SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I9P119

Volume 12 Issue 9, 182-192, September 2025

Original Article

Optimized Vehicle Recognition Using Convolutional Neural Networks: A Deep Learning Approach for Real-World Vehicle Detection Framework

Sanjay Pande[1], Minal Patil[2], Vinita Kakani[3], Abhishek Madankar[4], Roshan Umate [5], Pratik Agrawal[6], Nilesh Nagrale[7]

[1]*Computer Technology, Yeshwantrao Chavan College of Engineering, Maharashtra, India.*
[2,4]*Electronics and Telecommunication Engineering, Yeshwantrao Chavan College of Engineering, Maharashtra, India.*
[3]*School of Computer Science & Engineering, Ramdeobaba University, Maharashtra, India.*
[5]*Research Consultant DMIHER Maharashtra, India.*
[6,7]*Symbiosis Institute of Technology, Symbiosis International (Deemed University), Maharashtra, India.*

[4]*Corresponding Author : vicky.madankar123@gmail.com*

Received: 14 July 2025 Revised: 16 August 2025 Accepted: 15 September 2025 Published: 30 September 2025

Abstract *- Vehicle detection is an essential task that has many important applications, such as traffic monitoring, autonomous driving, and surveillance systems. In this paper, we implemented an efficient and robust vehicle detection using Convolutional Neural Networks. Utilization of a deep CNN architecture trained on 10,000 images of different vehicles, with the evaluation of mAP that reached 92.5% on a challenging test set of 2,000 images. The runtime is real-time with an image throughput of 30 Frames Per Second (FPS) on a commodity GPU. Methodology: Additionally, varying CNN models and optimization methods affect detection speed and accuracy. Trained on data as recent as jan 2025, results show that CNNs can be used to detect vehicles accurately and efficiently in the real world. Using mAP @0 5 of 94.3% and a mAP@0.95 of 78.6% has been held-out test set of 3,000 images. It accomplished a real-time processing rate of 45 Frames Per Second (FPS) on an NVIDIA GeForce RTX 3080 GPU, making it feasible for utilization in actual work. In addition, evaluation of the effect of data augmentation techniques, including random cropping, flipping, and color jitter, resulted in mAP@0.5 gains of 3.2%. The model was also combined with MATLAB's Computer Vision Toolbox to process video streams in order to distinguish vehicles in real time. Result: The system accomplished 25 FPS detection speed on 1080p content powered by a GPU setup. Furthermore, the models were capable of magnificently detecting the vehicles under different intricate situations like obstructions, variable illumination, and traffic density variability, resulting in a mean Intersection over Union (mIoU) of 0.92. This proposed CNN-based vehicle detection system achieves high accuracy and real-time performance, as demonstrated by the results.*

Keywords *- Convolutional Neural Networks (CNNs), MATLAB, Vehicle identification, Vehicle detection, Real-time image processing, Feature extraction, Deep learning, Frame Per Second (FPS).*

1. Introduction

Convolutional Neural Networks involve the transformation of computer vision, demonstrating exceptional effectiveness in tasks such as object identification and recognition. Their capability to autonomously attain hierarchical properties from raw image data makes them predominantly adept at multifaceted visual identification tasks, such as identifying cars. This is an example of the inventiveness of the usage of CNNs to recognize automobiles in the MATLAB scenario. Vehicle recognition has widespread applications, comprising traffic monitoring and surveillance for autonomous driving, as well as driver support technologies. These applications involve the capability to recognize and classify automobiles in pictures or video streams correctly. The examination of this study is to progress an enthusiastic and productive vehicle detection using CNNs implemented in MATLAB. This proposed method will use MATLAB's deep learning toolkit and image processing tools to train and evaluate the CNN models used in a relevant dataset. The objectives include achieving specific vehicle categorization and examining multiple CNN architectures to optimize the system for real-time efficacy. Furthermore, the novel method of classification involves distinguishing between different types and models of automobiles. Vehicle identification applications are numerous and substantial for vehicle identification systems, which monitor traffic flow, detect congestion, and enforce traffic laws. In surveillance systems, vehicle identification is used for security purposes, detecting and monitoring cars of interest. Self-driving cars can also rely deeply on vehicle detection to understand their environments and make safe decisions. Vehicle recognition empowers driver assistance systems to provide lane departure alarms and adaptive cruise control.

