CAR OBJECT DETECTION

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

by

P CHAITHRA SRI	**(2103A52027)**
U REETHU VARMA	**(2103A52037)**

Under the guidance of

Mr. Ramesh Dadi

Assistant Professor, Department of CSE.

SR UNIVERSITY

Department of Computer Science and Artificial Intelligence

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that project entitled **"CAR OBJECT DETECTION"** is the bonafied work carried out by **Chaithra sri , Reethu Varma** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2022-2023 under our guidance and Supervision.

Mr. D.Ramesh
Asst. Professor,
(CSE), S R University,
Ananthasagar, Warangal.

Dr. M.Sheshikala
Assoc. Prof. & HOD
S R University,
Ananthasagar, Warangal.

ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr. D.Ramesh, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M.Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

Car object detection is a critical task in the field of computer vision, with applications in autonomous driving, traffic monitoring, and surveillance systems. The goal is to develop algorithms or models that can accurately detect and locate vehicles within images or video frames. This involves addressing challenges such as variation in lighting and weather conditions, occlusion, scale variation, class imbalance, and real-time performance requirements. Advanced computer vision techniques and machine learning algorithms, such as KNN,Naive Bayes, are commonly used for car object detection. The objectives of car object detection are to improve safety, optimize traffic flow, enhance surveillance systems, provide valuable insights, and develop advanced computer vision techniques. Despite the challenges, car object detection is a rapidly advancing field with the potential to revolutionize transportation systems and improve the safety and efficiency of our roads.

Table of Contents

List of figures

INTRODUCTION

1.1 OVERVIEW

Object detection is a computer vision technique used to identify objects of interest within an image or video frame. It involves the use of machine learning algorithms to recognize and locate objects within an image or video frame. Object detection is a critical component of many applications, including self-driving cars, surveillance systems, and image and video analysis.

In the context of car object detection, the goal is to identify and locate vehicles within an image or video frame. This is important for a variety of reasons, including improving the safety of autonomous vehicles, optimizing traffic flow, and enhancing the accuracy of surveillance systems.

The techniques typically involve training a model on a large dataset of labeled images, which allows the model to learn to recognize and localize different types of vehicles under varying conditions, such as changes in lighting, weather, and camera angle.

Once trained, the model can be used to analyze new images or video frames in real-time, identifying the presence, location, and type of vehicles within the scene. This information can then be used to inform a variety of applications, such as autonomous driving, traffic management, and surveillance.

1.2 PROBLEM STATEMENT

In recent years many researchers have worked on video cameras which are considered as a sensor device for capturing and recognizing moving vehicle. Video based systems can capture a large variety of information which is less expensive to install and maintain cameras. It is not easy for a single moving camera to quickly capture information. The main difficulty in vehicle video is the variation of environment and the headlights are important features for initializing and tracking vehicle at night time. It is capable of detecting vehicles and measures traffic parameters like speed, count, incidents etc.

Vehicle tracking is the process of locating a moving vehicle using a camera. Capture vehicle in video sequence from surveillance camera is demanding application to improve tracking performance. This technology is increasing the number of applications such as traffic control, traffic monitoring, traffic flow, security etc. The estimated cost using this technology will be very less. Video and image processing has been used for traffic surveillance, analysis and monitoring of traffic conditions in many cities and urban areas. Various methods for speed estimation are proposed in recent years. All approaches attempt to increase accuracy and decrease cost of hardware implementation. The aim is to build an automatic system that can accurately localize and track the speed of any vehicles that appear in aerial video frames.

1.3 EXISTING SYSTEM

There are many existing systems for car object detection that use various techniques and algorithms. Here are some examples of such systems:

1. OpenCV: OpenCV is an open-source computer vision library that provides many functions for object detection, including car detection. It provides pre-trained models for car detection, such as Haar cascades, HOG+SVM, and deep learning models.
2. YOLOv4: YOLOv4 is a popular object detection algorithm that achieves state-of-the-art performance on various object detection benchmarks, including car detection. It is a single-stage object detector that uses a deep CNN to predict object classes and bounding boxes directly from the input image.
3. TensorFlow Object Detection API: TensorFlow Object Detection API is a framework built on top of TensorFlow that provides many pre-trained models for object detection, including car detection. It includes models such as Faster R-CNN, SSD, and RetinaNet.

