International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 3, May-June 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

VEHICLE PATTERN RECOGNITION USING MACHINE LEARNING & DEEP LEARNING TO PREDICT CAR MODEL

[1]Gogum Sravani, [2]Mrs B.Bhagya Lakshmi
[1](Student, Dept. of Master of Computer Applications, Amrita Sai Institute of Science and Technology, Paritala, Andhra Pradesh, 521180, India.
Email: sravanigogam2@gmail.com)
[2](Asst.Prof, Dept. of Computer Science & Engineering, Amrita Sai Institute of Science and Technology, Paritala, Andhra Pradesh, 521180, India.
Email: bhagya.bolla@gmail.com)

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

Abstract:

The study examines recognising vehicle attributes by studying their appearance in images, rather than focusing on facial recognition. A range of algorithms is reviewed to find out vehicle properties, whether at a broad level (vehicle types) or narrow level (model and type of vehicle). The paper explains two different methods: easy classification and adaptable metric learning. Simulating a situation involving vehicle attributes, we look into these methods and compare them in experiments. Using SVM, KNN, CNN and Linear Regression, our project predicts car models by combining recognised features of inference and patterns. This work uses the Stanford Cars dataset for its experiments and analysis.

Keywords *— Vehicle attribute recognition, image-based analysis, machine learning, vehicle type classification, vehicle make and model recognition, SVM, KNN, CNN, Linear Regression.*

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

1. Introduction

Statistics used in designing and planning transport infrastructure are largely gathered through traffic monitoring. Even though tracking how many vehicles pass by reveals some facts, it fails to point out the finer details of how the traffic is flowing. One result of this could be learning about popular routes for drivers, as seen through comparisons between heavy vehicles versus light ones, as well as tracking individual vehicles. By having accurate and precise details, analysts can evaluate transportation users properly which is important for judging the effect of upcoming changes in the transport industry.

Usually, people collect traffic data by going out to count cars or by performing roadside interviews in person. Even so, using these techniques may take away from enjoying your drive and sometimes they are not very efficient. Several technologies have been created to improve the process of collecting data.

Inductive ground loops track the magnetism caused by passing vehicles and can roughly guess their types. Secondly, laser scanners are capable of collecting the same type of data. Additionally, experts have studied using audio to identify vehicles by studying key characteristics present in short audio frames.

Because they are cheap and easy to find in the market nowadays, camera methods for traffic monitoring have grown popular recently. Thanks to cameras, we have a lot of information and can use different recognition methods as well. Unlike inductive loops and laser scanners, camera systems can make use of the latest trends in computer vision to classify different objects.

Improving computer vision methods has often been a challenge for cameras used in monitoring traffic. Still, with deep learning, this area has been transformed in important ways. In the last decade, image classification technologies have come very close to matching the accuracy people can achieve. Thanks to huge data collections, training and fine-tuning these modern models has become possible.

2. Literature Survey

2.1 Using Hierarchies with Many Details for Correct Object Detection and Segmentation

So far, performance in object detection has not improved much, so we propose R-CNN that pairs high-capacity CNNs with region proposals and as a result increases mAP by 30%. Cuts and segments various objects better when the data to label is not plentiful by first pre-training and then fine-tuning specifically for the subject matter.

2.2 Vehicle Detection and Tracking is based on analysing the motion in car video records.

The system uses real-time video analysis to maintain safety and assist in autonomous driving. We rely on geometry features and a hidden Markov model (HMM) to make vehicle detection strong, able to work in both day and night video conditions and allow studios to track vehicles almost instantly from car-mounted cameras.

2.3 HybridNet: A Fast System for Detection of Vehicles in Autonomous Driving

The HybridNet system runs in two stages and is designed to detect faces quickly with a high rate of accuracy. Compared to others, it improves on ideas created in the first step, doing so much faster and more accurately, as found on the KITTI and PASCAL VOC2007 datasets.