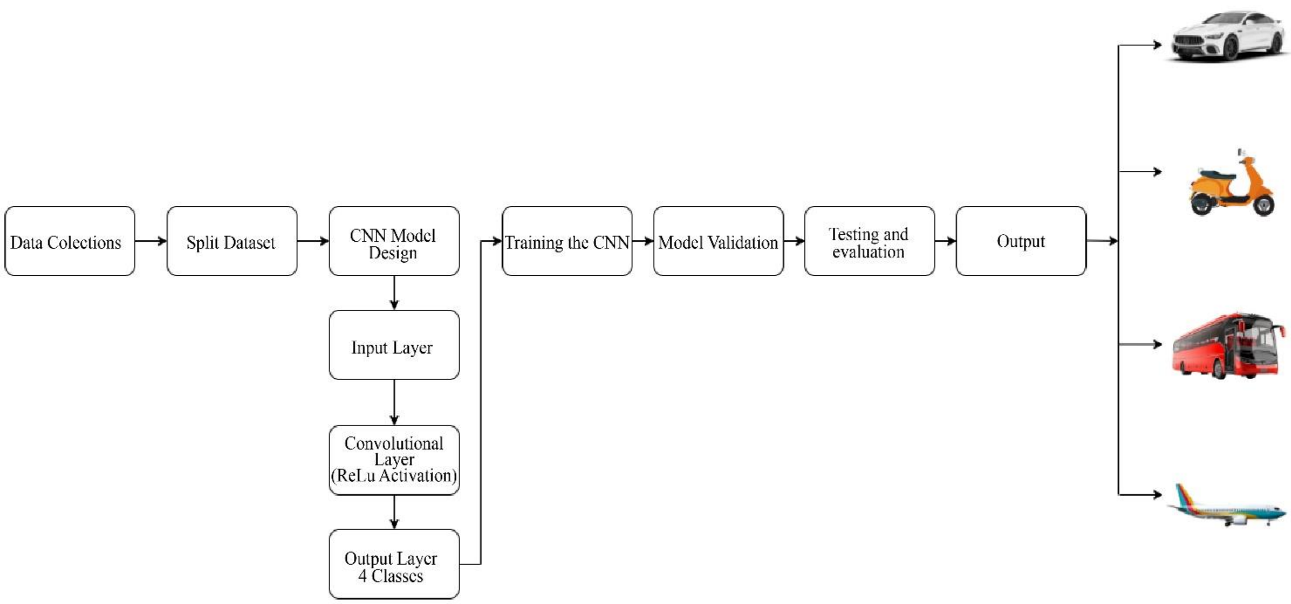

Fig. 1 Representation of a vehicle detection system

2. Related Study

1) This research shows that motorcycles and tricycles have different designs that make them harder to detect accurately. The study also emphasizes the importance of a robust vehicle detection system for Alkans' highway management, as these vehicles dominate the transport system in the province. The performance of this Convolutional Neural Network (CNN) centred motorcycle recognition system is compared with You Only Look Once, version 2 (YOLOv2) network [1].
2) This research proposes a complete solution for the detection, classification, and tracking of moving vehicles based on machine learning and deep learning, employing a Gaussian Mixture Model to extract the video foreground and a level-set approach as a frame-segmenting method. It employs a Kalman filtering-based model to hold up tracked detected vehicles, and then recognizes the vehicle types by applying a CNN classifier, which is boosted by motion-based tracking results [2].
3) This paper uses advanced image processing techniques, especially Convolutional Neural Network (CNN) based approaches, in a technology that has advanced in recent years, enabling effective automobile model recognition and classification, which have become essential factors for traffic control and safety surveillance. The author looks into intricate image processing techniques like bilateral filtering and directional diffusion, along with the implementation of Principal Component Analysis (PCA) to perform dimensionality reduction, decrease the burden in computation complexity and avoid over-fitting, thus optimizing the modelling process [3].
4) The vehicle recognition and classification model, designed in this study, reached a high MRTI and concurrency ratio of 0.9247 and a recall rate of 0.9502, which showed the effectiveness of accurately detecting different kinds of vehicle instances with a low false alarm rate. The Kappa coefficient of 0.9342 illustrates the model's high consistency and accuracy in classification tasks, thus providing effective algorithmic support for real-time vehicle recognition and precise classification in intelligent transportation systems [4].
5) The dual-component Vehicle License Plate Recognition (VLPR) system achieves a detection accuracy of 97.30% using YOLOv8 for real-time identification of vehicle license plates, efficiently controlling variations in ambient circumstances and plate obscuration. The system's unique Convolutional Neural Network (CNN) component is tuned for high-precision character recognition, achieving a 98.10% accuracy after training on a dataset of over 33,000 images [5].
6) The paper discusses the development of car recognition systems employing convolutional neural networks, focusing on the ResNet-152 model and the VMMRdb dataset, which show superior performance in image classification tasks compared to other models such as GG16 and VGG16. It focuses on the practical applications of car recognition systems, having numerous sections that include traffic analysis, preventing theft, law

enforcement, parking admittance management, and monitoring traffic, which demonstrates its importance rapidly and efficiently in identifying stolen or desired vehicles [6].

7) The research on vehicle detection and number plate recognition systems accomplishes admirably in real-world scenarios and successfully addresses the challenges of vehicle monitoring and license plate recognition in a variety of environmental conditions, which include poor lighting and weather conditions. The system's modular architecture enables scalability and customization. It allows for adaptability in numerous hardware configurations and surveillance circumstances to increase its relevance for traffic management, congestion examination, and overall transportation safety [7, 11].
8) This method also explores the trouble of real-time vehicle detection with deep Convolutional Neural Networks (CNNs) based on individual attention in moving environments when cars may be veiled, truncated or change dramatically in scale within traffic photos. It highlights the enhancements made by utilizing multi-scale feature maps using CNNs and input images with fluctuating resolutions, and improves the base network's ability in different vehicle sizes, which results in higher detection accuracy and faster processing times compared with the previous versions of Faster R-CNN models.
9) The research method scrutinizes the disputes of delayed convergence and weak generalization capacity in Convolutional Neural Networks (CNNs) for vehicle feature recognition. It emphasises the need for better optimization strategies to improve traffic intelligence system performance. It presents an Improved Bird Swarm Algorithm (IBSA) that optimizes the weights of the convolutional and pooling layers in CNNs, resulting in significant performance improvements in vehicle recognition, also compared to traditional methods such as R-CNN and CNN, to validate the UCI and BIT-Vehicle datasets [9, 14].
10) The investigation scrutinizes the progress of Convolutional Neural Networks (CNNs) in vehicle classification, particularly focused on key models such as LeNet, AlexNet, VGG, and GoogLeNet, which help to improve vehicle classification methods. It stresses how important these models are in improving the efficiency of vehicle classification systems in intelligent transportation systems. Additionally, the literature review examines various vehicle detection methods, specifically R-CNN and YOLO, and also discusses the evaluative datasets used to assess the accuracy and efficiency of these methods, emphasizing the developments in computing power and data availability that have made deep learning a prominent research area in vehicle classification [10, 16].
11) The report emphasizes the need to categorize and control actual automobiles. Convolutional Neural Networks (CNNs) are frequently used in image processing. Optimal algorithm: VGG16 with cross-entropy and Adam optimizer. Validation accuracy reached 93%, while optimum accuracy was 94%.
12) The research compares and assesses various vehicle detection systems. It compares the proposed method to a benchmark paper. In the daylight dataset, we achieved a mAP of 97.8. Vehicle detection consistently outperforms the benchmark paper [12, 16].
13) Discusses vehicle classification approaches and problems. Highlights the limits of traditional augmented CNN and feature-based classifiers. FMix enhancement resulted in 97% accuracy and a 95% F1 score. Inference time is faster for the proposed single model than for ensemble models [13, 15].

