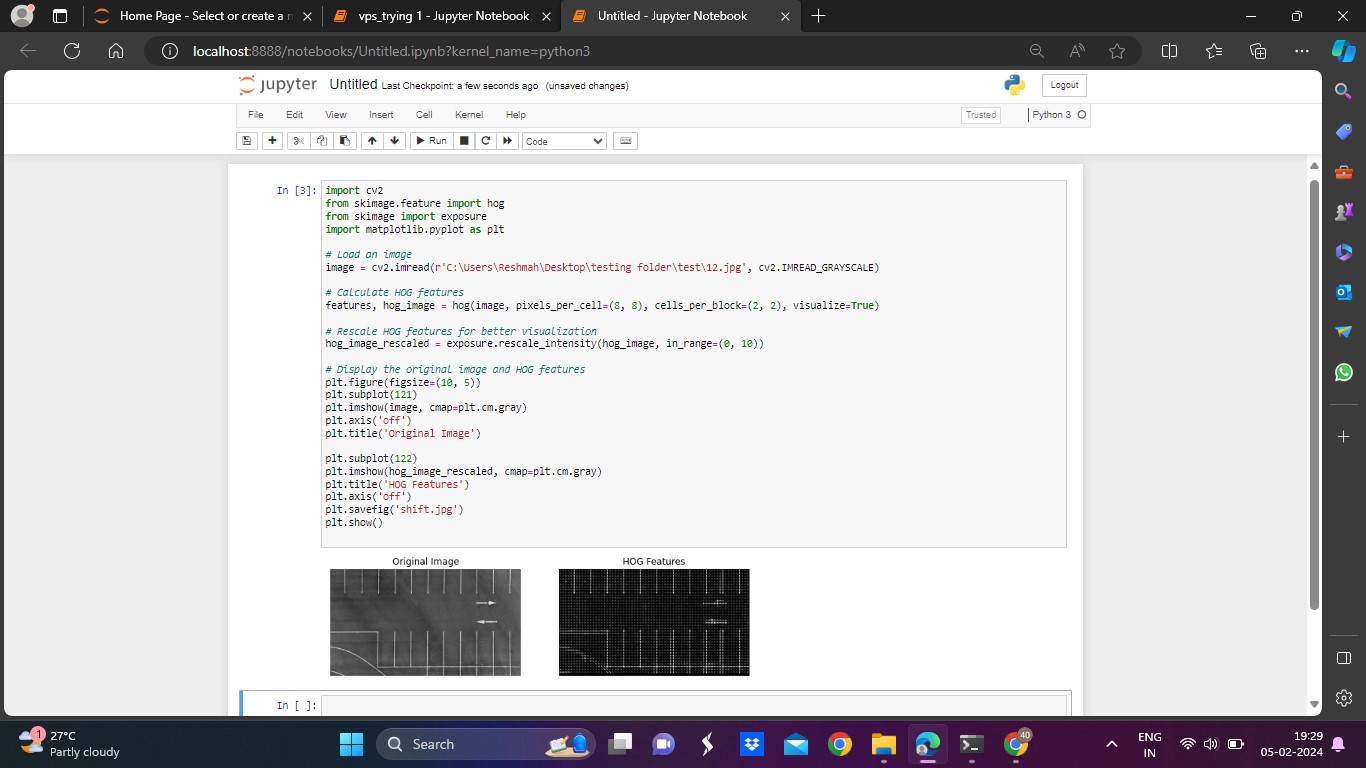


Figure 5.1.a. HOG for parking image

**5.2 Snapshots of the output for SVM model building and training:**

These are the results:

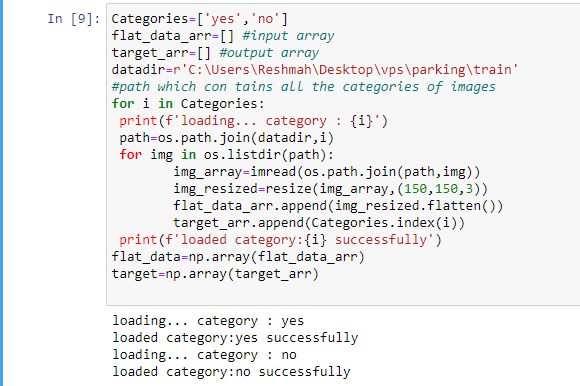


Figure 5.2.a. loading the necessary datasets

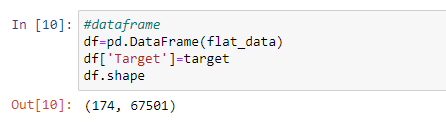


Figure 5.2.b. designing the dataframe

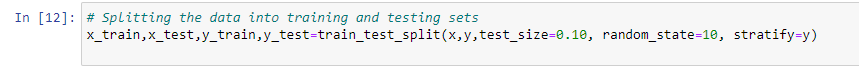


Figure 5.2.c. Splitting the dataset into training and testing sets

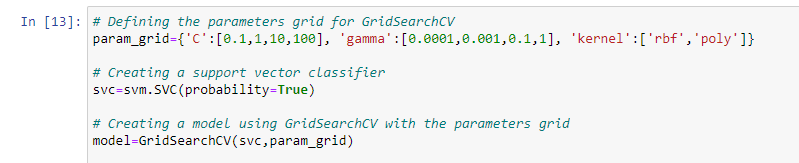


Figure 5.2.d. SVM classifier

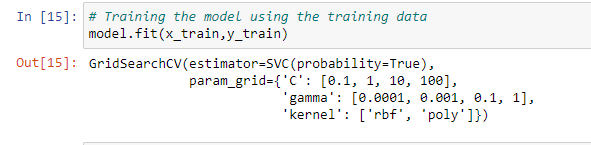


Figure 5.2.e. training the dataset with SVM

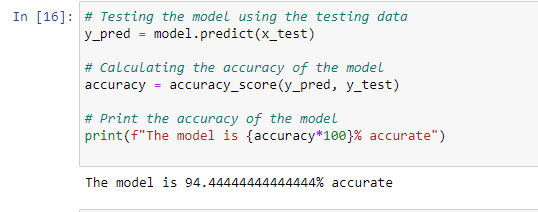


Figure 5.2.f. calculating the accuracy

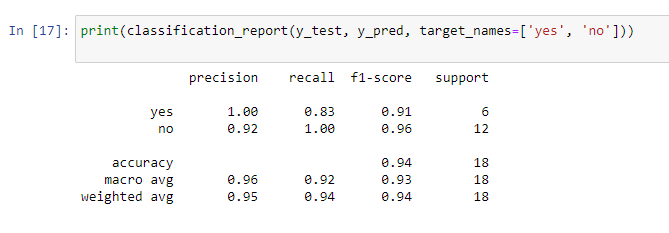


Figure 5.2.g. generating the classification report



Figure 5.2.f. Output for the no parking slot

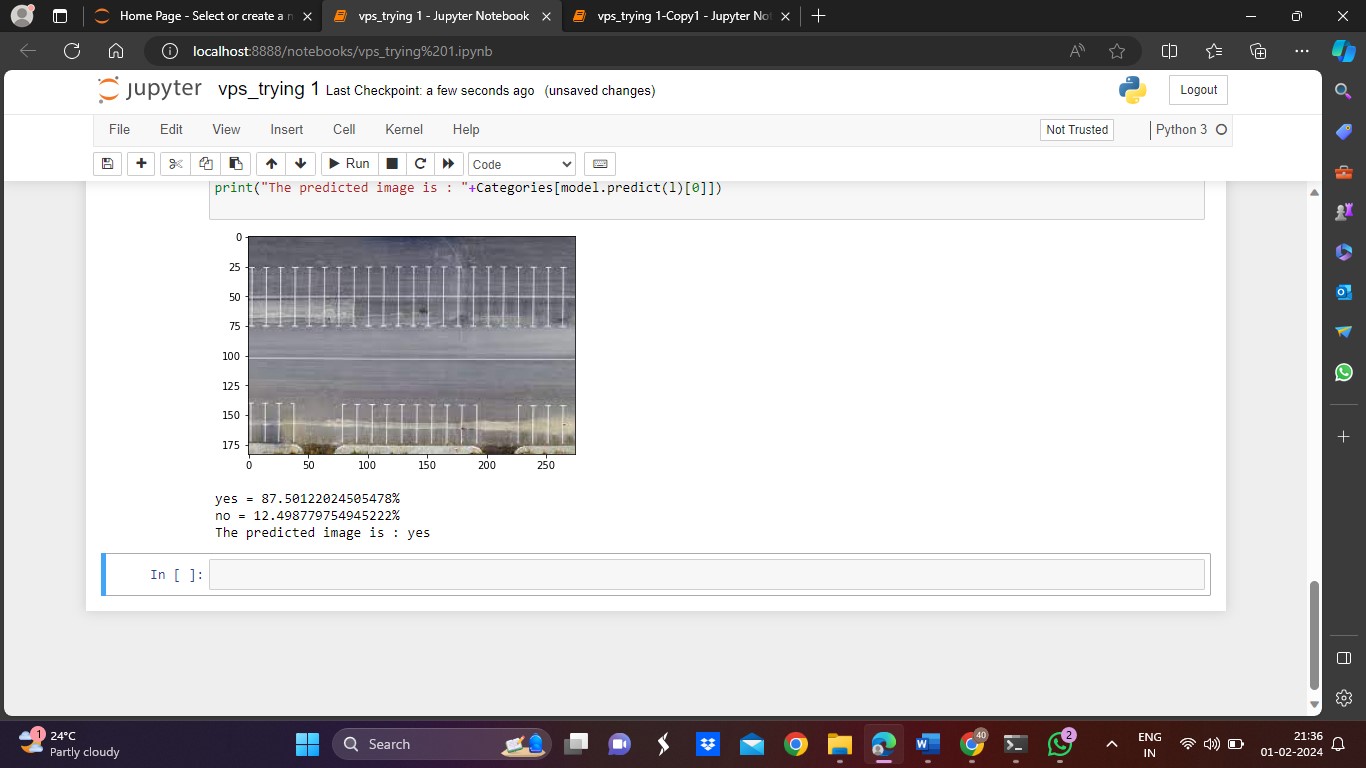


Figure 5.2.i. Output for the no parking slot

## CHAPTER 6

## CONCLUSION AND FUTURE WORKS

**6.1 Conclusion**

To conclude, this project is implemented successfully and in the first hand the analysis of the given data sets, the integration of SVM and HOG algorithms in a vehicle parking system represents a promising advancement in smart transportation and urban management. This synergy harnesses the power of real-time communication among vehicles and the sophisticated pattern recognition capabilities of SVM for effective parking space management. The developed system facilitates seamless navigation for drivers, optimizes parking space utilization, and contributes to overall traffic efficiency.

## 6.2 Future work

## HOG may have some limitations. The future works may include energy efficiency implementation and by considering more efficient feature extraction algorithm and implementing them using the VANET, by considering the power constraints of the vehicles. Implementing edge computing to process data makes the decision making easier and improve the accuracy of the model.

## In the future, CNN might be used instead of SVM for better outcomes and efficiency. Finding the hyperplane that maximizes the difference between the expected and actual values is how this machine learning technique operates. Both linear and nonlinear interactions between the dependent and independent variables can be modelled using it.

