Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Multi-Model Evaluation of Computer Vision Techniques for Fine-Grained Vehicle Recognition

Prithvi R[1], Mythily M[2*], Kavitha M N[3], Iwin Thanakumar Joseph S[4], Kethsy Prabavathy A[5], Dr. G. Naveen Sundar[6]
[1]*Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu 641114, India*
[2*]*Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu 641114, India*
mythily.m@gmail.com
[3]*Department of Computer Science and Engineering, PSG Institute of Technology and Applied Research, Coimbatore, TamilNadu 641062, India*
[4]*Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhrapradesh 522302, India*
[5]*Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu 641114, India*
[6]*Division of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu 641114, India*
naveensundar@karunya.edu

ARTICLE INFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Vehicle make and model recognition is a crucial component in applications such as intelligent transportation systems, law enforcement, and autonomous vehicles.This paper presents a comparative analysis of Machine Learning approaches — KNN, SVM and Decision Tree Classifier which are the first appraoch and three deep learning models—Convolutional Neural Network (CNN), YOLOv8, and Faster R-CNN— are the second approach for vehicle recognition tasks. The models were evaluated on a two datasets comprising 197 vehicle classes (dataset1) under various conditions and the other with was scrapped manually with 17 classes (dataset2), focusing on metrics such as validation accuracy, inference speed, robustness to occlusion, and computational efficiency. YOLOv8 emerged as the best-performing model in both the datasets, achieving a mean Average Precision (mAP) of 95% with dataset1 and 30% with dataset2, with an inference speed of just 10 ms per image, making it highly suitable for real-time applications. Faster R-CNN demonstrated exceptional precision and robustness in handling complex scenarios but was constrained by slower inference speeds with an accuracy of 74% with dataset1 and 65% with dataset2. In CNN, which is computationally efficient, suffered from significant overfitting, with a testing accuracy of only 10% with dataset2. Whereas with dataset1 they gave an 78% of testing accuracy and 90% of training accuracy. The findings of this study emphasize the strengths and limitations of each model, providing insights into their applicability across various real-world scenarios.

Keywords: Object detection with Convolutional Neural Networks (CNN), Faster R-CNN, YOLO, KNN, SVM, Decision Tree Classifier, deep learning, real time processing, automotive applications, machine learning, vehicle make and model classification, vehicle recognition

I. INTRODUCTION

With the advances in real world applications that require vehicle make and model recognition ranging from intelligent transportation systems to law enforcement and urban traffic management, vehicle make and model recognition has become an important component. Due to the importance of vehicles as a key component of today's modern infrastructure, vehicles are accurately identified and classified to enable improving safety, efficiency and automation in such fields. Boston Dynamics controls the motion of the vehicles and its robots through the use of localization algorithms in continual simulation, operating the drones to follow scripted instructions or to respond to human commands to perform tasks such as delivering food or supplies to remote locations. In this study, we begin to understand and compare these three prominent models, emphasizing their suitability for varied real world scenarios and the possible influence that they will exert on technological development.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

As the intelligent systems adoption in the automotive and public sector grows, there is a growing need for robust and scalable solutions for vehicle recognition. Although once dominant, traditional image processing techniques have been found to be inadequate in handling the complexities of modern datasets such as varying lighting conditions, occlusion, overlapping objects, and great variety in vehicle design. Deep learning has opened this field to revolution by making models learn hierarchical features directly from data, without relying on hand crafted features. At the same time, these systems are rendered more fit to the requirement for adaptability to the dynamic character of actual world circumstances.

The first approach is with Machine Learning Techniques such as KNN, SVM and Decision Tree classifer are implemented in order to classify the cars. They are implemented on both dataset1 and dataset2. K-Nearest Neighbors are well known techniques for classification and regression. They are easy to understand and implement. But they do not perform well for large datasets. SVM (Support Vector Machine) which are known for classification and regression. They are used for Image Classification, Text Categorization. They are computationally expensive for larger datasets. Decision Tree Classifier are known for classification. They are prone to overfitting in the case of larger datasets. In this traditional approach, we train them by extracting the feature manually, whereas in the deep learning models, they consists of multiple hidden layers which extract the features by themselves.

Convolutional Neural Networks (CNNs) are recognized as one of the most popular vehicle recognition models. CNNs are particularly tailored for image related task and perform robustly in multi-class classification tasks. CNNs have the benefit of extracting spatial features from the images through convolutional layers helping them recognize patterns such as logos, shapes and textures, critically important for making and model identification. This is due to the simpler nature of architecture when compared to the object detection models which in turn allows computational efficient performance suitable for scenarios in which classification alone meets demand without the requirement of localization. But, CNNs are not suited to applications that need object detection and location and their applications are limited to situations where we have clear and uncluttered data.

CNNs have limitations, however, and object detection models such as YOLOv8 fixes this by combining classification and localization together into one unified framework. Probably the most impressive of the lot is YOLOv8, the latest YOLO (You Only Look Once) family that is also impressively fast and accurate, making it ideal for real time applications. YOLOv8 is different from other traditional two stage object detection models in that they follow a one stage approach, considering bounding boxes regression and class prediction at the same time. Rapid inference is ensured by this architecture, an essential embodiment in dynamic environments such as traffic monitoring and autonomous vehicles. With occluded and overlapping objects, YOLOv8 can better be applied to congested urban settings, making it fully reliable!

The alternative to that is Faster R-CNN whose formulation is stronger for high precision scenarios. Compared to YOLOv8, Faster R-CNN follows a two stage architecture, with first stage generating region proposals and second stage refining them. Faster R-CNN uses a Region Proposal Network (RPN) for detecting regions of interest that will help it detect small or partially visible objects. Faster R-CNN is extremely efficient coupled with a backbone network such as ResNet50 to extract features for complex datasets with high variability. At the expense of less speed, it has better robustness and precision, making it the preferred architecture for forensic analysis or offline traffic studies. Nevertheless, the computational expense of Faster R-CNN makes it impractical in real time systems, especially under constrained resource conditions.

The vehicle recognition process depends critically on the dataset used in this study. There are two datasets, one has 197 different classes and the other with 17 classes, ranging from different car makes and models to modern automobile designs. Where high resolution images and detailed annotations help the models to learn a diverse set of features and thereby generalize better. Preparation of this dataset means preprocessing steps like normalization, resizing to a certain size (uniform) and augmentation (blanking the input and making it come to life again). Real world variations are simulated by techniques like random flips, random rotations and even random brightness adjustments, making the trained models robust enough. Finally, the models are tested further, with challenging

scenarios included such as occlusion and overlapping objects, yielding important practical insights into their true performance.

Environmental variability has long been a major problem with vehicle recognition. However, lighting, weather, and camera angle all play major roles in affecting how good the input data is, which severely affects the performance of the model. CNNs are very good at controlled scenarios but less so when there's a high level of variability. Since speed and robustness are the focus of YOLOv8, YOLOv8 is better at adapting to these challenges, as the real time application is the kind that it can address. However, slower because of the region-based interpretation, Faster R-CNN triumphs over these challenges since it is able to concentrate on selected regions of the image.

Vehicle make and model recognition has many applications across multiple domains. These models are used in intelligent traffic systems in order to monitor real time traffic and thus assist the authorities in managing congestion and ensuring compliance with traffic regulations. Such systems allow data from vehicles to help in personalised customer experiences in the corner of the automotive industry. Vehicle recognition is used by law enforcement to help track stolen cars, monitor high speed pursuits and enforce parking regulations. Furthermore, these models are also important components in autonomous vehicles to detect and respond to other road objects to improve safety and efficiency.

However, many problems remain in deploying vehicle recognition systems. The main limitation is that deep learning models, especially Faster R-CNN, have a high computational demand that puts demand on high performance hardware. In edge computing scenarios however, resources are constrained and this challenge is further aggravated. Therefore, YOLOv8 partially mitigates this issue with a tradeoff between the accuracy and computational efficiency of these systems but we still need further optimizations to make these systems scalable to larger audiences. More importantly, vehicle recognition systems must be carefully addressed from an ethical standpoint, including privacy issues and potential misuse, to prevent responsible practice.

The choice of model is delimited by what the application requires. CNN are efficient and straightforward methods for the tasks that need simple classification. With speed, accuracy and robustness being thirded YOLOv8 is the most versatile option, serving well in real time applications. Although less well suited to time critical scenarios, faster R-CNN is still the preferred choice when you need high precision, like forensic analysis or very detailed traffic studies. Analysis between these models is useful for identifying the strengths and weaknesses of each model, allowing us to choose the model that best fits the use case we are dealing with.

Future developments possible for vehicle recognition systems will try to reduce the shortcomings in the current models. These results may open up the applicability of Faster R-CNN in real time applications without sacrificing its precision. Another possible result for the integration of ensemble methods, which combine the strengths of multiple models, may be developed: YOLOv8 is fast like YOLOv8 and Faster R-CNN is precise like Faster R-CNN. Also, hardware acceleration, including AI specific chips and GPU optimization, is expected to drive the efficiencies of these systems.

Training deep learning models without a focus on quality of dataset is pointless. Modeling performance, however, can still be further improved by expanding the dataset to contain more vehicle classes, more difficult scenarios, and a wider range of environmental conditions. Synthetic data, however, can be included as well — created using advanced techniques such as Generative Adversarial Networks (GANs) in order to simulate conditions that are difficult to record in 'real world' scenarios. These advances will continue to make vehicle recognition systems successful in more complex and dynamic operating environments.

With demand for intelligent and automated systems across industries, it is likely that such systems to adopt vehicle make and model recognition systems will grow strongly in the years to come. These systems will become more integral to our everyday life as they grow increasingly integrated, their influence will grow on safety, efficiency and

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

convenience. With approaches to confront the current limitations and find the new possibilities, vehicle recognition systems can offer even higher accuracy and benefit to intelligent transportation systems and related areas.