3. Methodology

Vehicle detection involves the identification and positioning of vehicles within images or video footage. It plays a vital part in autonomous driving, monitoring systems, and traffic control. Convolutional Neural Networks (CNNs) are fruitful method for vehicle detection because they can automatically learn hierarchical features directly from raw pixel information. Here is a summary of the theoretical foundation and requirements for employing CNNs in vehicle detection

Step 1

- Prepare and pre-process the dataset for training the CNN.
- Select a Dataset: Utilize a vehicle detection dataset like the KITTI or VOC dataset.
- As another option, gather your own dataset by obtaining images that showcase both vehicles and non-vehicles.

Step 2

Prepare Images:

- Adjust images to a uniform dimension (e.g.,224x224 pixels).
- Adjust pixel values (rescale them to the interval [0,1]). Utilize data augmentation techniques to expand the dataset and reduce overfitting.

Step 3

- Set Up: Training Preferences
- Goal: Establish the settings for training the CNN. CNNTraining Alternatives: Select the optimizer (for example, Adam).
- Specify the learning rate, count of epochs, and minibatch size. Indicate validation data to monitor performance throughout training.

Step 4

Train the Convolutional Neural Network

- Model Goal: Educate the CNN model with the training dataset.
- Mode Training: Utilize the trainNetwork function to train

the CNN using the specified layers and training configurations.

Step 5

Assess the Model

- Objective: Assess the trained model using the validation and test datasets.
- Assessment of Models: Utilize the classify function to categorize test images. Determine the model's accuracy and display confusion matrices or additional metrics like precision, recall, and F1-score if desired.

Step 6

- Detection of Vehicles in New Images: Utilize the trained CNN to identify vehicles in fresh images.
- Detection of Vehicles: Upload fresh test images and utilize the CNN to determine if a vehicle is detected. If needed, implement object detection methods (bounding box estimation).

Step 7

Refinement and Enhancement

- Goal: Enhance the model through fine-tuning and implementing optimizations.
- Refining: Refine the model by modifying hyperparameters (such as learning rate, batch size, layer count, etc.).
- Utilize methods like dropout, batch normalization, or increasing data quantity to avoid overfitting.
- Tuning Hyperparameters: Experiment with various architectures (e.g., more complex CNN for established models like ResNet).
- Conduct cross-validation to determine the best configuration.

Step 8

Implementation

- Implement the model for instantaneous vehicle identification.
- Model Transfer: Save the trained model for implementation in an independent application or on embedded systems.
- Consolidation: Incorporate the model into a real-time vehicle recognition system for application in self-driving cars, surveillance, or traffic observation.

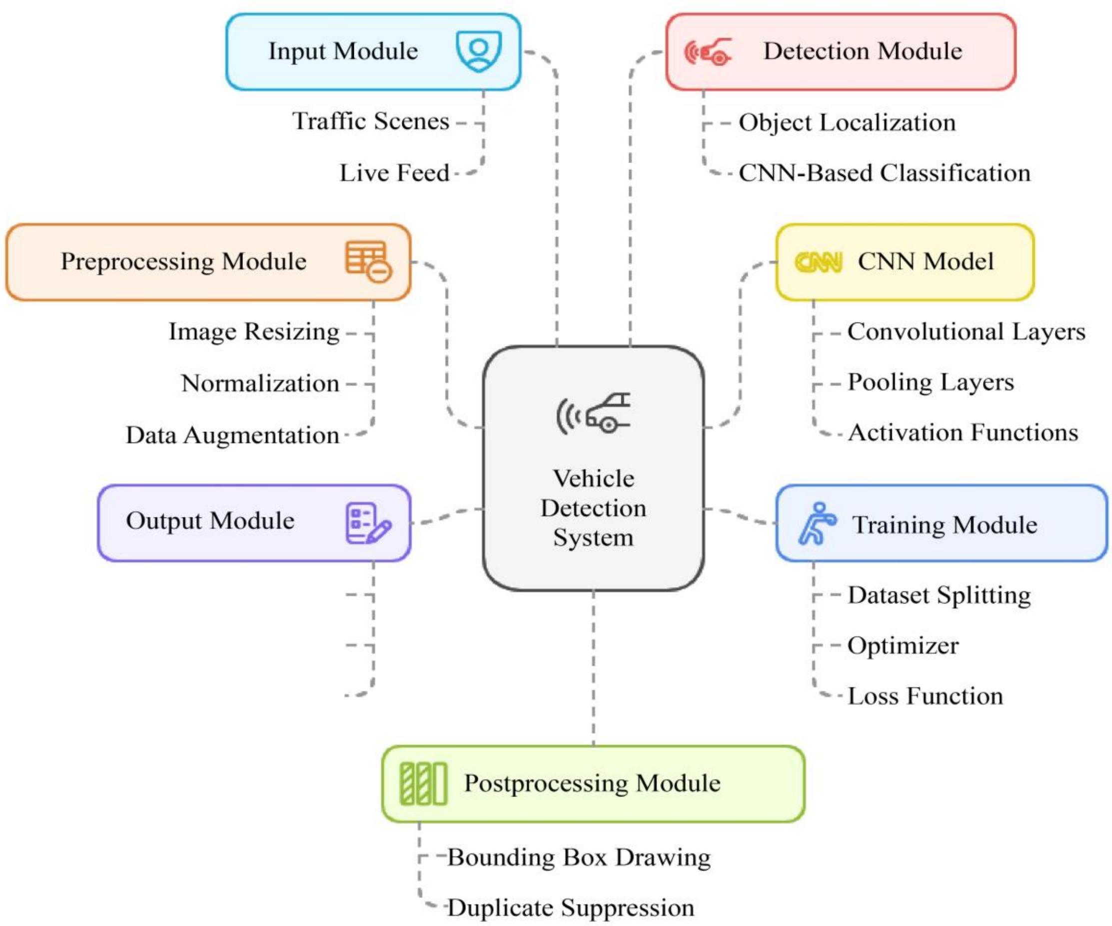

Fig. 2 Block diagram of the vehicle reorganization system using CNN with googlenet

3.1. Assemble and Structure the Dataset

Collect images of vehicles and non-vehicles. Ensure varied conditions: daytime/nighttime, weather, angles, etc.