1.4 PROPOSED SYSTEM

The proposed work predicts smoke by exploring the above mentioned four classification algorithms and does performance analysis. The objective of this study is to effectively predict if there is car or not. The data is fed into model which predicts the probability of car in the image.. Figure shows the entire process involved.

Training Data→ Classification techniques

Data Extraction→Classification→ →Result→End

Testing Data→ Test the model

1.5. OBJECTIVES

The objectives of car object detection are to:

1. Improve the safety of autonomous vehicles: Accurately detecting other vehicles on the road is critical for the safe operation of autonomous vehicles. Car object detection can help prevent collisions by identifying other vehicles and predicting their movements.
2. Enhance surveillance systems: Car object detection can be used in surveillance systems to detect and track vehicles in real-time, improving the accuracy and effectiveness of the system.
3. Provide valuable insights: Car object detection can provide valuable insights into traffic patterns and behavior, helping to inform decision-making in transportation planning and policy.
4. Detection of speed
5. Autonomous Driving.
   ex: Self-driving cars.
6. Image retrieval.
7. Video surveillance.
8. Traffic rules violation.

Overall, the objectives of car object detection are to improve safety, efficiency, and understanding of traffic and transportation systems.

1.6 ARCHITECTURE

The architecture of the proposed system is as displayed below

DATA EXTRACTION → EXPLORING DATA ANALYSIS → DATA PRE-PROCESSING → TRANING MODEL → TESTING MODEL → BUILDING THE MODELS → RESULT → CAR IS PRESENT OR NOT →END.

LITERATURE SURVEY

1. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" by Shaoqing Ren et al. (2016)

This paper presents the Faster R-CNN algorithm, which achieves state-of-the-art performance on various object detection benchmarks, including car detection. The algorithm combines a region proposal network (RPN) with a deep convolutional neural network (CNN) to detect objects in real-time.

2. "You Only Look Once: Unified, Real-Time Object Detection" by Joseph Redmon et al. (2016)

This paper introduces the YOLO algorithm, which is a single-stage object detector that achieves real-time performance while maintaining high accuracy. The algorithm uses a CNN to predict object classes and bounding boxes directly from the input image.

3. "Single Shot MultiBox Detector" by Wei Liu et al. (2016)

This paper proposes the SSD algorithm, which is a one-shot object detector that achieves high accuracy with a single pass through a deep CNN. The algorithm uses a set of default boxes of different scales and aspect ratios to predict object locations and classes.

4. "Deep Residual Learning for Image Recognition" by Kaiming He et al. (2016)

This paper introduces the ResNet architecture, which is a deep CNN that uses residual connections to enable the training of very deep networks. The ResNet architecture has been used as a backbone network for various object detection algorithms, including those for car detection.

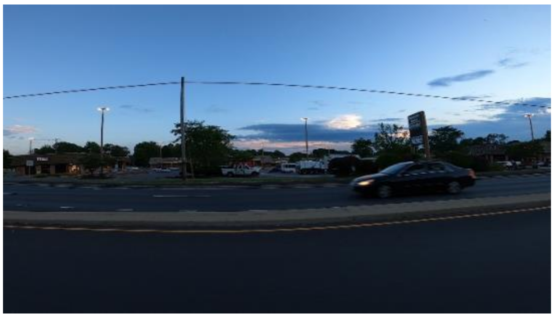
5. "End-to-end learning of object detection and segmentation from multimodal input" by Guosheng Lin et al. (2018)

DATA PRE-PROCESSING

3.1 DATASET DESCRIPTION

About the dataset:

- The dataset consists of 175 testing images.
- Dataset has 1001 training images.
- The dataset contains media of cars in all views, and your job is to create an algorithm to detect them.
- The x is the array of images and y is the output whether the car is present or not in the image.