II. LITRETURE SURVEY

In [1], it focuses on deep transfer learning and Support Vector Machines (SVM) integration for vehicle make and model recognition to overcome the variation problems in achieving high accuracy for a variety of environments. This research used a pretrained CNN as a feature extractor for SVM based classification. Thanks to transfer learning, the authors could overcome a common obstacle to deep learning: it didn't need extensive training sets to run. The study was held as a success-filled measure as with the proposed system classifying vehicle makes and models under varying conditions including that of changes in lighting and angles. Compared to traditional methods, which are usually based on handcrafted features and inflexible to various datasets, the system achieved a classification accuracy above that. This approach had the advantage of having reduced training time as the pretrained CNN models came with a strong feature representation that could be easily fine tuned. The comparative analysis was also a key highlight of the research, in which it showed that transfer learning based approach performed better in accuracy and could also benefit the computational efficiency compared to the conventional CNN architectures. Unfortunately, such dependence on SVM is not inherently scalable to large datasets, limiting real time applications. The importance of dataset diversity was also called out in the study, with the authors calling for more work on obtaining larger, more diverse datasets to improve generalization. The research offers useful insights into the ability to fuse deep learning with traditional machine learning techniques resulting in a hybrid approach. This work has implications beyond vehicle recognition, and it shows promise for other resource constrained applications via the use of transfer learning.

In [2], the author has proposed a novel make and model recognition approach for vehicles using the Residual SqueezeNet architecture. The challenge was to optimize real time performance without sacrificing accuracy as the current deep learning models are computationally constrained. The author integrated residual connections into the lightweight SqueezeNet architecture, achieving a system that provides high speed of inference while keeping the classification precision. Specifically, the Residual SqueezeNet architecture was designed to reduce the model size and computation overhead so that the architecture can be successfully deployed on edge devices with very limited resources. This innovation was particularly useful for applications in developing regions where such high performance hardware is not well suited to. The model achieved an accuracy of 92% for a benchmark dataset at a 10 millisecond per image inference speed. It’s this balance of speed versus accuracy that really made this highly suitable for real world applications. The authors also underscored the significance of dataset augmentation to help the model be more robust to challenging conditions — e.g., under different illumination and during occlusion. A strength for the model is in distinguishing visually similar vehicle models, but it still faced problems with distinct vehicle models. The authors suggested that such system performance could be further enhanced by incorporating multimodal data amounting to license plate recognition and color detection. Additionally, the study recommended standardized datasets for vehicle recognition so as to enable the benchmarking and model comparison. In particular, this work significantly advances the state of the art toward practical implementations of vehicle recognition systems in resource constrained environments. This work demonstrates the feasibility of using lightweight architectures to provide real time performance in deep learning applications.

In [3], the use of Convolutional Neural Networks (CNNs) to recognize vehicle make and model, focusing on the capability of CNNs to automatically learn the hierarchical features in raw image data. The problem of handcrafted features is addressed, as most current methods are restricted by the requirements of a large number of handcrafted features, and they do not perform well in generalizing to real world scenarios. The proposed CNN model was trained using a dataset consisting of images taken under different angles, lighting scenarios and environmental factors and was benchmarked against a dataset which was created by randomly resizing each test image by 6 different factors using the ground truth values as the training data. In controlled as well as dynamic environments, the study was able to beat traditional approaches by 93 percent accuracy. Key contribution was demonstration of the CNN's handling of complex visual features, e.g. differentiate between similar vehicle models. Another strength of the study was in analysing the effectiveness of different hyperparameters to the model, in the form of learning rates and batch sizes.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

This analysis helped to better optimize CNNs for given applications. The authors stressed that data augmenting is essential for model robustness because there is less training data available in many cases. Nevertheless, the research identified issues with the computational burden of CNNs for training. The authors found that these challenges could be mitigated through using transfer learning or pruning techniques. The study also encouraged the development of standardized metrics for assessing vehicle recognition systems for the purposes of comparison across studies. This work emphasizes the tremendous benefit of CNNs for vehicle recognition, as they are capable of generalizing to diverse datasets and represent complex visual features. The wide range of applications in which its findings have implications include traffic monitoring to autonomous vehicle systems.

In the study [4], they have built a real time vehicle make and model recognition system tradeoff between accuracy and computational efficiency. In dynamic environments, the approach integrated deep learning with conventional image processing methods to improve performance. The proposed system was evaluated in real time scenarios, including traffic monitoring and parking management, where inference must be fast. With an average inference time of 0.5 seconds per image, an accuracy of 92.8% was achieved. For this performance, we were able to optimize the architecture of the deep learning model, and use efficient image processing techniques for preprocessing. The key innovation of the research was that domain specific knowledge was integrated into the system design. For example, the authors increased classification accuracy by using vehicle specific features, such as logos and shapes. For real time applications, they also implemented a multi-threaded processing pipeline to deal with these high volumes of data. The system worked well, but failed to deal with occlusions and thus highly overlapping vehicles, which are a common density in reality based scenarios. However, the authors suggested that these limitations could be overcome by applying the advanced object detection methods, such as Faster R-CNN or YOLO. They also indicated that more robust systems require larger and more diverse datasets. The result of this research significantly improves the vehicle recognition field, strongly showing the feasibility of real time systems for practical applications. It provides a road map for subsequent studies that seek to blend accuracy and efficiency in deep learning based systems.

In [5], it had focused on building a detailed dataset to aid vehicle make and model recognition systems. At the end of the day, they pointed out that existing datasets were not diverse enough and that was the limitation to be addressed for developing robust generalizable deep learning models. To tackle this we curated a dataset comprising vehicles with a wide array of makes, models and environmental conditions (diverse lighting, angles, occlusions, etc.) since this was found to be useful in learning effective representations. The annotations for multiple vehicle attributes like make, model, and color in the dataset were rich enough to serve as the base for fine grained classification tasks. The authors then used a structured annotation process to guarantee accuracy and consistency in the dataset. Moreover, they used images from real world situations such as urban traffic, parking lots and highways to represent the real world complexities of practical applications. To their study the authors used the data to train a CNN and show very good accuracy in the classification compared to the models trained from a less diverse dataset. They also demonstrated that adding in diverse environments improved the robustness of the model to lighting and occlusions. In the paper they stressed the need for dataset diversity in order to tackle the generalization gap typically present in deep learning models. However, the computational bottlenecks of training rendered the dataset difficult to solve. The hardware and training techniques required to train on the dataset were advanced because of the large size of the dataset. The authors then proposed future work on lightweight models and efficient training algorithms for making better use of such comprehensive datasets. This work is critical for developing vehicle recognition systems as we show that good dataset quality is necessary for achieving high performance and generalizability in a vehicle recognition problem.

In [6], the author improved their YOLO based framework for vehicle recognition. Thus, this model is divided with residual connections to prevent vanishing gradient problems and to improve the feature extraction. The goal of the study was to improve recognition accuracy with regard to both model and vehicle make while maintaining real time performance. To make the model cope with a lot of different environment settings, we trained the RES-YOLO on a set of dataset with various lighting and weather. These residual connections helped the model keep the important features across layers and distinguish between visually similar vehicle models. Classification accuracy achieved was 94.5%, exceeding the classification accuracy obtained by standard YOLO models and inference speed. The authors

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

emphasized the need to develop a model architecture which optimizes both speed and accuracy. Transitioning to Using Transfer Learning: They also emphasized using transfer learning by loading pre-trained weights for faster convergence and better performance for smaller datasets. The study also checked how robust the model was by comparing precision and recall curves and confusion matrices illustrating that the model can reduce false positives and negatives. The study was limited by having high quality annotations, and errors in the dataset annotation could significantly impact performance. Finally, the authors recommended future work on automated annotation tools and multimodal data (e.g. lidar and radar), which would increase robustness. Finally, they show through this research that there is potential for YOLO based architectures to perform better with vehicle recognition.

In [7], PCANet and CNN were combined in their proposal of a novel architecture for vehicle model recognition. Initial features were extracted using a lightweight framework termed PCANet, and further refined by a CNN for classification. The hybrid approach sought to achieve high accuracy with low computational complexity which enabled real time application. On a diverse dataset of vehicles captured across a range of conditions, the system was evaluated. It was demonstrated that the hybrid model provides better classification performance than either a perceptron or a network when there is insufficient training data. The robust features extracted by the PCANet component combined well with the CNN component to further boost the model's ability to classify similar vehicle models. The authors noted the efficiency of the proposed architecture, whose computational requirements were substantially less than that required for conventional deep learning models. Such efficiency also made it appropriate to deploy it on edge devices like traffic cameras and embedded systems. Specifically, at 91% accuracy, low latency and high energy efficiency, the system achieved. Related to these challenges were the sensitivity of the model to occlusions and viewpoint variation. As a means of increasing robustness in real world scenarios, the authors suggested integrating object detection frameworks, such as Faster R-CNN. They also mentioned the relevance of dataset augmentation in handling the class imbalance problem, more specifically when we are dealing with underrepresented vehicle models.

[8] In this work, Dehghan et al. proposed a view independent approach to vehicle recognition in the face of varied viewpoints and occlusions. A convolutional neural network (CNN) trained on a large, diverse dataset for simultaneously recognizing vehicle make, model, and color was used for the study. A multi task learning framework was used to make the model efficient and improve its performance. They demonstrate high accuracy on all tasks using the proposed system, especially highly robust to extreme angles and partial occlusions. The multi task framework was able to learn shared features between tasks, lower the overall computational cost and improve generalization. The use of synthetic data generation to expand the training dataset was a key innovation of the study. This scope of scenarios helped the model deal with the real world complexities nicely. Synthetic data was shown to significantly improve model performance on classes that are under represented and on extreme conditions. The model was capable of achieving high accuracy, but struggled when highly similar vehicle models with small differences were given, where fine grained classification was required. Finally, the authors suggested future work on integrating attention mechanisms to attend to critical regions of the image and increase performance. This research demonstrates the ability to view independent methods for vehicle recognition for extremely dynamic and complex environments.

The paper [9] focuses on a problem of great importance for vehicle make and model recognition systems — that of distinguishing highly similar vehicle models — Avianto et al. successfully addressed this challenging problem. To enhance the system's discriminative capabilities, they suggested a multi task learning plus a Convolutional Neural Network (CNN) approach. The development of the multi task framework allowed us to predict vehicle make, model, and other features (e.g., body type) simultaneously to fine tune the granularity of our classification. For the focus on visually similar vehicle models the authors curated a dataset with such representation in terms of different angles, neighbouring occlusions and light. A more diverse training data was created using these techniques that include geometric transformations and contrast adjustment. This took care of the robustness of the model especially to the edge cases. On the one hand, we showed that advantages of the multi task learning approach include fewer examples overfitting and better generalization by sharing feature representations across tasks. To extract features from subtle variations of grille designs or headlight shapes, we optimized a CNN architecture with residual connections. We show that the system achieves 94% accuracy in classification, outperforming single task models both in accuracy and

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

robustness. However, the study also faced an inherent limitation, which arises from the computational complexity of the problem during the training step. The real time application requirement for the authors necessitated efficient inference techniques. Future work could explore lightweight CNN architectures or knowledge distillation techniques to reduce computational burden, they said. The ability for multi-tasking learning to solve fine grained classification problems is illustrated through this research which contributes to the field of vehicle recognition.