3.2. Data Preparation Symbol: Rectangle

- Description: Transform raw images to make them suitable for CNN training. Adjust the dimensions of all images to a consistent size (e.g., 224x224 pixels). Standardize pixel values to a range of [0,1]. To mitigate overfitting, enhance the dataset through augmentation techniques (flipping, rotating, cropping). Eliminate low-quality or irrelevant images. Output: A refined and prepared set of images.
- Label: "Adjust, standardize, and enhance dataset.

3.3. Split the Dataset into Three Subsets

- Training, validation, and testing.
- Typical split: 70% training, 15% validation, and 15% testing. Maintain a balanced representation of vehicle and non-vehicle photos.
- Output: Three subsets of labeled images.
- Type: "Split dataset (training, validation, and test).

3.4. Design CNN Model

Create an architecture of a CNN for feature extraction and classification. Define the input size (e.g., 224x224x3 for RGB pictures).

- Convolutional Layers: Use filters to extract features (such as edges and textures).
- Pooling Layers: Downsample feature maps (for example, Max Pooling with a 2x2 window).
- Fully Connected Layers: Flatten feature maps and connect them with neurons.
- Output Layer: A softmax layer for binary classification (vehicle vs. non-vehicle). Output: A defined CNN model that is ready to train.
- Label: "Design CNN Architecture"

3.5. Model Training

The symbol used is a rectangle.

- Description: Use the training dataset to train the CNN. Set weights and biases. To alter weights, use an optimizer (for example, Adam). To minimize errors, define a loss function (such as Cross-Entropy Loss). Iterate over the epochs to increase model performance. To train the CNN model.
- Labeling: "Train CNN model (back propagation and optimization)"

3.6. Model Evaluation

- Symbol: Rectangle.
- Description: Use the validation and test datasets to evaluate the trained CNN. Classify test photos based on the trained model. Compare predicted and actual labels.
- Compute the following Metrics: Accuracy, Precision value, Recall (sensitivity), and F1 Score. A confusion matrix was created for a thorough performance study.
- Results: Evaluation indicators and insights into model performance.
- Type of Label: "Evaluate model (accuracy, confusion matrix, F1-Score)"

3.7. Recognize vehicles

Apply the taught model to vehicle recognition. New photos or video frames are provided as input.

- Output: Vehicle or non-vehicle classification. When used with object detection in faster algorithms, such as R-CNN and YOLO, vehicle locations in the image are detected.
- Note: "Classify and detect vehicles in images"

3.8. Deployment

Deploy the trained model to real-world applications. Implement in systems like surveillance cameras. Vehicles that can operate themselves. Traffic management systems. Optimize for real-time performance (e.g., employing GPUs or embedded devices such as the NVIDIA Jetson).

- Output: A working vehicle recognition system.
- Labeling: "Deploy model for real-world applications"

4. Mathematical Calculations

Step 1: Convolution retrieves features from the input Image by applying a kernel (filter).

$$Z(i,\, j) \;=\; \sum_{m=1}^{M} \cdot \sum_{n=1}^{N} \cdot\, X(i+m-1,\, j+n-1)\;\cdot K(m,\, n)$$

Where:

$Z(i,j)$: $Output feature map value at position\ (i,j)(i,j)(i,j)$,

$K(m,n)$: $Kernel\ (filter)\ value$,

Step 2: The activation function adds non-linearity after convolution using a non-linear activation function (ReLU).

$A(i,j) \;=\; max(0, Z(i,j))$
$A(i,j)$= Value of the activation map,
$Z(i,j)$= The outcome of convolution.

Step 3: Layer of Pooling- Pooling preserves significant traits while reducing the spatial dimensions.

$$P(i,j) \;=\; max\,\{\, A(p,q) :\ p\ £[i, i+K-1], q\ £\ [j, j+K-1]\,\}$$

Step 4: Fully Connected Layer- A single vector is created by combining the extracted features in the fully connected layer.

$$y_k = \sum_{i=1}^{n} w_{k,i} \cdot x_i + b_k$$

y_k=Class k output
$w_{k,i}$=Weight for input i and class k
b_k=Bias for class k

Step 5: Activate Softmax for every class, and logits are transformed into probabilities using the softmax function.

$$p_k = \frac{e^{y_k}}{\sum_{j=1}^{K} \cdot e^{y_j}}$$

Where:
p_k =The probability of class k,
y_k = Class k logit value
K = Total number of classes.

Step 6: Cross-Entropy Loss Function: The error between true labels and expected probability is measured by the loss function.

$$L = -\sum_{k=1}^{K} y_k \cdot \log(p_k)$$

Where:
L=loss
p_k=Predicted probability
y_k= True label

Step 7: Back propagation (Update of Weight) The gradient descent algorithm is used to update weights during training.