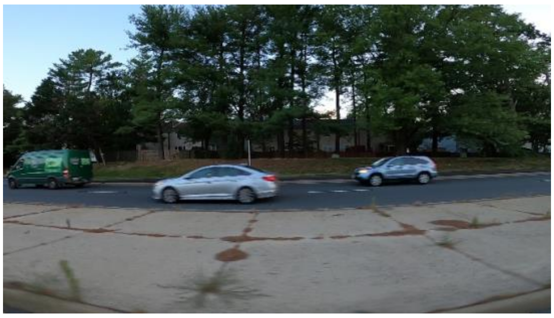


Image	bounds
vid_5_26560.jpg	0.0 0.0 1.0 1.0 0.0 0.0 1.0 1.0
vid_5_26580.jpg	0.0 0.0 1.0 1.0 0.0 0.0 1.0 1.0
vid_5_26600.jpg	0.0 0.0 1.0 1.0
vid_5_26620.jpg	0.0 0.0 1.0 1.0
vid_5_26640.jpg	0.0 0.0 1.0 1.0
vid_5_26660.jpg	0.0 0.0 1.0 1.0
vid_5_26680.jpg	0.0 0.0 1.0 1.0
vid_5_26700.jpg	0.0 0.0 1.0 1.0
vid_5_26720.jpg	0.0 0.0 1.0 1.0
vid_5_26740.jpg	0.0 0.0 1.0 1.0
vid_5_26760.jpg	0.0 0.0 1.0 1.0
vid_5_26780.jpg	0.0 0.0 1.0 1.0

……………………….

vid_5_30820.jpg	0.0 0.0 1.0 1.0
vid_5_30840.jpg	0.0 0.0 1.0 1.0
vid_5_30860.jpg	0.0 0.0 1.0 1.0
vid_5_30920.jpg	0.0 0.0 1.0 1.0
vid_5_30940.jpg	0.0 0.0 1.0 1.0
vid_5_31020.jpg	0.0 0.0 1.0 1.0
vid_5_31040.jpg	0.0 0.0 1.0 1.0
vid_5_31560.jpg	0.0 0.0 1.0 1.0
vid_5_31600.jpg	0.0 0.0 1.0 1.0
vid_5_31620.jpg	0.0 0.0 1.0 1.0
vid_5_31700.jpg	0.0 0.0 1.0 1.0
vid_5_31720.jpg	0.0 0.0 1.0 1.0

3.2 DATA CLEANING

Data cleaning is an important step in any machine learning task, including car object detection. It involves identifying and correcting errors and inconsistencies in the dataset to ensure that the data is accurate, complete, and ready for analysis.

Here are some common data cleaning steps that can be applied to car object detection datasets:

1. Remove duplicates.

2. Remove outliers
3. Check for missing data
4. Normalize the data:
5. Check for class imbalance.

Overall, data cleaning is an important step in preparing the dataset for car object detection. By removing errors and inconsistencies, and ensuring that the data is accurate and complete, the model can be trained on high-quality data that will lead to more accurate and reliable predictions.

3.4 DATA VISUALISATION

The car object detection data set contains images with and without car in them in array form i.e x the output contains symptoms of as features and whether the car is present in the image or not i.e is y.

METHODOLOGY

4.1 PROCEDURE TO SOLVE THE GIVEN PROBLEM

Here in this project we can use these machine learning algorithms:

- ✓ Logistic Regression.
- ✓ KNN (K-Nearest neighbours).
- ✓ Decision Tree.
- ✓ Random Forest.
- ✓ SVM (Support Vector Machine).
- ✓ Naïve Bayes

LOGISTIC REGRESSION:

- Logistic regression is used to obtain odds ratio in the presence of more than one explanatory variable.
- The procedure is quite similar to multiple linear regression, with the exception that the response variable is binomial.
- The result is the impact of each variable on the odds ratio of the observed event of interest.
- Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set.
- A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.
- There are three main types of logistic regression: binary, multinomial and ordinal.
- Its features are sepal length, sepal width, petal length, petal width.
- Besides, its target classes are setosa, versicolor and virginica.
- However, it has 3 classes in the target and this causes to build 3 different binary classification models with logistic regression.

KNN (K-Nearest neighbours):

- K-Nearest Neighbor (KNN) Algorithm for Machine Learning
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- **The K-NN working can be explained on the basis of the below algorithm:**

  Step-1: Select the number K of the neighbors

  Step-2: Calculate the Euclidean distance of K number of neighbors

  Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

  Step-4: Among these k neighbors, count the number of the data points in each category.

  Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

  Step-6: Our model is ready

DECISION TREE:

- A decision tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered.
- The model is a form of supervised learning, meaning that the model is trained and tested on a set of data that contains the desired categorization.
- A decision tree typically starts with a single node, which branches into possible outcomes.
- Each of those outcomes leads to additional nodes, which branch off into other possibilities.
- This gives it a treelike shape.
- There are three different types of nodes: chance nodes, decision nodes, and end nodes.
- The main components of a decision tree include a root node, decision nodes, chance nodes, alternative branches, and an endpoint node.
- Optional features include rejected alternatives

RANDOM FOREST:

- Random forest is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result.
- Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.
- The random forest is a classification algorithm consisting of many decisions trees.
- It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.
- It is a set of Decision Trees.
- Each Decision Tree is a set of internal nodes and leaves.
- In the internal node, the selected feature is used to make decision how to divide the data set into two separate sets with similars responses within.

SVM (Support Vector Machine):

- Support Vector Machine Algorithm
- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.
- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

NAVIE BAYES:

Naïve Bayes is a simple probabilistic machine learning algorithm that is commonly used for classification tasks. It is based on Bayes' theorem, which describes the probability of an event given prior knowledge of related events. In Naïve Bayes, the input variables are assumed to be independent of each other, which simplifies the computation of the probability of the output variable given the input variables.

The algorithm works by first training a model on a labeled dataset, which involves calculating the probabilities of each input variable given each class label. These probabilities are then used to calculate the probability of each class label given a set of input variables using Bayes' theorem. The class label with the highest probability is then assigned to the input variables.

Naïve Bayes is particularly useful for text classification tasks, such as spam filtering, sentiment analysis, and topic classification. It can also be used for other classification tasks, such as image classification, as long as the assumption of input variable independence holds.

One of the advantages of Naïve Bayes is that it is relatively fast and requires less training data compared to other more complex machine learning algorithms. However, its performance may be limited by the assumption of input variable independence, which may not hold in some real-world applications.

4.2 MODEL ARCHITECTURE:

LOADING DATA SET → IDENTIFYING THE ATTRIBUTES FINDING THE RISK OF FIRE → COLLECTION OF DATA AND PRE -PROCESSING → K-NEAREST NEIGHBOURS OR SVM → ARTIFICIAL NEURAL NETWORKS → OBTAIN RESULTS → CONCLUSION

4.3 SOFTWARE DESCRIPTION

Software requirements:

Operating system: Windows 11

Platform: google colab.

Programing language: python

RESULT

Output:

```
X=[[array([[[254, 220, 165],
        [254, 221, 165],
        [254, 220, 165],
        ...,
        [255, 204, 148],
        [254, 203, 147],
        [253, 204, 148]],
       [[254, 221, 165],
        [253, 220, 164],
        [255, 222, 166],
        ...,
        [255, 204, 148],
        [255, 204, 148],
        [253, 204, 148]],
       ...,
       [[ 63,  47,  35],
        [ 63,  47,  34],
        [ 63,  49,  36],
        ...,
       [[ 62,  46,  33],
        [ 63,  47,  34],
        [ 64,  48,  35],
        ...,
        [ 56,  43,  27],
        [ 52,  41,  27],
        [ 55,  44,  29]]]], dtype=uint8), 0]
……..
……...
...,
        [ 92,  77,  51],
        [ 91,  76,  50],
        [ 92,  75,  48]],

       [[ 86, 154, 187],
```