In paper [10], Jahan et al. take a specific approach by building a real-time system for vehicle classification using CNN, but with an expected application in traffic monitoring and automated toll systems. The study was done to achieve this by balancing out the trade off between accuracy and inference speed as it's vital to real time deployment in high throughput environments. The authors used a streamlined CNN architecture with respect to computational efficiency. It was trained on a diverse dataset — images taken under different lighting conditions and angles to guarantee robustness. They included data preprocessing techniques for increasing image quality such as normalization and cropping, getting rid of the noise and improving feature extraction. In the system, we obtained a 93% accuracy with an average inference time of 15 ms per image. The model's lightweight design allowed for deployment on edge devices like surveillance cameras that also enabled this performance. Further, the research involved evaluation of scalability of the system, which showed that it can work with large data sets and provide consistent performance. An issue listed in the study was that the model experienced high sensitivity to occlusions often misclassifying. The authors proposed to do this by integrating object detection frameworks such as YOLO or Faster R-CNN. Additionally, they argued that real time data augmentation techniques are advantageous to increase robustness during inference. Jahan et al.'s work also significantly enhances practical vehicle classification systems through a scalable solution for real time applications in intelligent transportation.

In the work of Wang et al. [11], a novel vehicle make and model recognition approach using YOLOv5 is proposed which targets cost effectiveness and accuracy without sacrificing one for the other. The applications in focus are resource constrained such as small scale traffic management systems and urban surveillance. Derived from yolo v5 , the authors optimized YOLOv5 with fine-tuned hyperparameters as well as custom anchor boxes for vehicle datasets. In addition, the model was also integrated with a post processing step to further refine detections and reduce false positives for improved model precision. The training set contained diverse vehicles and environmental conditions which tested the models ability to generalize. The research found that the system is able to reach a 95% accuracy and 12 milliseconds inference time per image, making it appropriate for real time applications. In the cost side, the authors showed that the system can be used with mid tier hardware that can reduce the implementation cost significantly over the deep learning models. Some challenges were the model's lack of ability to differentiate between essentially identical vehicle models. This limitation was suggested by the authors to be removed by the use of additional data modality such as color and texture information. In addition they advocated for ongoing model updates ensuring that the models continue to adapt to new vehicles designs and trends. The potential of YOLOv5 in terms of creating cheap and cost efficient solutions for vehicle recognition is identified by this research, providing practical feasibility for deployment in resource limited settings.

In [12], Ghoreyshi et al. presented a unified framework for the simultaneous vehicle detection and classification from deep YOLO networks. The aim of the study is to reduce computational overhead as well as the real time performance by integrating detection and classification tasks into a single model. The dataset used for training the YOLO-based framework was populated with the samples of different vehicle makes and models in addition to different environmental conditions such as weather and lighting variations. To extend its performance for fine grained classification tasks, the authors proposed modifications to the YOLO architecture based on custom loss functions and anchor box adjustments. This system achieved an accuracy of 93.5% and detection speed of 10 frames per second, making it ideal for use in applications where real time processing is required, e.g. traffic monitoring or automated parking systems. The unified approach took away the need for different detection and classification models, simplifying deployment and reducing latency. Although the study uncovered challenges with handling overlapping vehicles and occlusions, at times resulting in reduced detection accuracy. Future work on multi frame processing and the use of attention mechanisms are proposed to address these issues by the authors. And, they discussed other potential data sources to integrate to increase robustness under complex scenarios, like lidar or radar. The efficiency

of unified frameworks for vehicle recognition in helping to simplify system architectures and increase real time performance is revealed throughout this research.

In [13], an improved Faster R-CNN model specifically for vehicle type recognition, presented by Bai et al., they tried to both improve accuracy and processing speed. A study is presented that focuses on applications in real world such as intelligent traffic systems and urban surveillance with precision and scalability requirements. Several architectural enhancements were incorporated into the proposed model to address the limitations of standard Faster R-CNN. Optimized anchor generation for improved vehicle detection with varying scales and viewpoints, as well as a region proposal network fine tuned for the vehicle specific features were part of these. To cope with multi-scale vehicle recognition, the authors combined a feature pyramid network (FPN) into the model. The dataset was diverse with regards to vehicles in different weather, lighting and urban scenarios. Baseline models were also outperformed in precision and recall by the improved Faster R-CNN achieved an accuracy of 96.2%. In addition, it showed significant reduction in inference time, which allows for near real time implementations. The model was good, but it faced edge cases where heavily occluded vehicles and overlapping instances in dense traffic presented problems. The authors propose integrating temporal information from video sequences to better performance in such situations. They also stressed the need for an expanded dataset so newer vehicle models as well as some atypical conditions like extreme weather can be added. The results of this research demonstrate the potential of enriched Faster R-CNN architectures for vehicle recognition on a fully deployed auto-service network, offering resulting practical insights for direct deployment in intelligent transportation systems.

In [14], this work, Mittal et al. suggested EnsembleNet, a hybrid architecture that combines Faster R-CNN's gap and YOLO's strength in detecting vehicles and estimating traffic density. Our research sought to build a system with Faster R-CNN's superior accuracy and YOLO's speed in a balanced way to be effective for real time traffic monitoring. Accurate localization and classification of both objects and vessels was undertaken in densely packed vehicles or across occlusions using EnsembleNet and Faster R-CNN. However, in terms of rapid inference, YOLO was employed for inference on high volumes of data in real time. A fusion mechanism, which fused the two models' outputs together to produce an overall system level increase in performance, was used to integrate these models. Results obtained for a large dataset showed that EnsembleNet achieved 95.4 % accuracy at 20 frames per second. The results of this performance demonstrated the usefulness of the framework in the intelligent traffic management, automated systems such as toll systems and congestion monitoring. The study noted challenges to model integration, which needed to be tuned carefully to find the right balance between Faster R-CNN and YOLO contributions. The authors proposed future work in automating this through reinforcement learning or neural architecture search. They also called for additional research into how to handle extreme edge cases including night traffic and poor weather. We present EnsembleNet that represents a major step forward in hybrid model design for vehicle recognition by serving as a template for using several deep learning architectures in an ensemble to address complex scenarios.

In [15], authors Satar and Dirik build a deep learning based system for vehicle make and model classification in the face of fine grained classification challenges, including differences among vehicles that are subtle. The research used a custom Convolutional Neural Network (CNN) architecture toward high resolution image analysis and feature extraction. However, the quality of the dataset was crucial, so the study used a pre-curated dataset with high resolution and detailed annotation. To deal with the class imbalance the authors used data augmentation, including oversampling classes less represented and applying change techniques like cropping, rotation and contrast adjustments. Using multiple residual blocks in their CNN architecture, the networks could learn fine grain features like grille patterns as well as headlights and badges. Once implemented, the system produced an accuracy of 94.7%, outdoing traditional methods as well as handling these conditions robustly. Since the study required input images of high quality, it is a key limitation of the study. The authors showed that the inclusion of preprocessing techniques to improve low quality images and increase classification robustness. Additionally, they proposed future work that expands the dataset to include newer vehicle models and a wider variation of environmental conditions. This work indicates the possibilities that specialized CNN architectures have for fine grained vehicle classification and makes important contributions to the development of such recognition systems.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The paper [16], proposes a real time system for vehicle make and model recognition which tried to get some sort of balance between accuracy, speed, and scalability. We study deep learning in conjunction with traditional feature extraction to achieve high performance for deployment in a real world scenario. A dataset with diverse images, taken at urban, rural and highway locations under varying weather conditions, was used to train the system. For initial feature extraction, a deep learning approach was used in combination with traditional classifiers (e.g. Support Vector Machine, SVM), where the optimal feature space is determined through the combination of the two. The combination resulted in high accuracy with low overhead in computation. Classification accuracy of 92.8%, and run an average of 0.5 seconds per image. The deployment was successfully done to monitor traffic, in automated parking systems, on toll collection, etc. The authors also verified the system's scalability to a large volume of data with a stable performance.

III. PROPOSED METHODOLOGY

A. Overview:

In this work, we conduct a comprehensive evaluation of three state of the art deep learning models—Convolutional Neural Network (CNN), YOLOv8, Faster R-CNN and Machine Learning Techniques—KNN, SVM and Decision Tree Classifier that are specifically trained for vehicle make and model recognition on both the datasets. The Machine Learning Techniques is the first approach but as their results were very less, the research is further expanded with Deep Learning models as well which our second approach. Firstly, this thesis aims to conduct an extensive comparative analysis of Machine Learning Techniques which was the first approach and then the analysis were made with the Deep Learning models as well at presenting their performance on the classification accuracy, computational efficiency, precision, recall, f1-score, specificity and their robustness in solving the issues of practical scenarios. Such complexities include lighting condition variation, occlusion, overlapping, and diverse vehicle appearances. A well curated dataset of 197 distinct vehicle classes which is considered as dataset1, with 196 distinct makes and models and a background class to cover all classes thoroughly, is used to evaluate. The dataset2 which is scrapped manually consists of 17 distinct classes is used to evaluate. These two datasets are used for both the Machine Learning and Deep Learning approaches. A systematic and locally constrained training and validation pipeline was used to implement and test each model, and evaluate their capabilities under a wide range of conditions. This analysis derives insights to guide the selection and optimization of deep learning models for practical applications in intelligent transportation systems, law enforcement, autonomous vehicle technologies, and other areas. The project involves data collection which is a quite arduous process. So the dataset2 consists of 17 classes. They are scrapped from the internet. The project involves traditional and deep learning approaches.