2.4 Applying Convolutional Neural Network with YOLO for Finding People and Cars

The paper introduces a modified version of YOLO to detect people in real time for ADAS. When Many CNN layers are applied, the system achieves both high reliability and good performance in ADAS systems.

2.5 Exchange of Information Between Vehicles

With plans made, cars will be able to warn drivers of hazards by 2020. First, systems will send alerts, while future ones will be able to control cars during emergencies and enable self-driving technology. The paper looks into V2V technologies and the effects they might have on traffic safety.

3. System Analysis

3.1 Existing System

Unlike traditional approaches in classification, DNNs have proved to be highly effective. DNNs can understand more complicated patterns, so their training algorithms for identifying objects do not rely on manual design. Still, it is complicated by things such as different objects, their positions, overlapping objects and varying lighting.

3.2 Proposed System

Our aim is to use machine learning such as SVM, KNN, CNN and Linear Regression, along with feature inference and pattern recognition, to predict car models. The car dataset from Stanford is what we use during the implementation.

3.3 Feasibility Study

The feasibility study cheques if the proposed system can work, function well and be financially feasible.

- Minimum hardware or software is required for this system to run. Since the project makes use of existing resources and

technology available at NIC, it is cost-effective.

- The system is appropriate for the company's operations. This has been created according to how users need it to work, so we do not anticipate any issues from them. Everything is planned in the project to help improve how the system utilises its resources.

- Required technology is available and the system being suggested is able to carry out necessary processes and respond appropriately to queries. The system can handle more data and is always accurate, reliable and secure. You can develop AI using what your company already possesses and free software to fit it in with your current technology.

4. System Requirements Specification

4.1 The process of software development is called the Software Development Life Cycle (SDLC).

What is meant by the term SDLC?

The SDLC involves a series of steps that are organised to help create and maintain software applications.

SDLC Phases:

1. Gather Information: See what the client wants from the finished product.
2. Prepare a document covering the whole design, including its functional and non-functional aspects.
3. Use UML to develop the main sections of Low-Level Design such as use case and sequence diagrams.
4. Coding: Bring together and develop different software modules.
5. Testing: Make sure that the system satisfies the client's needs; make changes if needed.
6. Push the application to the production environment when it has passed testing.
7. Support: Supply help and updates to the system once it has been put into use.

4.2 Checking the Software

What does the term Software Validation mean?

Software validation is used to ensure the software is how the business needs it and addresses what customers expect. It cheques whether the software operates properly in actual conditions, using the software requirements specifications (SRS).

Once verification is complete, validation happens next and is generally the end stage of development.

4.3 What the software needs to do

Functional requirements specify what a system should do such as calculate information or handle data. They are taken from use cases and show the behaviour the system should display. All functional requirements are described by saying, "The system must do <requirement>."

4.4 Requirements Other Than Functional

Performance, security, usability and scalability are among the quality attributes set by non-functional requirements. Such requirements specify the limits the system must follow such as being able to handle 10,000 users using the system at the same time.

Advantages:

- ⇨ Helps maintain conformity with the standards set by law and work performance.
- ⇨ Boosts the user's security and ease of usage.

Disadvantages:

- ⇨ Might raise the costs and complexity involved in development.
- ⇨ After architecture is finalised, it can only be adapted with difficulty.

Key Learning:

- ⇨ The focus of non-functional requirements is on the system's speed, dependability and security level.

- ⇨ When setting out functional requirements, you focus on actions and non-functional requirements dictate how these actions are carried out.

5. System Design

5.1 System Specifications

Hardware Requirements:

Category	Specification
System	Pentium i3
Storage	500GB
Display	14" Colour Monitor
Input	Optical Mouse
Memory	4GB

TABLE1 : HARDWARE-REQ

5.2 System Architecture

Python programming is used in the system to organise the architecture of a product so that hardware and software operate together and users can interact smoothly.