[87, 152, 191],

[102, 152, 178],

...,

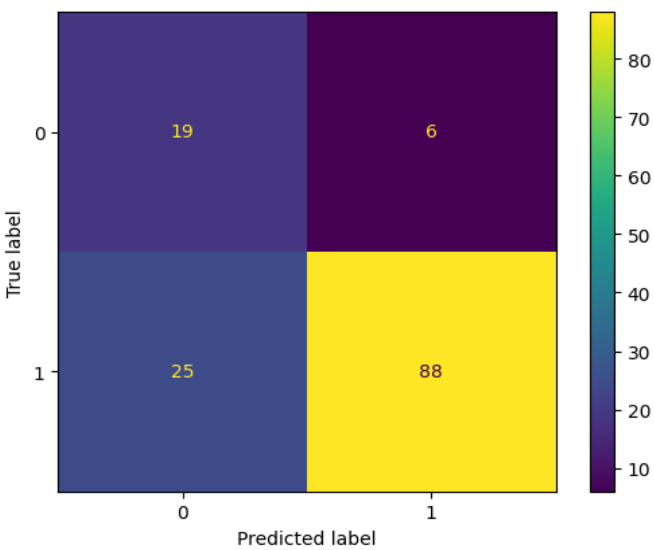
[91, 76, 50],

[88, 73, 47],

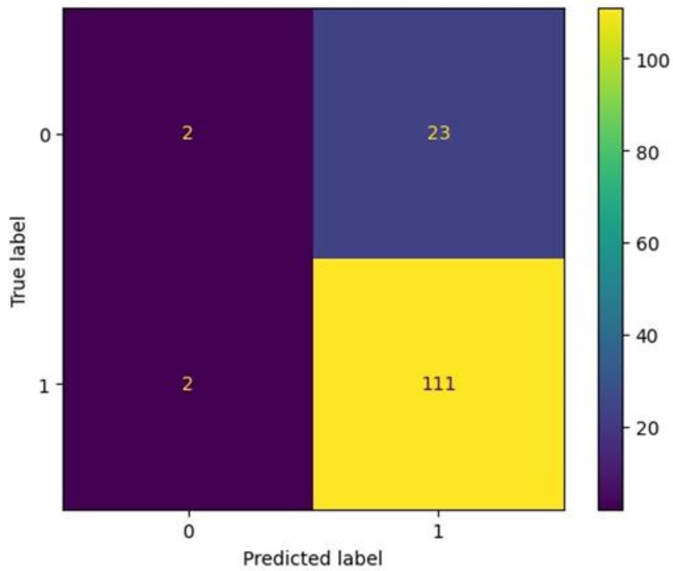
[87, 74, 46]]], dtype=uint8), 1]]

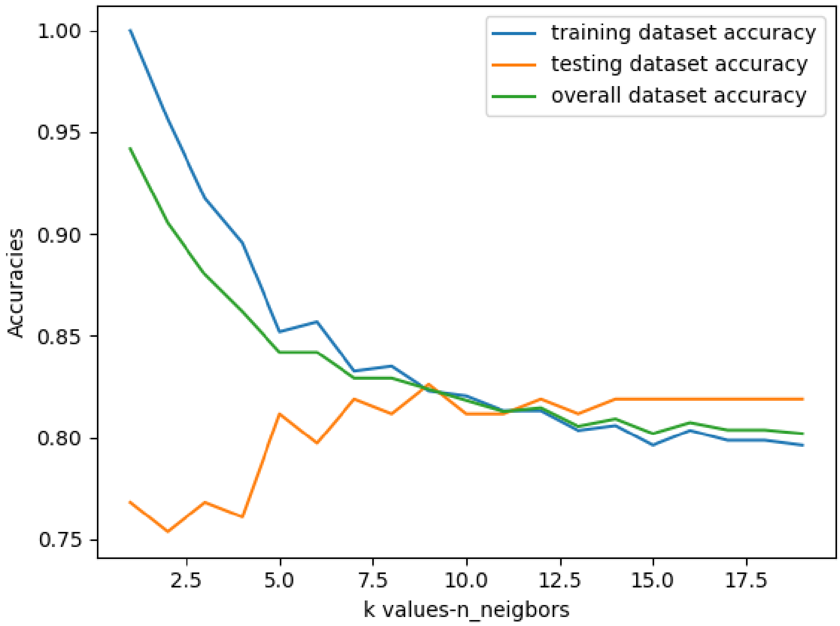
Y =[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…………0,1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,1, ……………1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

#LOGISTIC REGRESSION
Confusion matrix:

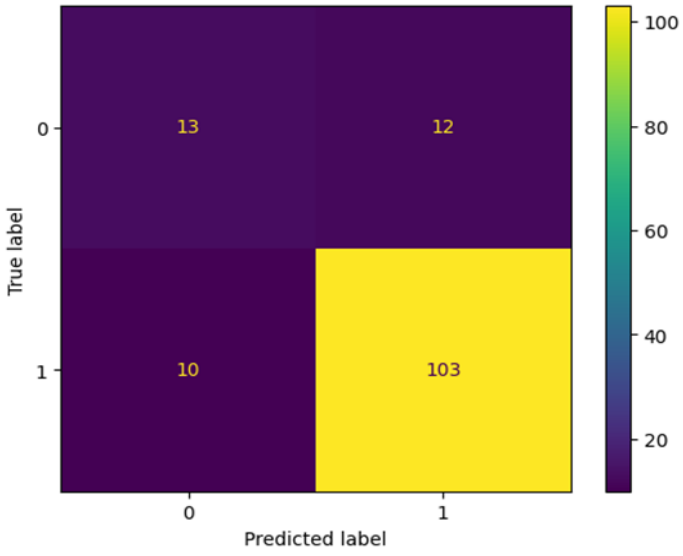


#KNN
Confusion matrix:

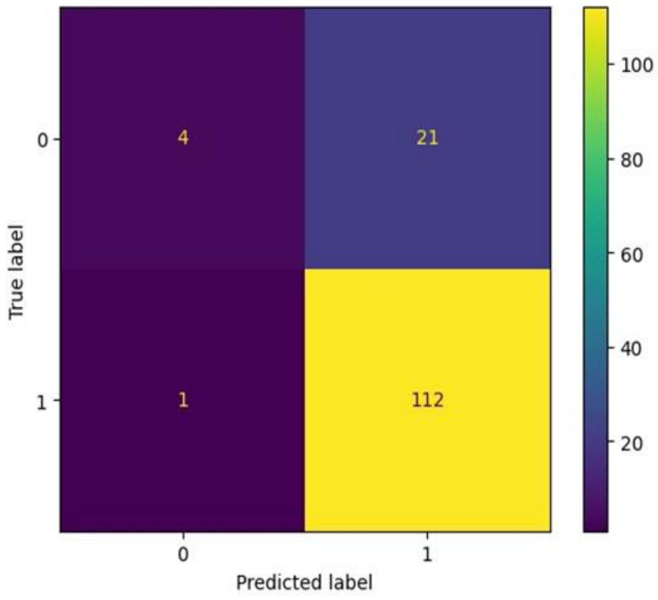




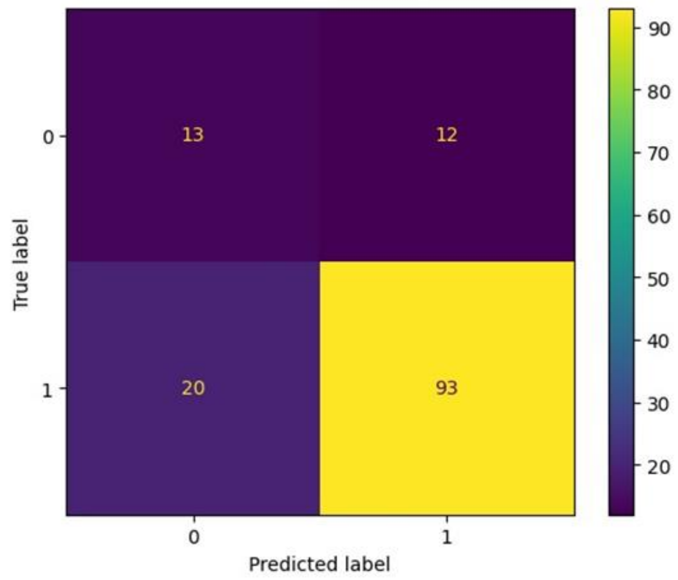
#DECISION TREE
Confusion matrix:



#RANDOM FOREST
Confusion matrix:



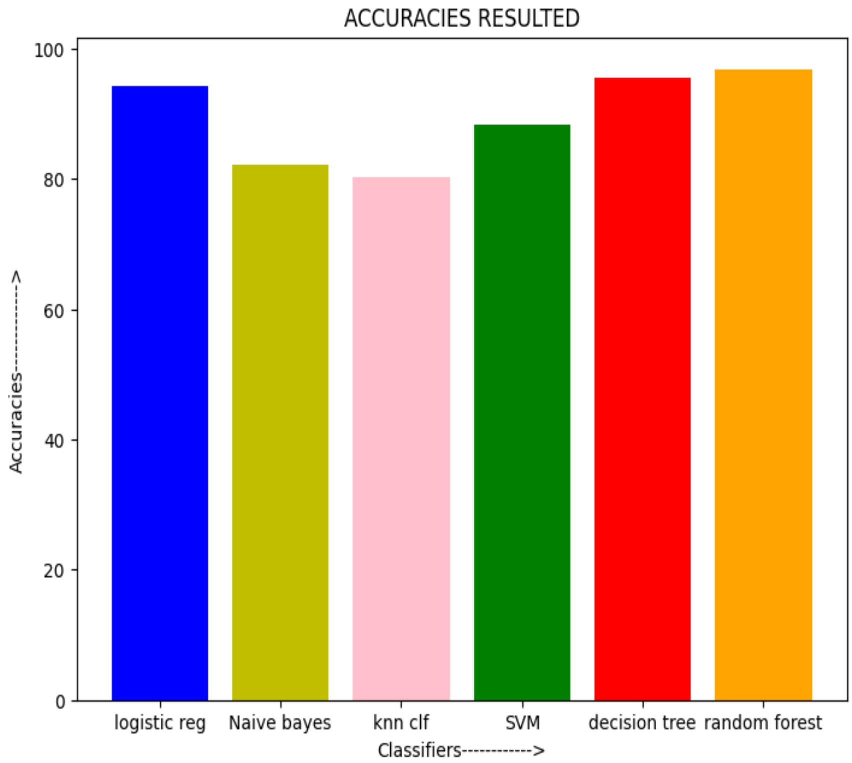
#NAIVE BAYES
Confusion matrix:



Accuracy:

Logistic Regression: 0.9436363636363636
Navie Bayes: 0.8218181818181818
KNN: 0.8018181818181818
Support Vector Machine: 0.8836363636363637
Descision Tree: 0.9545454545454546
Random Forest: 0.9690909090909091

Here the random forest has the high accuracy. It is able to correctly classify a high percentage of the instances in the dataset. In other words, the classifier is able to predict the correct class label for a large proportion of the instances in the dataset compared to other classifiers.


ACCURACIES RESULTED
Accuracies------------->
0
20
40
60
80
100
logistic reg
Naive bayes
knn clf
SVM
decision tree
random forest
Classifiers----------->

CONCLUSION AND FUTURE SCOPE

we are able to detect cars more precisely and identify the objects individually with exact location of an cars in the picture Car object detection is an important task in the field of computer vision, with applications in autonomous driving, traffic monitoring, and surveillance systems. Recent advancements in deep learning and machine learning algorithms, such as KNN,SVM,Naïve Bayes, have led to significant improvements in the accuracy and efficiency of car object detection models. However, challenges such as variation in lighting and weather conditions, occlusion, scale variation, and real-time performance requirements continue to pose significant challenges in the field.

The objectives of car object detection are to improve safety, optimize traffic flow, enhance surveillance systems, provide valuable insights, and develop advanced computer vision techniques. These objectives are critical to the development of more efficient and safe transportation systems. In recent years, significant progress has been made in the field, and car object detection continues to be a rapidly advancing area of research with promising future applications.

Overall, car object detection is a challenging and important problem with significant potential to revolutionize transportation systems and improve the safety and efficiency of our roads. Ongoing research and development in this field will be critical to realizing this potential and addressing the challenges that still need to be overcome.

.

REFERENCES

1. A. Bochkovskiy, C. Wang, and H. M. Liao, Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

2. K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

3. M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

4. W. Yang, Z. Huang, L. Wang, and X. Gao, "Deep Learning-Based Vehicle Detection in Aerial Images: A Review," Remote Sensing, vol. 12, no. 11, 2020.

5. Y. Liu, J. Yan, and Y. Ouyang, "Object Detection in 20 Years: A Survey," arXiv preprint arXiv:1905.05055, 2019.

6. S. Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Advances in Neural Information Processing Systems (NIPS), 2015.

7. A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI Vision Benchmark Suite," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

8. L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 40, no. 4, 2018.