## APPENDICES

import pandas as pd

import os

from skimage.transform import resize

from skimage.io import imread

import numpy as np

import matplotlib.pyplot as plt

from sklearn import svm

from sklearn.model\_selection import GridSearchCV

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

Categories=['yes','no']

flat\_data\_arr=[] #input array

target\_arr=[] #output array

datadir=r'C:\Users\Reshmah\Desktop\vps\parking\train'

#path which con tains all the categories of images

for i in Categories:

print(f'loading... category : {i}')

path=os.path.join(datadir,i)

for img in os.listdir(path):

img\_array=imread(os.path.join(path,img))

img\_resized=resize(img\_array,(150,150,3))

flat\_data\_arr.append(img\_resized.flatten())

target\_arr.append(Categories.index(i))

print(f'loaded category:{i} successfully')

flat\_data=np.array(flat\_data\_arr)

target=np.array(target\_arr)

#dataframe

df=pd.DataFrame(flat\_data)

df['Target']=target

df.shape

#input data

x=df.iloc[:,:-1]

#output data

y=df.iloc[:,-1]

# Splitting the data into training and testing sets

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.10, random\_state=10, stratify=y)

# Defining the parameters grid for GridSearchCV

param\_grid={'C':[0.1,1,10,100], 'gamma':[0.0001,0.001,0.1,1], 'kernel':['rbf','poly']}

# Creating a support vector classifier

svc=svm.SVC(probability=True)

# Creating a model using GridSearchCV with the parameters grid

model=GridSearchCV(svc,param\_grid)

# Training the model using the training data

model.fit(x\_train,y\_train)

# Testing the model using the testing data

y\_pred = model.predict(x\_test)

# Calculating the accuracy of the model

accuracy = accuracy\_score(y\_pred, y\_test)

# Print the accuracy of the model

print(f"The model is {accuracy\*100}% accurate")

print(classification\_report(y\_test, y\_pred, target\_names=['yes', 'no']))

path=r'C:\Users\Reshmah\Desktop\testing folder\train\yes\jpeg'

img=imread(path)

plt.imshow(img)

plt.show()

img\_resize=resize(img,(150,150,3))

l=[img\_resize.flatten()]

probability=model.predict\_proba(l)

for ind,val in enumerate(Categories):

print(f'{val} = {probability[0][ind]\*100}%')

print("The predicted image is : "+Categories[model.predict(l)[0]])

**REFERENCES**

[1] S. Hossain and X. Lin, "Short-range and Long-range Obstacle Detection Method for a Delivery Robot Based on Multi-sensor Fusion," 2023 IEEE Transportation

Electrification Conference & Expo (ITEC), Detroit, MI, USA, 2023, pp. 1-5, doi:

10.1109/ITEC55900.2023.10186935

[2] C. Liu, L. Zhai and X. Zhang, "Research on local real-time obstacle avoidance path

planning of unmanned vehicle based on improved artificial potential field method," 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI),

Nanjing, China, 2022, pp. 1-6, doi: 10.1109/CVCI56766.2022.9964763

[3] A. Kumari, H. R. Daksh, A. Utsav and A. Abhishek, "Self-Automated Car with

Obstacles Detection," 2022 13th International Conference on Computing

Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022

[4] Tanti, Hardik & Kasodariya, Pratik & Patel, Shikha & Rangrej, Dhaval. (2020).

Smart Parking System based on IOT. International Journal of Engineering Research

[5] Pomaji, Amol & Boinwad, Suraj & Wankhede, Shrikant & Singh, Pushpendra &

Dhakulkar, Bhagyashree. (2019). Smart Parking Management System. International

Journal of Computer Sciences and Engineering.

[6]Addie Lawrence ;“Data Analytics and Machine Learning: Let’s Talk Basics”-explains about the machine learning that invokes natural language is also targeted toward business users who can perform the analysis themselves (a development known as [augmented analytics](https://www.answerrocket.com/augmented-analytics/))

[7] Q. Wang, "Support Vector Machine Algorithm in Machine Learning," 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2022, pp. 750-756, doi: 10.1109/ICAICA54878.2022.9844516. keywords: {Training; Machine learning algorithms; Support vector machine classification; Transforms;Complexity theory; Pattern recognition; Risk management; Machine Learning; Statistical Learning;Classification;Support Vector Machine},

[8] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Schoellkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428. keywords: {Support vector machines; Machine learning; Algorithm design and analysis; Pattern recognition; Neural networks; Training data; Polynomials; Kernel; Character recognition; Web pages}