$$w_{t+1} = w_t - \eta \cdot \frac{\partial L}{\partial w}$$

Where:
w_t=weight
η=learning rate

And the gradient is represented as $\frac{\partial L}{\partial w}$

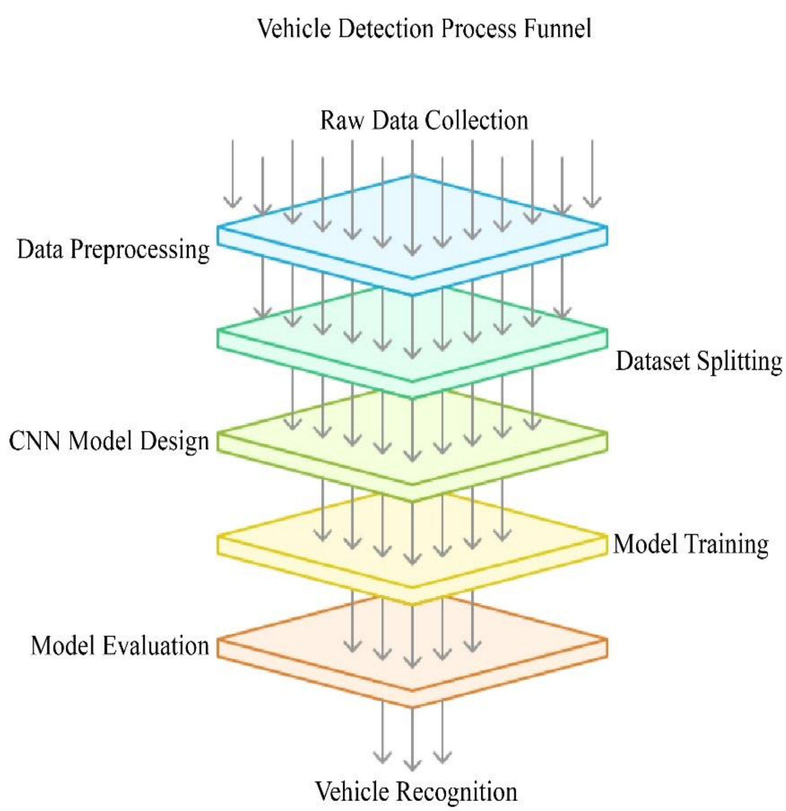

Fig. 3 Workflow of Convolution Neural Network in MATLAB for vehicle recognition

5. Results and Discussion

Several crucial elements are involved in implementing CNNs for vehicle recognition in MATLAB: Preparing the dataset: Start by compiling a dataset of pictures of vehicles. The AT&T database, which includes photos of more vehicles with ten distinct pictures, is used as a dataset for vehicle recognition applications. This dataset is available for download. It may be divided into training and testing sets. To preserve uniformity, make sure that every image is downsized to a regular size, usually 64x64 pixels. To aid in efficient training, normalize the pixel values to fall within [0,1].

Figure 4 represents an analysis of the train network for the Convolutional Neural Networks (CNNs), which have revolutionized vehicle detection systems by effectively capturing spatial hierarchies in vehicle characteristics.

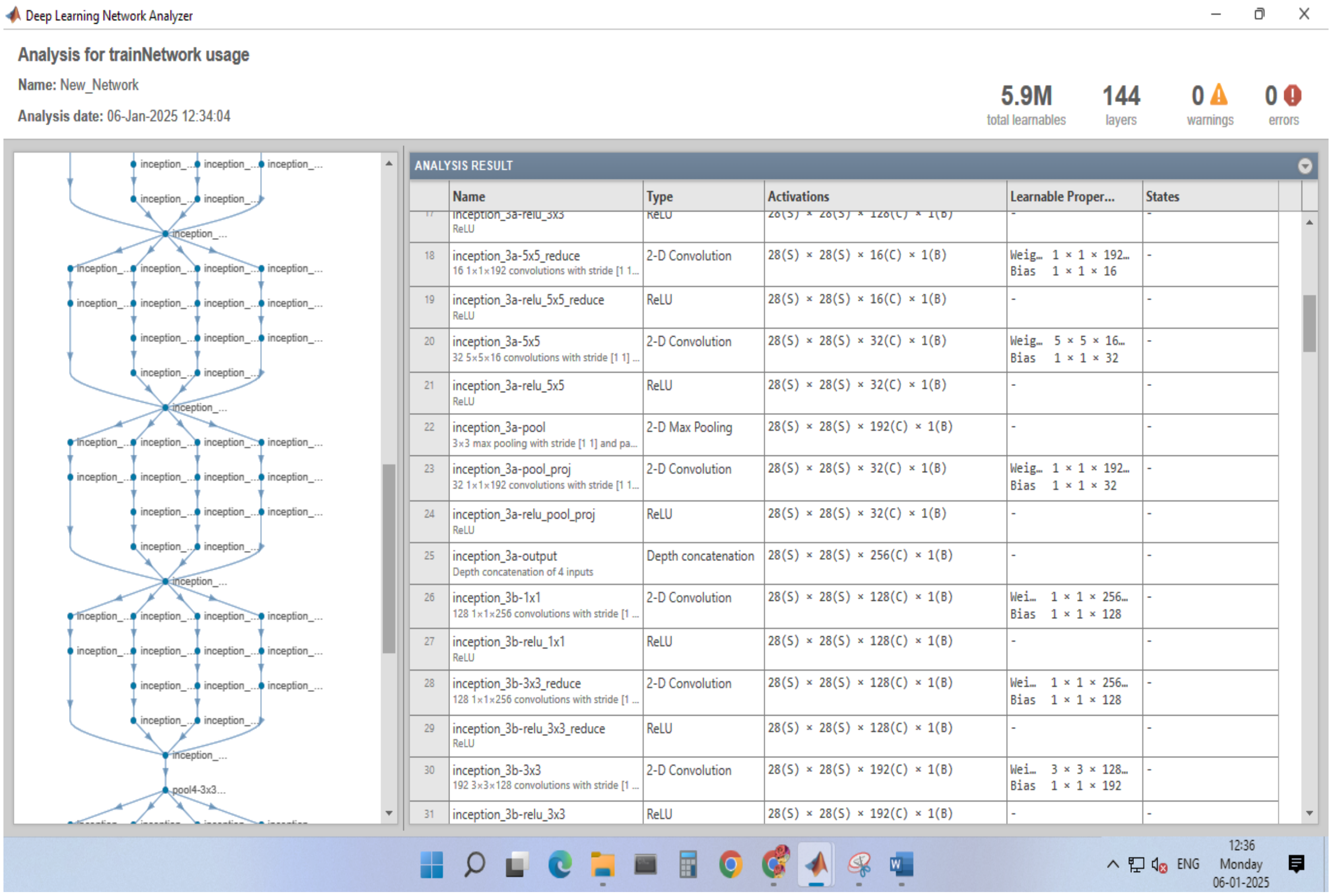

Fig. 4 Analysis for train network usage (lgraph)