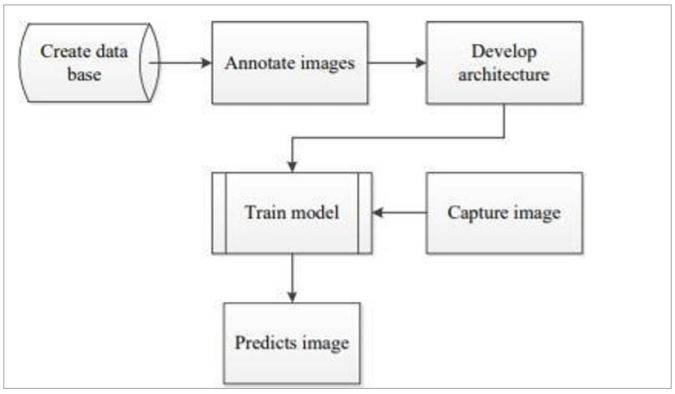

Fig1 : Sys Architecture

5.3 Overview of UML Diagrams

UML is a common tool used to model object-oriented programmes. This way of modelling can be applied everywhere to declare, describe, illustrate and document information about a system. UML drawings make it simpler to represent and model software designs and engineering methods. What UML tries to do:

- ➢ Allow the user to easily design and describe systems using a drawing-based language.
- ➢ Allow for the development of specific features.
- ➢ Make it independent of both programming languages and the process of development.
- ➢ Include collaborations, frameworks and components as part of the platform.
- ➢ Support the use of OO (Object-Oriented) tools.

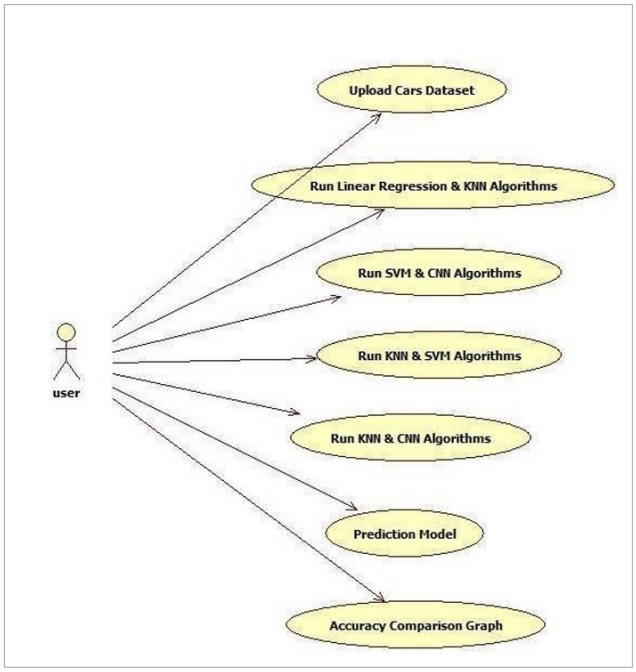

Fig2: Uml Diagram

6. Testing

6.1 System Testing

When the quality of the software is tested, it is to confirm that it performs according to the given requirements. Once each module is tested and combined, the entire system should be tested to match the original design. Our goal is to identify any issues that could affect how the system works and resolve them. At this stage, the software and hardware are tested to ensure they can communicate properly.

Testing has two main methods, known as black box testing and white box testing. In black box testing, the main objective is to cheque the external operations of the software, without looking at the internal workings. It allows you to notice if a function is absent, the interface isn't working or the performance is off. White box testing, however, looks at the internal code to test the validity of every possible route, choice option and data structure.

Tests are carried out on several different levels.

- Unit testing makes sure that every module performs as expected, based on what it is designed to do.
- In integration testing, various modules are checked to confirm they communicate with each other smoothly.
- By conducting functional testing, you are checking that the system operates as intended for input, output and work requirements.