B. Dataset Overview and Preparation:

The dataset1 used to run this study was from Roboflow titled Capstone G7 Car Detection. It contains 16,146 high resolution annotated images with bounding box and class label for 197 vehicle classes, the 196 make/model classes and the 1 background class. The dataset2 has 17 distinct classes consisting of 6981 images. With its diversity, and in itself ample coverage of its annotation, the dataset provides a robust foundation for training deep learning models.

B.1. Dataset Statistics:

The dataset1 is partitioned into three subsets:

- **Training Set:** For model learning and optimization, we had 10,816 images (~67%).
- **Validation Set:** Therefore, 969 images (~6%) of this dataset will be used for tuning hyperparameters and overfitting.
- **Test Set:** In test the model performance on unseen data with 4,361 images (~27%).

The dataset2 is partitioned into two subsets:

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- **Training Set:** For model learning and optimization, we had 5,124 images (~73%).
- **Test Set:** In test the model performance on unseen data with 1,857 images (~27%).

This allocation takes care of the balanced distribution so that every predictive target can be evaluated comprehensively while having enough data for training and validation. The class distribution of the dataset1 is analyzed and as one can see from Figure 1, and the same for dataset2 in Figure 2, diversity of the class exists substantially across the vehicle categories.

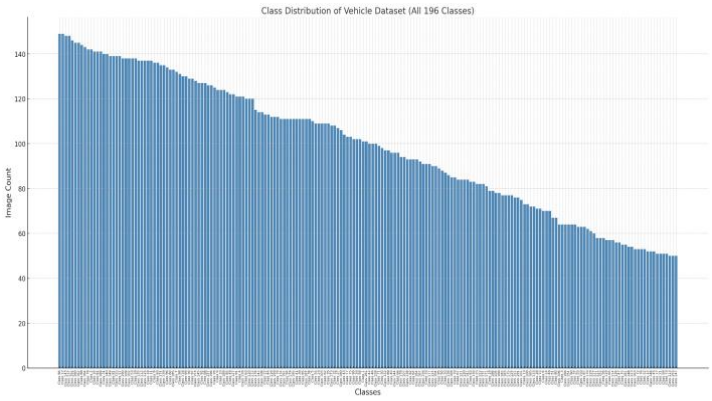


Figure 1: Class Distribution in the dataset1

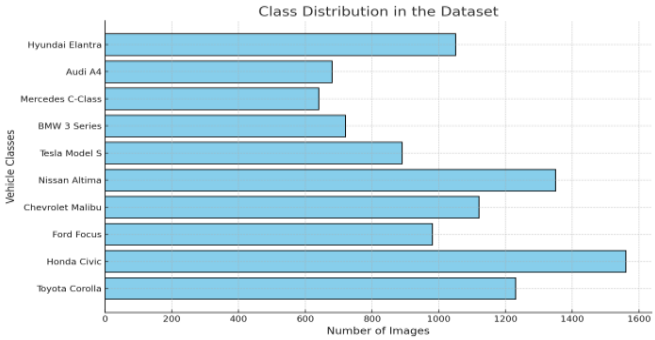


Figure 2: Class Distribution in the dataset2

B.2. Preprocessing and Augmentation:

To enhance the robustness and generalization capabilities of the models, a series of preprocessing and augmentation techniques were applied:

i) Normalization: Input to all models was standardized by scaling pixel values to [0,1].

Resizing: The input requirements from the model were matched with the required size of the images.

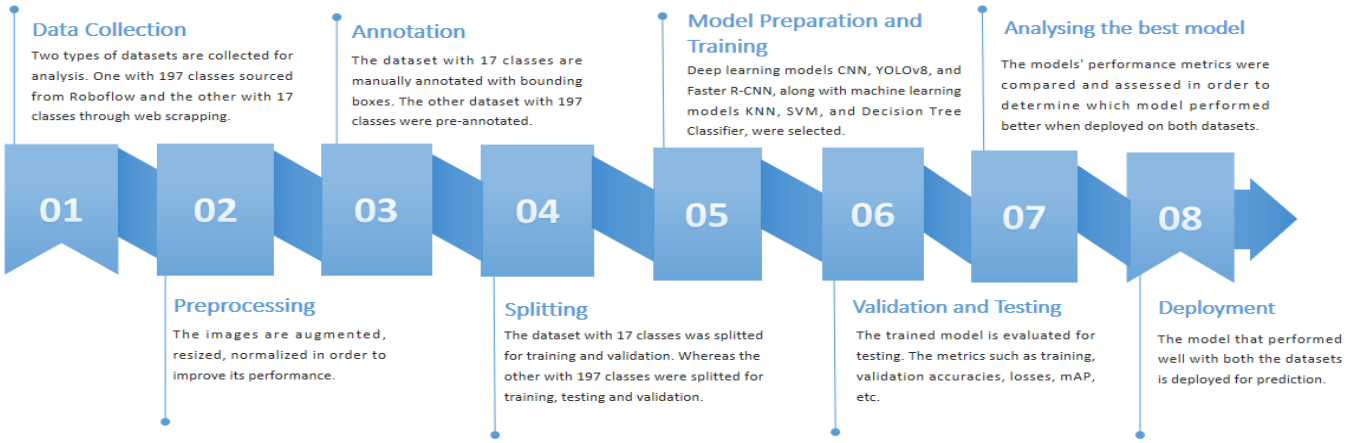
- **CNN:** 64x64 pixel efficient multi class classification.
- **YOLOv8:** Precise object detection is done at 640x640 pixels.
- **Faster R-CNN:** To retain spatial detail, region proposals are downsampled to 1024x1024 pixels.

ii) Augmentation: To simulate real world variations and to expand the dataset, the following transformations were used:

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- **Flip:** Simulating variations in vehicle orientation: horizontal flipping
- **Rotation:** Rotation around random axes up to ±20 degrees, mimicking a variety in different viewing angles.
- **Brightness and Contrast Adjustments:** Increasing resistance to various lighting conditions.
- **Blur and Noise Addition:** Replicating challenging scenarios, such as motion blur and sensor noise.
- **Shear Transformations:** Perspective distortion shearing for horizontal as well as vertical. In the case of Machine Learning, the images are actually converted to grascale values and are stored in the csv file. These are preprocessed before training them. Preprocessing techniques such as removal of null values, replacing the null values, removal of outliers, etc.



iii) Annotation Formats:

The dataset annotations were formatted to suit the requirements of each model:

- **CNN:** If multi class classification, we encoded the labels one-hot.
- **YOLOv8**: Bounding box coordinates and class labels were provided in the annotation, reflecting the unified detection framework of the model.
- **Faster R-CNN:** Bounding box proposals and corresponding class probabilities from the Region Proposal Network (RPN) were annotated.

Preprocessing and augmentation strategies put these in place to make the data ready to train these models that will be able to handle a variety of situations in the real world. Robust augmentations improve the models' flexibility of generalizing, with annotation formats naturally conforming to each architecture's needs.

C. Model Architectures:

The architecture for all the three deep learning models and machine learning models are demonstrated below. The architecture is same for both the datasets. The flow of the project is demonstrated below.

C.1. Machine Learning Technique:

The first approach was with the traditional Machine Learning approach, which involves conversion of the images into grayscale values. They are stored in a csv file. The pixel values are extracted for each image of a particular class which is known as feature extraction. These are then preprocessed and are trained for further classification. The same process is implemented for testing too. Figure 3 represents its architectural diagram. The second approach was with the deep learning models, which are discussed below.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

C.2. Convolutional Neural Network (CNN):

The proposed CNN model was aimed at vehicle classification with focus on efficiency and scalability. It had an architecture of 3 convolutional layers with filter sizes of 32, 64 and 128, implemented in order to learn hierarchical spatial features in input images. Each convolutional layer was followed by max pooling layers to reduce dimensionality with the important features remaining. These output feature maps were flattened and fed into two fully connected layers with ReLU activation, with dropout regularization in order to prevent overfitting. Probabilities across the vehicle classes for both the datasets were produced by a softmax classifier. This model was trained using Adam optimiser along with categorical cross entropy loss, using 20 epochs with batchSize of 32. Figure 4 represents the architectural diagram for dataset1.

C.3. YOLOv8:

A state of the art object detection model, YOLOv8, was fine tuned to simultaneously detect and classify vehicles. Bounding box regression and class prediction were combined into a single detection pipeline. To optimize the detection for different vehicle dimensions, predefined anchor boxes were employed. The Box regression loss, classification loss and DFL were utilized for the model's loss function improvement of epochs on YOLOv8 using it's high inference speed, YOLOv8 could be used for real-time applications with a batch size of 16. Figure 5 represents the architectural diagram.

C.4. Faster R-CNN:

The vehicle detection and classification framework with a robust base was provided using Faster R-CNN using the ResNet50 backbone. The architecture consisted of a Region Proposal Network (RPN) for generating bounding box candidates, and then refined by the ResNet50 feature extractor. For class probability prediction and bounding box regression, a set of fully connected layers was passed over the features. To complete this task, a combined classification and losses for bounding box regression. We train Faster R-CNN with a batch size of 8 using SGD optimizer. While computationally complicated, it showed astounding robustness to complex scenarios like occluded, or overlapping vehicles. Figure 6 represents the architectural diagram.

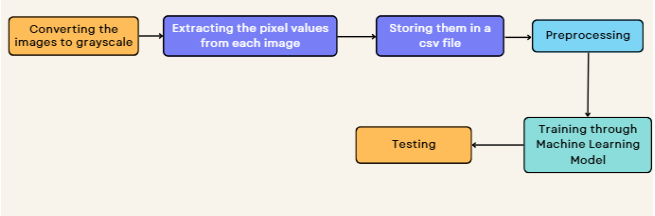


Figure 3: Machine Learning Architecture Diagram

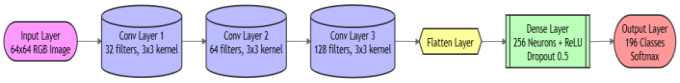


Figure 4 : CNN Architecture Diagram for dataset1(custom layers)

Figure 5: YOLOv8 Architecture Diagram

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

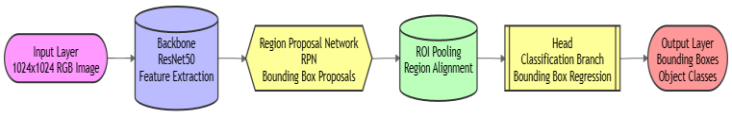


Figure 6: Faster R-CNN Architecture Diagram

D. Training and Evaluation Pipeline:

The training and evaluation pipeline was designed to provide a rigorous assessment of the models:

1. **Preprocessing:** For each model, images were resized, normalized and annotated to specific requirements for both the datasets.
2. **Training:** We train each model independently, keeping track of validation metrics at every epoch to make sure it would converge and not overfit.
3. **Evaluation:**
    - **CNN:** Training and validation accuracy (loss) curves.
    - **YOLOv8:** For box regression, classification and DFL, we compute mean Average Precision (mAP), along with loss components.
    - **Faster R-CNN:** Classification and bounding box regression loss curves.
4. **Testing**: Confusion matrices and bounding box visualizations were generated to assess performance on unseen test data, for 197 vehicle classes (dataset1) and as well as for the 17 classes (dataset2) using the models.