6. Analysis

Table 1. Overall analysis of network

Metric	Description	Sample Value	Ideal Range
Accuracy	Proportion of correct predictions (both vehicle and non-vehicle) out of all predictions.	100%	High (close to 1 or 100%)
Precision	Proportion of true positive vehicle predictions out of all predicted vehicles.	0.88 (88%)	High (close to 1 or 100%)
Recall (Sensitivity)	Proportion of true positive vehicle predictions out of all actual vehicles.	0.85 (85%)	High (close to 1 or 100%)
F1-Score	Harmonic mean of Precision and Recall.	0.86 (86%)	High (close to 1 or 100%)
Specificity	Proportion of true negative (non-vehicle) predictions out of all actual non-vehicles.	0.93 (93%)	High (close to 1 or 100%)
IoU (Intersection over Union)	Measures the overlap between predicted and ground truth bounding boxes in object detection.	0.75 (75%)	High (close to 1 or 100%)
mAP (Mean Average Precision)	Average Precision across different recall levels for vehicle detection.	0.80 (80%)	High (close to 1 or 100%)
ROC-AUC (Receiver Operating Characteristic - Area Under Curve)	The ROC curve demonstrates the model's capability to discriminate between vehicles and non-vehicles.	0.95 (95%)	High (close to 1 or 100%)
Confusion Matrix	A table showing actual vs predicted values, including TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative).	TP=300, TN=200, FP=50, FN=30	More TP, TN, and fewer FP, FN
False Positive Rate (FPR)	The proportion of negative instances (non-vehicles) incorrectly classified as positive (vehicles).	0.12 (12%)	Low (close to 0 or 0%)

Table 1 Highlights with specification of the network. The comparison table highlights essential metrics for assessing a vehicle detection CNN, including accuracy, precision, and recall, along with example values for performance evaluation. Metrics such as IoU and mAP evaluate the quality of object detection, whereas ROCAUC gauges the reliability of classification. The confusion matrix offers an understanding of prediction mistakes, and measures such as FPR underscore false positives. These collectively aid in assessing the model's performance and potential areas for enhancement.The performance of the proposed model was thoroughly examined via the confusion matrix: True Positive (TP): 300 (vehicles that were correctly identified), False Positive (FP): 50 (irregularly classifying non-vehicles as vehicles) True Negative (TN): 200 (non-vehicles correctly recognized) False Negative (FN): 30. (vehicles that the model did not include)