6.2 Sample Test Case

Test Case ID	Test Case Name	Test Case Description	Test Steps	Expected Result	Actual Result	Priority
01	Start the Application	Host the application and test if it starts, making sure the required software is available	If it doesn't start, we cannot run the application	The application hosts successfully	High	High
02	Home Page	Check the deployment environment for properly loading the application	If it doesn't load, we cannot access the application	The application is running successfully	High	High
03	User Mode	Verify the working of the application in freestyle mode	If it doesn't respond, we cannot use the freestyle mode	The application displays the freestyle page	High	High
04	Data Input	Verify if the application takes input and updates	If it fails to take the input or store in the database, we cannot proceed further	The application updates the input to the database	High	High

TABLE2 : SAMPLE TEST CASE

7. Conclusion

All things considered, our new convolutional neural network is both simple and extremely efficient. When it comes to object detection, the convolutional features from our system are more effective than the current image classification networks. Using our approach allows for greater accuracy, as it trades off some flexibility for more efficient and quicker processing in both parts of the process. Detecting noise from images is something out model does not handle at the moment and could be improved in the coming days.

Acknowledgement

I am thankful to the Management of Amrita Sai Institute of Science and Technology for giving me an opportunity to work with his project.

I would like to the thank **Dr. M. Sasidhar**, Principal, Amrita Sai institute of science and technology, for his constant encouragement and support during the progress of this work.
I am deeply grateful to **Dr. P. Chiranjeevi**, Professor and Head of the Department, for his valuable guidance and consistent support during the course of the project.
A special note of thanks to my internal guide, **Mrs.B.Bhagya Lakshmi (M.Tech)**, for her exceptional guidance, constant motivation, and continuous encouragement, which played a crucial role in the successful completion of this project.

GOGUM SRAVANI

References

[1] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627-1645, Sep. 2010.
[2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 675-678.
[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2012, vol. 1, pp. 1097-1105.
[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Real-time multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7291-7299.
[5] Z. Yang and R. Nevatia, "A multi-scale cascade fully convolutional network face detector," in *Proc. 23rd Int. Conf. Pattern Recognit.*, Cancun, Mexico, Dec. 2016, pp. 3943-3948.
[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504-507, Jul. 2006.
[7] Y. Bengio, "Learning deep architectures for AI," *Foundations Trends Mach. Learn.*, vol. 2, no. 1, pp. 1-127, Jan. 2009.
[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 779-788.
[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137-1149, Jun. 2017.
[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580-587.
[11] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154-171, 2013.
[12] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391-405.
[13] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Washington, DC, USA, Dec. 2015, pp. 1440-1448.
[14] K. Lenc and A. Vedaldi, "R-CNN minus R," *Comput. Vis. Pattern Recognit.*, pp. 1-12, 2015.
[15] H. Jiang and E. L. Miller, "Face detection with the faster R-CNN," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Washington, DC, USA, Jun. 2017, pp. 650-657.
[16] Y. H. Byeon and K. C. Kwak, "A performance comparison of pedestrian detection using faster R-CNN and ACF," in *Proc. 2017 6th IIAI Int. Congr. Adv. Appl. Informatics*, Hamamatsu, Japan, Jul. 2017, pp. 858-863.
[17] X. Zhao, W. Li, Y. Zhang, T. A. Gulliver, S. Chang, and Z. Feng, "A faster R-CNN-based pedestrian detection system," in *Proc. IEEE Vehicular Technol. Conf.*, Montreal, QC, Canada, Sep. 2016, pp. 1-5.
[18] M. C. Roh and J. Y. Lee, "Refining faster-RCNN for accurate object detection," in *Proc. 15th*

IAPR Int. Conf. Mach. Vis. Appl., Nagoya, Japan, May 2017, pp. 514-517.
[19] M. Laavanya and V. Vijayaraghavan, "A sub-band adaptive visushrink in wavelet domain for image denoising," *Int. J. Recent Technol. Eng.*, vol. 7, no. 5S4, pp. 289-291, 2019.