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177. keywords: {Histograms;Humans;Robustness;Object recognition;Support vector machines;Object detection;Testing;Image edge detection;High performance computing;Image databases}

[10] M. Kitayama and H. Kiya, "HOG feature extraction from encrypted images for privacy-preserving machine learning," 2019 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), Bangkok, Thailand, 2019, pp. 80-82, doi: 10.1109/ICCE-Asia46551.2019.8942217. keywords: {encryption-then-compression block-based encryption;histgram of oriented gradients}

[11] Y. Yamauchi, C. Matsushima, T. Yamashita and H. Fujiyoshi, "Relational HOG feature with wild-card for object detection," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 2011, pp. 1785-1792, doi: 10.1109/ICCVW.2011.6130465. keywords: {Feature extraction;Training;Histograms;Object detection;Probability density function;Memory management;Vehicles}

[12] S. Bougharriou, F. Hamdaoui and A. Mtibaa, "Linear SVM classifier based HOG car detection," 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, 2017, pp. 241-245, doi: 10.1109/STA.2017.8314922. keywords: {Support vector machines;Automobiles;Vehicledetection;Histograms;Feature extraction;Classification algorithms;Advanced Driver Assistance Systems;vehicle detection;histogram of oriented gradients features descriptor;support vector machine}

[13] X. Li and X. Guo, "A HOG Feature and SVM Based Method for Forward Vehicle Detection with Single Camera," 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 2013, pp. 263-266, doi: 10.1109/IHMSC.2013.69. keywords: {Vehicles;Feature extraction;Support vector machines;Vehicle detection;Histograms;Training;Safety;Histogram of gradient;Support vector machine;Vehicle classifier;Forward vehicle detection;Automotive safety driver assistance system}

[14] M. Kitayama and H. Kiya, "Generation of Gradient-Preserving Images allowing HOG Feature Extraction," 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Penghu, Taiwan, 2021, pp. 1-2, doi: 10.1109/ICCE-TW52618.2021.9603248. keywords: {Support vector machines;Machine learning algorithms;Face recognition;Conferences;Machine learning;Feature extraction}

[15] Xiaowu Sun, Lizhen Liu, Hanshi Wang, Wei Song and Jingli Lu, "Image classification via support vector machine," 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), Harbin, China, 2015, pp. 485-489, doi: 10.1109/ICCSNT.2015.7490795.

[16] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428. keywords: {Support vector machines;Machine learning;Algorithm design and analysis;Pattern recognition;Neural networks;Training data;Polynomials;Kernel;Character recognition;Web pages}

[17] Q. Wang, "Support Vector Machine Algorithm in Machine Learning," 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2022, pp. 750-756, doi:10.1109/ICAICA54878.2022.9844516. keywords: {Training;Machine learning algorithms;Support vector machine classification;Transforms;Complexity theory;Pattern recognition;Risk management;Machine Learning;Statistical Learning;Classification;Support Vector Machine}

[18] S. Ghosh, A. Dasgupta and A. Swetapadma, "A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification," 2019 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 2019, pp. 24-28, doi: 10.1109/ISS1.2019.8908018. keywords: {Support vector machines;Kernel;Classification algorithms;Training data;Supervised learning;Machine learning algorithms;Training;Pattern analysis;SVM;Classification;Machine Learning}

[19] T. -t. Dai and Y. -s. Dong, "Introduction of SVM Related Theory and Its Application Research," 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Shenzhen, China, 2020, pp. 230-233, doi: 10.1109/AEMCSE50948.2020.00056. keywords: {Support vector machine;statistical learning theory;kernel function;multi-class classification}

[20] Yujun Yang, Jianping Li and Yimei Yang, "The research of the fast SVM classifier method," 2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 2015, pp. 121-124, doi: 10.1109/ICCWAMTIP.2015.7493959. keywords: {Training;Support vector machine classification;Kernel;Testing;Matrix decomposition;Sensitivity;SVM;Classifier;Framework;Support vector machine},