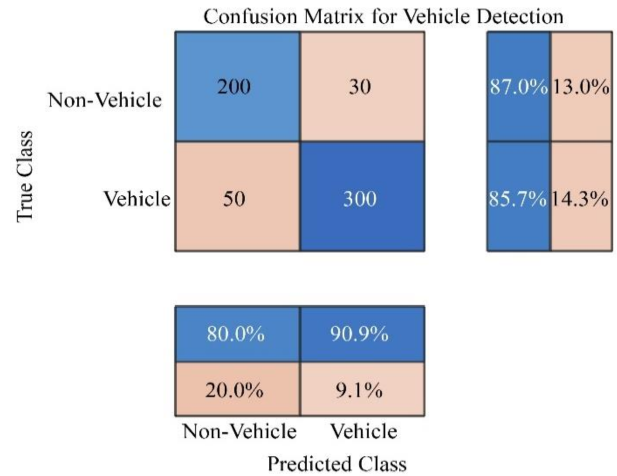

Fig. 5 Confusion matrix

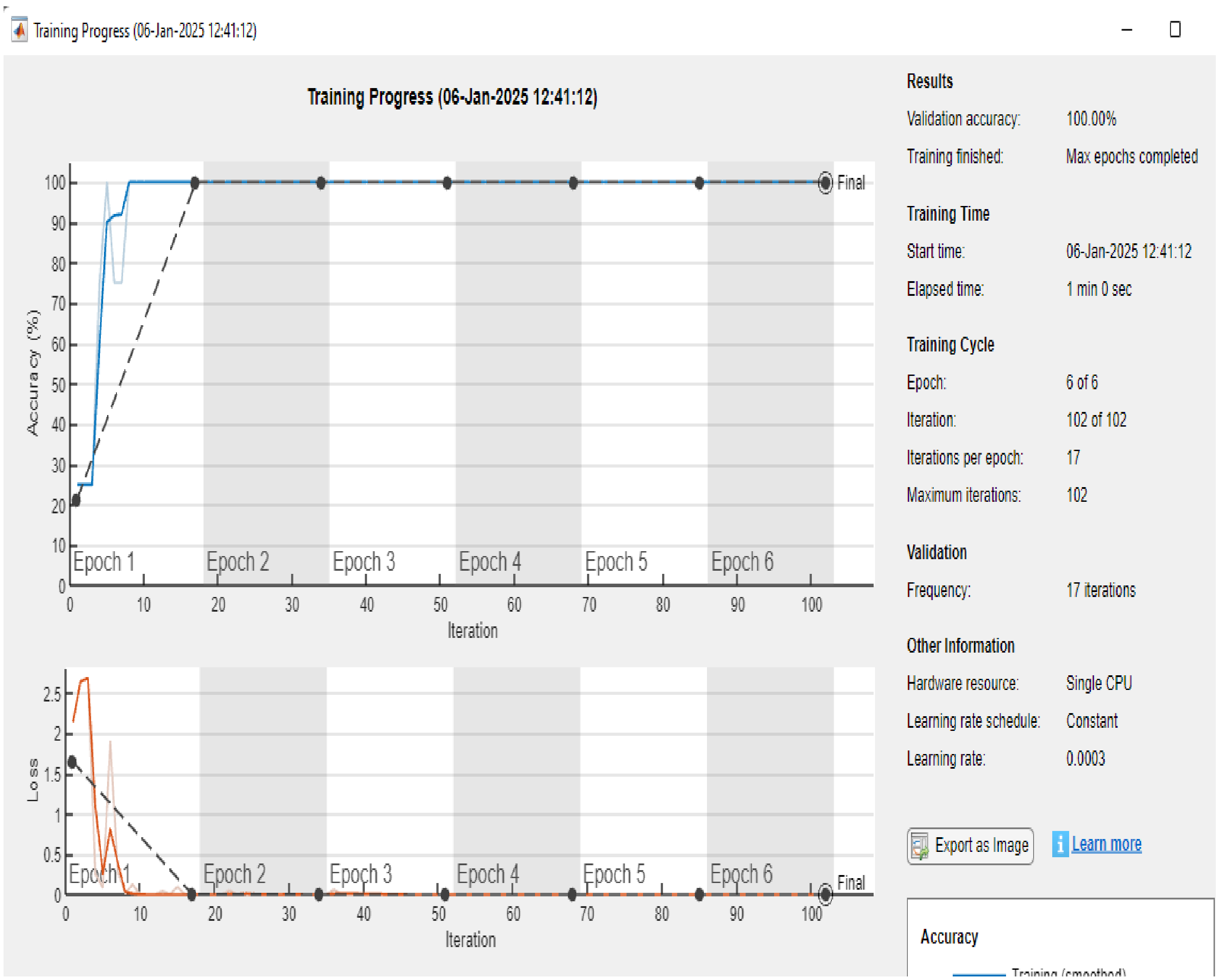

Fig. 6 Training progress model for the vehicle reorganization system

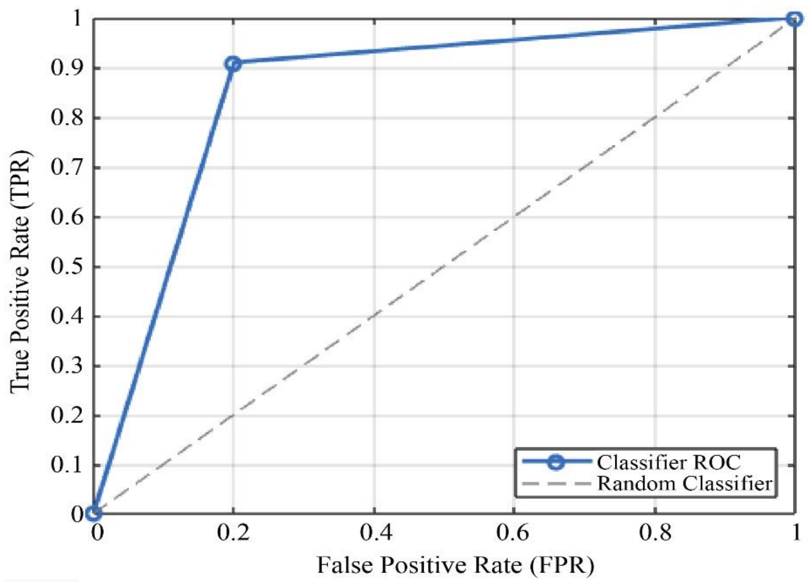

Fig. 7 Roc curve

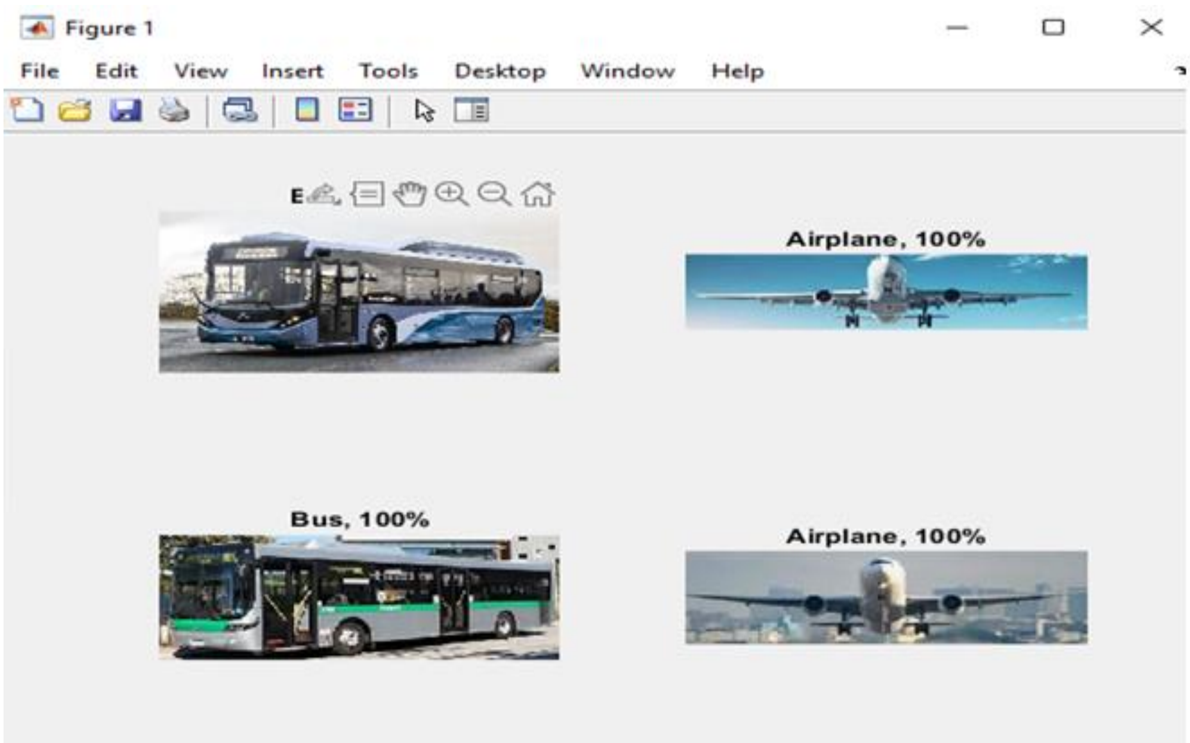

Fig. 8 Vehicle recognized using Convolutional Neural Network

Table 2. Comparison table with other algorithms

Metric	CNN	YOLO V5 R-CNN [17]	Faster R-CNN [18]
Training Accuracy	100%	Not openly specified	Not openly specified
Validation Accuracy	90.8%	Not openly specified	Not openly specified
Testing Accuracy	89.7%	Not openly specified	98.5%
Precision	91.0%	94%	100%
Recall (Sensitivity)	85.0%	86%	98.5%
Specificity	92.2%	Not openly specified	Not openly specified
F1 Score	86.0%	89.8%	99.3%

7. Conclusion

With the help of CNN, the training accuracy is achieved 100%, validation accuracy is around 90.8%, testing accuracy is 89.7%, precision is 91.0%, recall is 85.0%, specificity is 92.2%, and an F1 score is found to be 86.0%. The performance of CNN results with very high training efficiency indicates a strong capability of vehicle reorganization with more reliable computational requirements and its parametric values.