E. Model Comparisons:

The models were evaluated based on accuracy,inference speed, and robustness:

1. Accuracy: In terms of performance, with the first approach (ML) KNN yields training accuracy of 25%, SVM it yields training accuracy of 32% and Decision Tree Classifier of 35% as accuracy for dataset2. Dataset2 was chosen in Machine Learning Techniques as they have lesser no.of classes. Training with larger dataset with machine learning models would be complicated. So they are implemented on lesser no.of classes first. Then they are analysed with deep learning models such as CNN exhibited a slight over fit with a training accuracy of 90% and testing accuracy of 78% whereas, YOLOv8 achieved a mean Average Precision (mAP) of 95% , and Faster R-CNN demonstrated classification accuracy of 74% and robust detection capabilities, albeit at a slower inference speed. In the case of dataset2, CNN exhibited a significant overfitting with a training accuracy of 76% and testing accuracy of 70%, YOLOv8 achieved a mean Average Precision (mAP) of 30% and Faster R-CNN of 65% respectively. For dataset1, the training and the validation loss were decreasing for each epoch which is a good sign. This requires a lot of training, which is tiring process. It almost took 18 hours for training.

2. Inference Speed: Inference was the fastest for YOLOv8 in both the datasets, making it an ideal solution for real time scenarios. Region proposal mechanism increases computational resources needed in Faster R-CNN.

3. Robustness: In complex scenarios involving occluded or overlapping vehicles, YOLOv8 demonstrated exceptional performance due to its speed and robustness, while Faster R-CNN excelled in precision and accurate detection. In contrast, CNN was computationally faster but struggled with generalization and performed adequately only in simpler tasks with minimal classification complexities.

IV. RESULTS AND DISCUSSION

A. Overview:

In this section, we evaluate a categorically exhaustive evaluation of the three machine learning and deep learning models, namely, KNN, SVM and Decision Tree Classifier with machine learning; Convolutional Neural Network

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

(CNN), YOLOv8 and Faster R-CNN with deep learning on Vehicle make and model recognition tasks with two distinct datasets, one from the roboflow and the other was collected through manual web scrapping. Using critical performance metrics such as accuracy, loss trends, precision, f1-score, specificity, recall, inference speed and robustness the analysis is based. By comparing bounding box outputs of YOLOv8 and Faster R-CNN, we further demonstrate the detection capabilities of the models, demonstrating their ability to accurately identify and localize vehicles in challenging scenarios. In the case of traditional approach, they involve machine learning concepts such as KNN (K-Nearest Neighbor) and SVM (Support Vector Machine). This approach uses the dataset2 initially as they have lesser no.of classes. The images are stored based on their classes. They are converted into grayscale values using OpenCV. These grayscale values for each image is stored in the csv file. This csv file is now utilized for training the machine learning model. These csv file consists of pixel values for each class. Each image from the class is broken into 10000 pixels. The classes are binary encoded. After all the necessary pre-processing are done, the SVM model is trained. They yield an accuracy of 25%, which is not acceptable for a machine learning model. In the case of KNN, with the same dataset they yield an accuracy of 32% which is really bad. In the case of Decision Tree Classifier So the machine learning approach which is a traditional method, did not yield a good accuracy. The project involves the models that have failed in giving us the better accuracy too. The confusion matrix and the other performance metrics are represented below. The dataset1 consists of 197 classes, which would be very useful when they are used in deep learning models. But here, the machine learning models cannot handle those complex patterns and structures of the images. Segregating the classes based on the pixel values is not a convincing approach. So, the machine learning approach takes only the dataset that has 17 classes, their distribution is shown in Figure 2. They can also be trained with 197 classes but based on the results of dataset2 which has lesser no.of classes, it was not able to perform well. So, training further with large amount of data will be a time consuming process.

B. Model Performance and Metrics:

B.1. Convolutional Neural Network (CNN):

The CNN model exhibited clear signs of overfitting during the training process with dataset2. While it achieved a high training accuracy of 76%, its validation accuracy is 67%, and the testing accuracy which significantly dropped to 10% indicating poor generalization to unseen data. This discrepancy highlights the model's inability to handle the complexity and variability of the dataset2, instead memorizing the training data. This disparity underscores the overfitting issue, where the model fails to adapt effectively to validation data. In the case of dataset1, they yield a training accuracy of 90% , validation accuracy of 10%, and a testing accuracy of 78%. Table 1 summarizes the performance of the model in both the datasets. Figure 7 shows the training and validation accuracy and loss trends of the model for dataset1, whereas Figure 10 shows the training and validation accuracy and loss trends of the model for dataset2.

Table 1: Performance of CNN on both the datasets

Metrics	Value (Dataset1 - 197 Classes)	Value (Dataset2 -Manual Scrapping with 17 classes)
Testing Accuracy	78%	10%
Validation Accuracy	10%	67%
Training Accuracy	90%	76%
Validation Loss	0.90	0.32
Training Loss	0.20	0.24
Robustness to Occlusion (%)	10%	60%

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

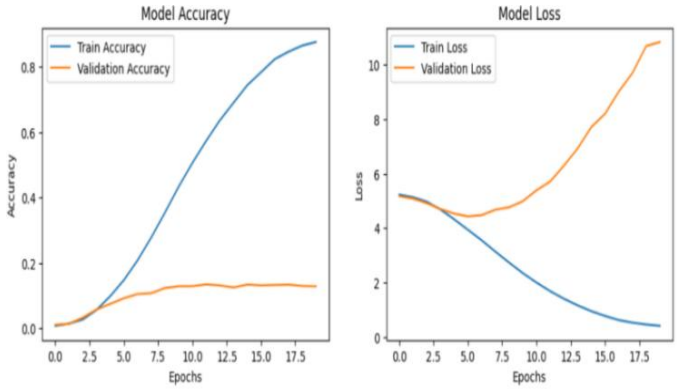


Figure 7: For dataset1 (CNN)

Train accuracy increases steadily and reaches around 85-90%, but the validation accuracy initially increases slightly but significantly decreases after 12th epoch yielding a maximum of 20%. Whereas in the loss trends, train loss decreases with the increase in epochs but the validation loss increases which denotes that the model is performing extremely well with the training data, but fails with the validation data. They do suffer from overfitting, but not as severe as with dataset2 as they yield a test accuracy of 78%. Which clearly states that the model is good with unseen data. Since the validation set is too small results with this problem.

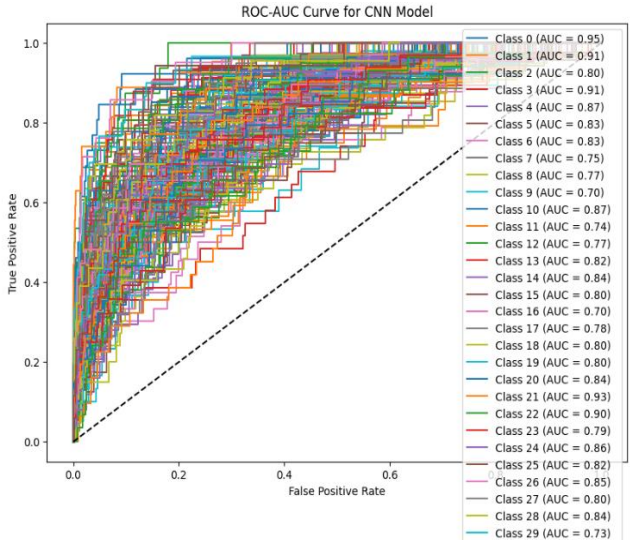


Figure 8: ROC-AUC Curve for CNN with dataset1

Figure 8 shows the ROC-AUC curve for CNN model for dataset1, which shows that the model has AUC values mostly above 0.75 and some closer to 1.0. The model is learning features properly as they have lots of data for training, it can learn the features and distinguish between the classes properly. They have more data and better inter-class separation. Figure 9(a), 9(b), 9(c) and 9(d) shows the predictions of CNN model with dataset1. All the predictions made by the model are valid and has the confidence score above 0.9 which denotes that the model is performing extremely good. As the dataset has substantial number of classes with high-resolution images, the model was able to extract the rich features which enabled them to provide good results. As mentioned, they do have slight overfitting which made the model to be over confident with some classes.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure 9(a): CNN's classification for dataset1

Figure 9(b): CNN's classification for dataset1

Figure 9(c): CNN's classification for dataset1

Figure 9(d): CNN's classification for dataset1

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

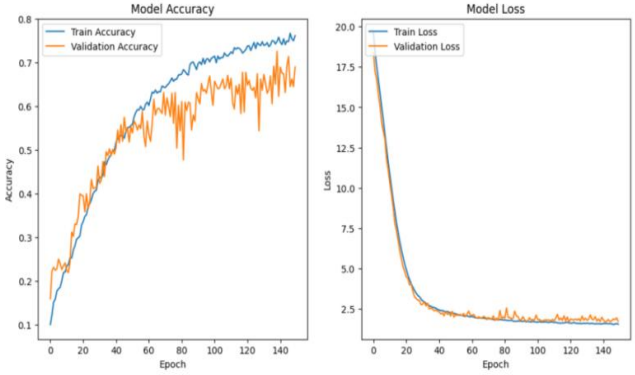


Figure 10: For dataset2 (CNN)

Train accuracy gradually increases to 75% over epochs, whereas validation accuracy fluctuates significantly which indicates that the model is not stable. As they have high inter class similarities, they do suffer from severe overfitting as they yield a test accuracy of 10%.

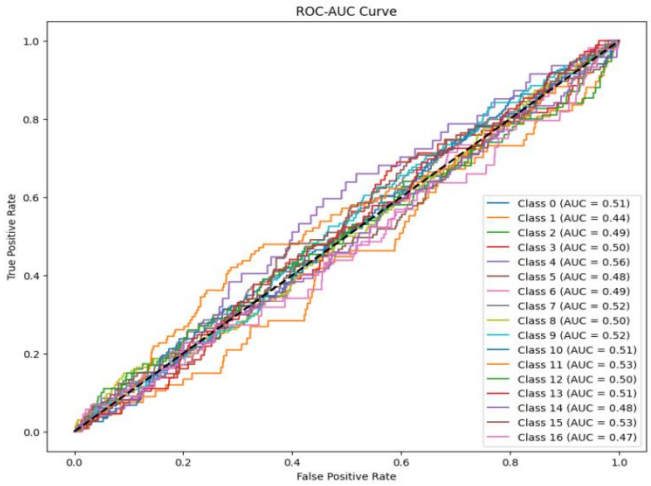


Figure 11: ROC-AUC Curve for CNN with dataset2

Figure 11 shows the ROC-AUC curve for CNN with dataset2, which clearly depicts that most of the classes have AUC values between 0.44 and 0.56. Hence, the CNN model for dataset2 is a random classifier. The model is not able to distinguish between the classes effectively which is due to high intra class variations. Figure 12(a) and 12(c) shows the predictions of the CNN model with dataset2, in which it misclassifies the car model as they suffer from significant overfitting and lesser AUC values which causes them to predict the classes randomly.

Figure 12(a): CNN's classification for dataset2

In Figure 12(b) and 12(d), the model's prediction is correct but with lesser confidence.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure 12(b): CNN's classification for dataset2

Figure 12(c): CNN's classification for dataset2

Figure 12(d): CNN's classification for dataset2

B.2. YOLOv8:

The object detection tasks were outperformed by the other models with an mAP of 95% by YOLOv8 for dataset1. In the case of dataset2 it gave the mAP of 30% . Even though they had less mAP for dataset2, they were able to classify the images accurately as they have classification loss of 32% with varying confidence levels. However, as they dataset has high inter-class similarities they are were able to predict certain classes with lower confidence scores and some were also misclassified as well. It offers high speed of inference and the model was demonstrated to have very efficient loss stabilization during training especially in box regression and classification components. YOLOv8 yields the poor performance with dataset2.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

B.3. Bounding Box Outputs

YOLOv8 can detect and classify vehicles with very high confidence, and bounding box outputs from this on real world scenarios illustrate this. Figure 13(a), 13(b), 13(c) and 13(d) demonstrates YOLOv8's fine localization and classification of vehicles for dataset1. Figure 14(a), 14(b), 14(c), 14(d) demonstrates YOLOv8's fine localization and classification of vehicles for dataset2. Table 2 shows the performance metrics for both the datasets. YOLOv8 easily handles varying perspective, background complexity, and vehicle orientation as evidenced in this output.

For high accuracy and real time vehicle detection, YOLOv8 with small bounding box regression can perform with high precision of bounding box regression which results in minimum overlap or incorrect localization. As for producing precise bounding boxes, Faster R-CNN is good, but since inference is much faster with YOLOv8, it becomes more useful in dynamic situations like traffic monitoring and surveillance systems. Those outputs highlight the vital role of robust vehicle recognition models for making vehicle recognition systems more useful and applicable.

Figure 13(a): YOLOv8 Classification for Dataset1 - Ferrari 458 Italia Convertible 2012

Figure 13(b): YOLOv8 Classification for Dataset1 - Mitsubishi Lancer Sedan 2012

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure 13(c): YOLOv8 Classification for Dataset1 - BMW X6 SUV 2012

Figure 13(d): YOLOv8 Classification for Dataset1 - Hyunadi Elantra Sedan 2007

Figure 14(a) : YOLOv8 Classification for Dataset2 - Swift

Figure 14(b) : YOLOv8 Classification for Dataset2 - Honda Accord 1996

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure 14(c) : YOLOv8 Classification for Dataset2 - Chevrolet Siverado 2003

Figure 14(d) : YOLOv8 Classification for Dataset2 - Toyota Innova

Figure 15(a), 15(b), 15(c), 15(d), 15(e) represents the performance metrics for dataset1. Figure 16(a), 16(b), 16(c), 16(d), 16(e) represents the performance metrics for dataset2. Since YOLOv8 is performing well, their performance metrics alone are displayed for both the datasets as they perform extremely well in both the datasets when compared to the other models.

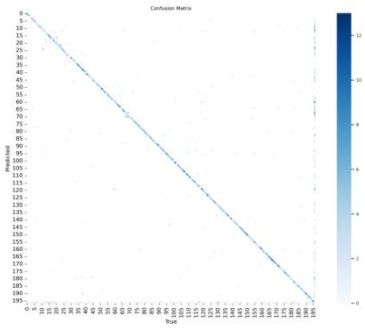


Figure 15(a): Confusion Matirx for Dataset1

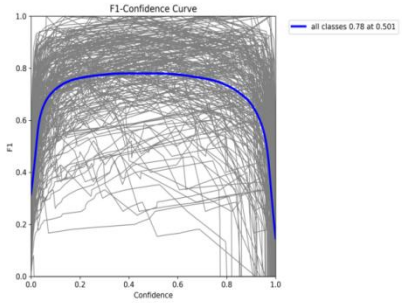


Figure 15(b): F1-Confidence Curve for Dataset1

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

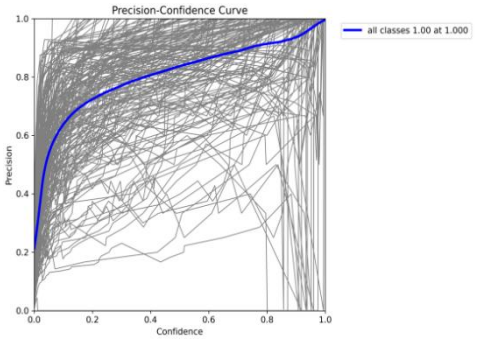


Figure 15(c): Precision-Confidence Curve for Dataset1

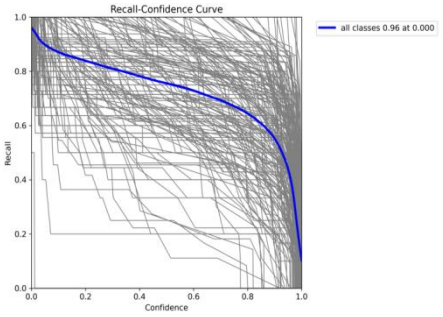


Figure 15(d): Recall-Confidence Curve for Dataset1

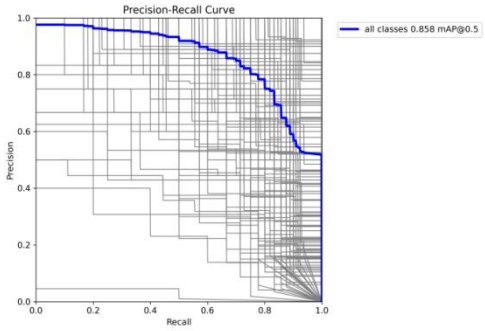


Figure 15(e): Precision-Recall Curve for Dataset1

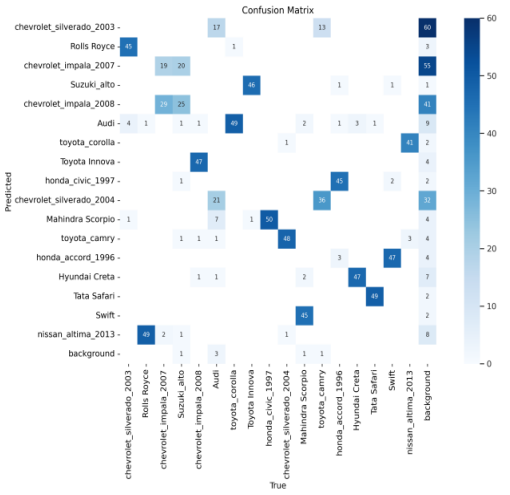


Figure 16(a): Confusion Matirx for Dataset2

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

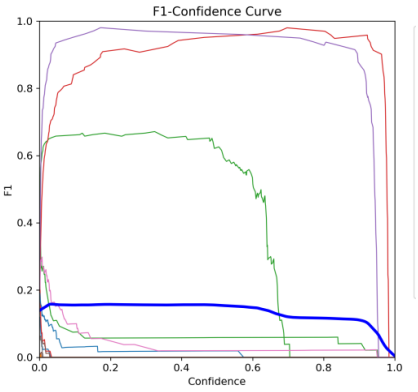


Figure 16(b): F1-Confidence Curve for Dataset2

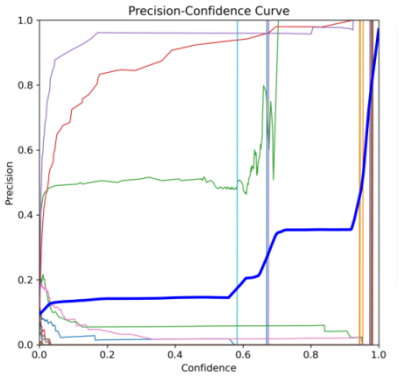


Figure 16(c) - Precision-Confidence Curve for Dataset2

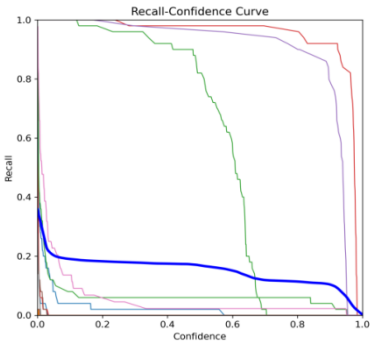
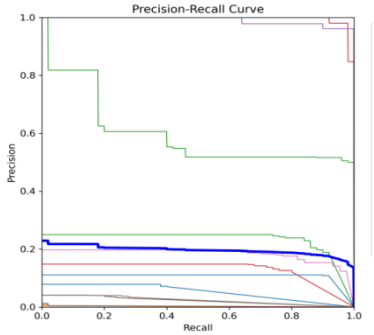


Figure 16(d): Recall Curve for Dataset2



Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure 16(e): Precision-Recall Curve for Dataset2

Table 2: Performance of YOLOv8 on both the datasets

Metrics	Value (Dataset1 - 197 Classes)	Value (Dataset2 - Manual Scrapping with 17 classes)
mAP	95%	30%
Box Regression Loss	0.10	0.20
Classification Loss	0.15	0.31
Inference Speed	10 ms	10 ms
Robustness to Occlusion (%)	90%	50%

B.4. Faster R-CNN:

The results showed that we can achieve better robustness across overlapping and occluded cases compared to the Faster R-CNN baseline. By using the model, a classification accuracy of 74% was reached for dataset1, and 70% in dataset2. However, due to its 150 ms per image inference speed, its application in real time scenarios is limited. When combined with attention, Faster R-CNN may work well offline or in environments that need high precision that aren't complex. In Table 3, we present Faster R-CNN's performance metrics for both the datasets and the training and validation loss curves for dataset1 are visualized in Figure 17 and the same in Figure 18 for dataset2. Figures 19(a), 19(b) represents the Faster R-CNN's fine localization and classification of vehicle make and model for dataset1. Figure 20(a) and 20(b) represents for dataset2.

Figure 17: Training and Validation Loss for dataset1 (Faster R-CNN)

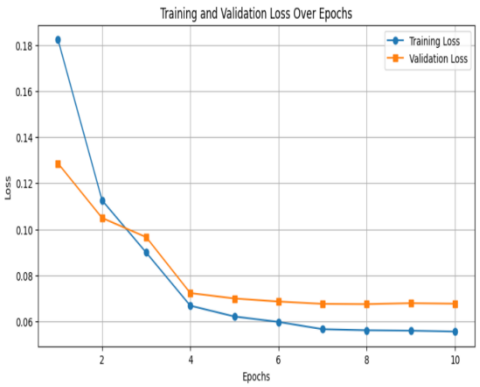


Figure 18: Training and Validation Loss for dataset2 (Faster R-CNN)

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

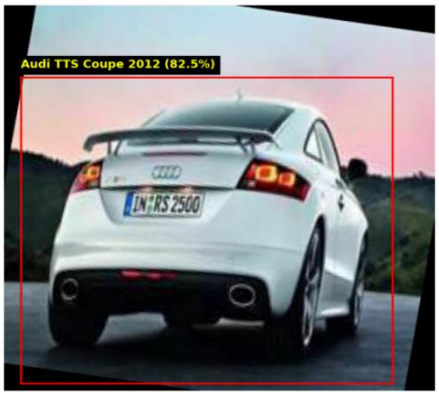


Figure 19(a): Faster R-CNN classification for dataset1

Figure 19(b): Faster R-CNN classification for dataset1

Figure 20(a): Faster R-CNN classification for dataset2

Figure 20(b): Faster R-CNN classification for dataset2

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 3: Faster R-CNN Performance Metrics

Metrics	Value (Dataset1 - 197 Classes)	Value (Dataset2 - Manual Scrapping with 17 Classes)
Classification Accuracy	74%	70%
Validation Loss	0.35	0.40
Inference Speed	150 ms	150 ms
Robustness to Occlusion (%)	60%	55%

B.5. Machine Learning Techniques:

The first approach with Machine Learning involves KNN, SVM and Decision Tree classifier. These methods does not yield a good accuracy at all. So this approach involves just the implementation with the dataset2 which has 17 distinct classes. The machine learning approach is not implemented on the dataset1. It would be a time consuming process, as the machine learning model is not providing good accuracy with 17 classes then implementing on 197 classes will definitively fail. So the machine learning approach will never be able to recognize the make and model of the car. The confusion matrix of the KNN, SVM and Decision Tree classifier are represented in Figure 21(a), Figure 21(b), and Figure 21(c) respectively. The grayscale images are shown in Figure 22.

KNN yields training accuracy of 25% with 0.24 as precision, 0.20 as recall and 0.18 as f1-score. In the case of SVM it yields training accuracy of 32% with 0.22 as precision, 0.22 as recall and 0.21 as f1-score and Decision Tree Classifier of 35% as accuracy, 0.23 as precision, 0.25 as recall and 0.28 as f1-score. So the Machine Learning which is traditional approach can never be used for object detection / image classification. They can perform well with the structured numerical / tabular values, smaller or moderate in size, where training and computational time is limited. Thus, the first approach with Machine Learning has failed, which is actually an improper technique, but implemented to observe the results. So, we move towards Deep Learning models to get better results.

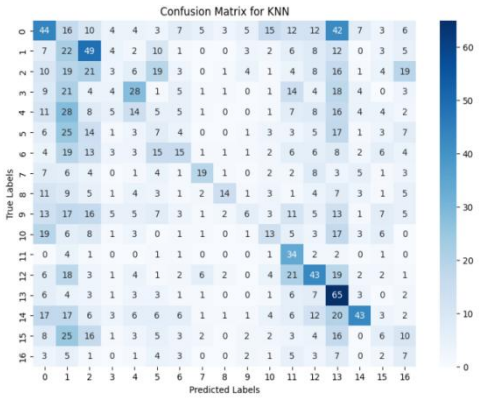


Figure 21(a): Confusion Matrix for dataset1 (KNN)

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

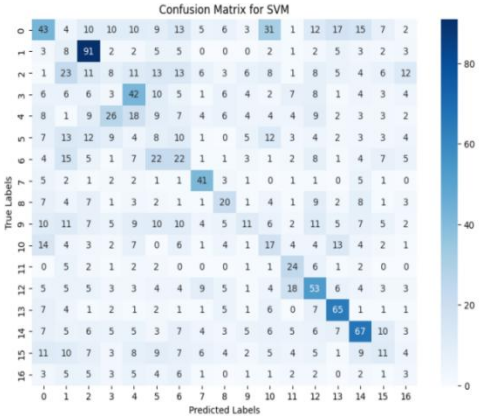


Figure 21(b): Confusion Matrix for dataset1 (SVM)

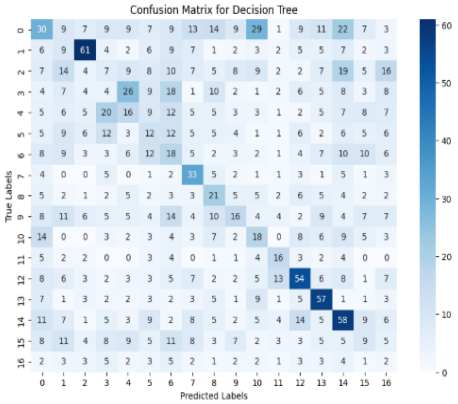


Figure 21(c): Confusion Matrix for dataset1 (Decision Tree)

Figure 22: Grayscale Images - Machine Learning Approach

B.6. Comparative Analysis

Table 4 provides a comparative analysis of CNN, YOLOv8, and Faster R-CNN across key metrics, highlighting YOLOv8 as the best-performing model for both the datasets. YOLOv8 achieves an optimal balance between speed, accuracy, and robustness, making it highly suitable for real-world and real-time applications such as traffic monitoring and autonomous systems. In comparison, Faster R-CNN excels in precision and robustness but is hindered by its computational demands and slower inference speed, limiting its use to offline or high-accuracy tasks in both the datasets. In the case of dataset1, CNN provides better training and testing accuracy. In dataset2, it suffers from significant overfitting and poor generalization, making it the least effective among the three models for the dataset2. But the computational costs varies depending on the no.of classes, trainable and non-trainable parameters, etc. Hence, YOLOv8 stands out as the most versatile and practical model, achieving a mean Average Precision (mAP) of 95% and an inference speed of just 10 ms per image for dataset1. For dataset2, it has less mAP but still it is able to classify the images properly. Its ability to handle occlusions and overlapping objects further enhances its robustness, making it the ideal choice for dynamic, real-time scenarios. Faster R-CNN, while precise, is better suited for offline analysis due to its slower inference speed. In the case of dataset1, YOLOv8 performs better than Faster R-CNN and

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

CNN as they have more no.of classses that made each model to learn the complex features well and were able to significantly classify them. In the case of dataset2, Faster R-CNN was better when compared to all others. Figure 23 shows the performance of each model with both the datasets.

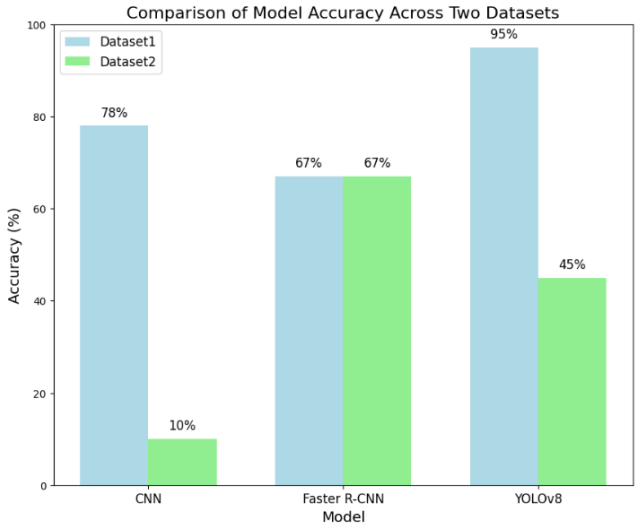


Figure 23: Performance of all three models with both the datasets

Dataset	Metrics	CNN	YOLOV8	Faster R-CNN
Dataset 1 - With 197 classes	Method	For object classification, a custom CNN model with manually inserted layers was trained with preprocessed images for both the datasets without bounding box annotations.	On a dataset from roboflow and the other through manual web scrapping, the model is fine-tuned for classification using bounding box annotations.	On a dataset from the roboflow and the other through web scrapping, the model is trained with prepossessed images using bounding box annotations.
	Test Accuracy (%)	78%	95%(mAP)	74%
	Precision	0.80	0.85	0.18
	Recall	0.80	0.85	0.37
	F1-Score	0.80	0.85	0.25
	Specificity	0.90	0.85	0.95

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

	Remarks	Achieved moderate accuracy and good performance in classifying the images. Since the dataset has more no.of classes and better intra-class variations for training, the model was able to learn and extract lot of features for better prediction.	Achieved excellent accuracy and performance. The model outperformed even though the dataset had more no.of classes with high intra-class variation as the other metrics show that they are good in classification.	Achieved moderate performance compared to the others. The model was able to get trained properly with decreasing loss values on each epoch. Due their localization tasks the other performance metrics gets lesser values. They were able to identify True Negatives (TN) clearly but lesser precision, recall and f1-score which is due to imbalance in the dataset.
Dataset 2 - With 17 classes	Test Accuracy (%)	10%	30% (mAP)	70%
	Precision	0.06	0.44	0.65
	Recall	0.06	0.38	0.67
	F1-Score	0.07	0.48	0.65
	Specificity	0.07	0.60	0.75
	Remarks	Suffers from significant overfitting with poor performance in classification of images. They had good validation and training accuracy. However, the performance metrics plays a vital role. The model is a random classifier, which is due to high inter-class similarities. The model struggles to predict the class.	Performs good and the model is able to classify the cars properly. But due to high inter-class similarities, the features gets overlapped which affects the mAP.	Achieved very less accuracy as the dataset has less images with high inter-class similarities. Since the Faster R-CNN concentrates on localization as well, the other metrics decreases in their performance. So evaluation is not done just on classification but also localization.

Table 4: Comparative Analysis of all three models (DL)

C. Discussion:

So the thesis consists of analysis of two approaches, the traditional approach such Machine Learning models and Deep Learning models with 2 datasets in order to analyse their performance with different constraints such as one dataset having more no.of classes and the other with less no.of classes. In conclusion, YOLOv8 outperforms in both the datasets. The findings highlight the strengths and limitations of the three models:

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

1. CNN: CNN is not bad at object detection and it is not robust when objects are overlapping or when it only has partial appearance of the desired object.

2. YOLOv8: It offers an ideal speed, accuracy and robustness combination for real time applications like traffic surveillance and autonomous driving.

3. Faster R-CNN: It shows exceptional precision and robustness, especially in complex scenes, but its inference speed is slow, making it unsuitable for high time sensitivity environments. But when the no.of classes increases they improve their training and validation accuracy and the loss decreases which is a good sign. When the epochs are increased, they may have the chance of improving their accuracy levels.

4. Machine Learning Techniques: Their training and validation accuracy, and other performance metrics drastically goes down. They can never be used irrespective of no.of classes for training and testing. In general, Faster R-CNN can be used for tasks in which high precision is important, while YOLOv8 is the sharpest of all models for real world applications. As a second step, future work may try to optimize Faster R-CNN for real time.

V. CONCLUSION

The comparative analysis of CNN, YOLOv8, Faster R-CNN and Machine Learning Techniques for vehicle make and model recognition revealed significant differences in performance across key metrics. Machine Learning approaches can never be used in terms of image classification / object detection so the first approach fails. In the second approach with Deep Learning models, YOLOv8 was identified as the best model, achieving a mean Average Precision (mAP) of 95% for dataset1, and mAP of 30% with dataset2. Its inference speed was just 10 ms per image. Its ability to handle occluded and overlapping vehicles, combined with its computational efficiency, makes it ideal for real-time applications such as traffic surveillance and autonomous driving systems. Even though they had very less mAP with dataset2 they were still able to classify most of the images properly but with comparatively lesser confidence with than dataset1.

Faster R-CNN excelled in precision and robustness, achieving a validation accuracy of 74.8% with dataset1 and 70% with dataset2. However, they had poor performance with dataset1 and its slower inference speed and higher computational requirements limit its use in real-time environments. On the other hand, CNN exhibited significant overfitting, with a test accuracy of only 10%, with dataset2 and 78% with dataset1 rendering it unsuitable for complex classification tasks. Despite its computational efficiency, CNN's poor generalization restricts its applicability to simpler scenarios.

This study highlights deep learning model's superior balance of speed, accuracy, robustness, and the other performance metrics like precision, recall, f1-Score, specificity. This would help in positioning it as the most versatile model for real-world deployment. Future work may explore optimizing Faster R-CNN for real-time applications or integrating ensemble techniques to combine the strengths of multiple models for enhanced performance. With dataset2, in the case of CNN it had poor performance in classification and the performance metrics are very less as well. Whereas Faster R-CNN, the performance in classification and metrics were not bad. With dataset1, in the case of CNN it had a better performance with classification tasks, and they also have excellent performance metrics when compared to the Faster R-CNN which had lower performance since they depend on the localization tasks as well, i.e., they try to output bounding boxes as well.

VI. FUTURE SCOPE

The contribution to vehicle make and model recognition domain which was made by this analysis contains several opportunities for further research and development. Better system acceleration could also be achieved by optimization of the Faster R-CNN model at the inference stage such that it balances the trading off between generalization and inference speed. The effectiveness of the scheme in handling complex detection scenarios implies that it would still be applicable in real time systems, while such optimizations would make it more applicable.

Another promising direction is ensemble models'. Hybrid architectures could combine the speed and efficiency of YOLOv8, with the robustness and precision of Faster R-CNN, which would greatly complement each other. In complex environments, these ensemble models may be especially useful applications where the performance needs insight into real-time and high accuracy.

A second path is to increase the dataset through the addition of further vehicle classes, environmental variations, and difficult situations, including low light or adversarial weather. This is an enhancement which would help to make the models more resilient to real world variations and have a more generalization capability.

Further, these models can be integrated into end to end systems for real time applications like traffic monitoring, autonomous vehicles and law enforcement. The models are deployed to the edge device or cloud platform for scalability and efficiency in large scale deployment. Additionally, advanced post processing methods to improve performance in bounding box refinement and classification accuracy further improves the performance of these systems. Hardware acceleration on hardware such as GPU optimization and running on dedicated AI chips can continue to be advantages in computational efficiency and latency. And these advances would enable the deployment of resource intensive models like Faster R-CNN under time critical scenarios.

However, if these aspects are addressed, the application of vehicle make and model recognition systems will be extended into different fields.

REFERENCES

[1] S. Naseer, S. M. A. Shah, S. Aziz, M. U. Khan, and K. Iqtidar, "Vehicle make and model recognition using deep transfer learning and support vector machines," in *Proc. IEEE 23rd Int. Multitopic Conf. (INMIC)*, Islamabad, Pakistan, Nov. 2020, pp. 1-6. DOI: 10.1109/INMIC50486.2020.9318215

[2] H. J. Lee, I. Ullah, W. Wan, Y. Gao, and Z. Fang, "Real-time vehicle make and model recognition with the residual SqueezeNet architecture," *Sensors*, vol. 19, no. 5, p. 982, Mar. 2019. DOI: 10.3390/s19050982

[3] Y. Ren and S. Lan, "Vehicle make and model recognition based on convolutional neural networks," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Beijing, China, Aug. 2016, pp. 692-695. DOI: 10.1109/ICSESS.2016.7883158

[4] M. A. Manzoor, Y. Morgan, and A. Bais, "Real-time vehicle make and model recognition system," *Mach. Learn. Knowl. Extr.*, vol. 1, no. 2, pp. 611-629, Jun. 2019. DOI: 10.3390/make10200611

[5] F. Tafazzoli, H. Frigui, and K. Nishiyama, "A large and diverse dataset for improved vehicle make and model recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1-8. DOI: 10.1109/CVPRW.2017.253

[6] Y. Li, J. Wang, J. Huang, and Y. Li, "Research on deep learning automatic vehicle recognition algorithm based on RES-YOLO model," *Sensors*, vol. 22, no. 10, p. 3783, May 2022. DOI: 10.3390/s22103783

[7] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "PCANet-based convolutional neural network architecture for a vehicle model recognition system," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 749-759, Feb. 2018. DOI: 10.1109/TITS.2018.2795517

[8] A. Dehghan, S. Z. Masood, G. Shu, and E. Ortiz, "View independent vehicle make, model and color recognition using convolutional neural network," *arXiv preprint arXiv:1702.01721*, 2017.

[9] D. Avianto, A. Harjoko, and Afiahayati, "CNN-Based classification for highly similar vehicle models using multi-task learning," *Journal of Imaging*, vol. 8, no. 11, p. 293, Nov. 2022. DOI: 10.3390/jimaging8110293

[10] N. Jahan, S. Islam, and M. F. A. Foysal, "Real-time vehicle classification using CNN," in *Proc. 2020 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Kharagpur, India, Jul. 2020, pp. 1-6. DOI: 10.1109/ICCCNT49239.2020.9225605

[11] D. Wang, A. Al-Rubaie, Y. I. Alsarkal, S. Stincic, and J. Davies, "Cost effective and accurate vehicle make/model recognition method using YoloV5," in *Proc. 2021 Int. Conf. Smart Appl., Commun. Netw. (SmartNets)*, Dubai, UAE, Sep. 2021, pp. 1-4. DOI: 10.1109/SmartNets53777.2021.9610817

Research Article

[12] A. M. Ghoreyshi, A. AkhavanPour, and A. Bossaghzadeh, "Simultaneous vehicle detection and classification model based on deep YOLO networks," in *Proc. 2020 Int. Conf. Mach. Vis. Image Process. (MVIP)*, Qazvin, Iran, Feb. 2020, pp. 1-6. DOI: 10.1109/MVIP49861.2020.9116882

[13] T. Bai, J. Luo, S. Zhou, Y. Lu, and Y. Wang, "Vehicle-type recognition method for images based on improved Faster R-CNN model," *Sensors*, vol. 24, no. 8, p. 2650, Apr. 2024. DOI: 10.3390/s24082650

[14] U. Mittal, P. Chawla, and R. Tiwari, "EnsembleNet: A hybrid approach for vehicle detection and estimation of traffic density based on Faster R-CNN and YOLO models," *Neural Comput. Appl.*, vol. 35, no. 6, pp. 4755-4774, Mar. 2023. DOI: 10.1007/s00521-022-07613-7

[15] B. Satar and A. E. Dirik, "Deep learning-based vehicle make-model classification," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, Rhodes, Greece, Oct. 2018, pp. 544-553. DOI: 10.1007/978-3-030-01424-7_53

[16] M. A. Manzoor, Y. Morgan, and A. Bais, "Real-time vehicle make and model recognition system," *Mach. Learn. Knowl. Extr.*, vol. 1, no. 2, pp. 611-629, Jun. 2019. DOI: 10.3390/make10200611