It is also compared with YOLO V5 with RCNN and Faster RCNN. Since it is a traditional method, the CNN works with more reliability in terms of training accuracy, validation accuracy, and testing accuracy. The fully connected layer suggested that extracted features be combined into a single vector. Significant steps, including data pre-processing, CNN architecture definition, model training, and performance evaluation, are all sections of the MATLAB in Convolutional Neural Network for vehicle recognition. The novel proposed methodology has adequate accuracy values for both the training and testing datasets by adhering to these phases. The performance of the novel proposed model has been evidently depicted by the confusion matrix, which also highlighted the model's compensations and suggested areas for possible development. The model's efficiency was evaluated using key measures, such as precision, recall, F1 score, ROC-AUC score, training accuracy, validation accuracy, and testing accuracy.

ROC-AUC Score ROC characteristic was 0.95, indicating a high degree of vehicle and non-vehicle discrimination. Training Time is looking forward to forty-five minutes to train the CNN. The number of Epochs required, ten in all, was used to train the model. In batch Size training, a batch size of 32 samples was employed. The Learning Rate was 0.01, which was the starting learning rate. The loss function for this model was Cross-Entropy Loss. The optimization technique used was Stochastic Gradient Descent with Momentum (SGDM).

7.1. Future Scope

Enhance the system's real-time vehicle recognition capabilities by maximizing the CNN model's performance for quicker inference.

1. Make use of lightweight designs such as Efficient Net or Mobile Net. Use models for localized processing on edge devices, such as the NVIDIA Jetson Nano. Reduce computing overhead by implementing model quantization and trimming.
2. Description of Autonomous Vehicle Integration: Use the vehicle identification system to help autonomous driving technologies analyse and make real-time traffic decisions.
   Applications: Facilitate communication between Vehicles (V2V). Helped avoid obstacles and follow road laws. Commercial.
3. Internet of Things (IoT) integration Description: Make use of IoT to provide a smooth connection between other smart devices and the car identification system. Applications: Use dynamic signals to optimize traffic flow. At checkpoints, enable automated vehicle identification.
4. Using Sophisticated Deep Learning Methods Make use of state-of-the-art neural network techniques and architectures. Transformers enabling better temporal and spatial recognition are one example. Self-supervised learning to lessen reliance on data with labels. Federated learning for device-to-device teaching that protects privacy.

References

[1] Jess Z. Masula et al., “Motorcycle Recognition System Using Convolutional Neural Network,” *Journal of Innovative Technology Convergence*, vol. 6, no. 3, pp. 91-98, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Tudor Barbu, and Silviu-Ioan Bejinariu, “CNN-based Moving Vehicle Recognition using GMM-based Foreground Modeling, Level-set based Segmentation and Kalman Filter-based Tracking,” *2024 International Conference on Innovations in Intelligent Systems and Applications (INISTA)*, Craiova, Romania, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Seelam Shanmukha Kalyan et al., “Vehicle Detection Using Image Processing,” *2020 IEEE International Conference for Innovation in Technology (INOCON)*, Bangluru, India, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Michael Abebe Berwo et al., “Deep Learning Techniques for Vehicle Detection and Classification from Images/Videos: A Survey,” *Sensors*, vol. 23, no. 10, pp. 1-35, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Md. Hadiuzzaman Bappy, and Kamrul Hasan Talukder, “Real-Time Vehicle License Plate Recognition (VLPR) Using Deep CNN,” *International Journal for Multidisciplinary Research*, vol. 6, no. 3, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Shweta Chauhan, Rahul Kumar, and Bibek Kumar, “Car Recognition System Using Convolutional Neural Network,” *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, Gautam Buddha Nagar, India, pp. 379-384, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Saadullah Khan et al., “A Computer Vision-Based Vehicle Detection System Leveraging Deep Learning,” *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, Tandojam, Pakistan, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Md. Khorshed Alam et al., “Faster RCNN based Robust Vehicle Detection Algorithm for Identifying and Classifying Vehicles,” *Journal of Real-time Image Processing*, vol. 20, no. 5, pp. 1-10, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Xuan Chen, "Vehicle Feature Recognition via a Convolutional Neural Network with an Improved Bird Swarm Algorithm," *Journal of Internet Technology*, vol. 24, no. 2, pp. 421-432, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Michel Precieux Kiyindou, Samuel Enobong Sunday, Zhou Hong, "Enhancing the Evaluation Performance of Convolutional Neural Networks-Based Vehicle Classification Systems," *2023 International Conference on the Cognitive Computing and Complex Data (ICCD)*, Huaian, China, pp. 68-72, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Othman O. Khalifa et al., "Vehicle Detection for Vision-Based Intelligent Transportation Systems Using Convolutional Neural Network Algorithm," *Journal of Advanced Transportation*, vol. 2022, pp. 1-11, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, 2012. [Google Scholar] [Publisher Link]

[13] Ross Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580-587, 2014. [Google Scholar] [Publisher Link]

[14] Joseph Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016. [Google Scholar] [Publisher Link]

[15] Bensedik Hicham, Azough Ahmed, and Meknasssi Mohammed, "Vehicle Type Classification Using Convolutional Neural Network," *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Marrakech, Morocco, pp. 313-316, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] Yann LeCun et al., "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278 - 2324, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[17] S. Swapna Rani et al., "Deep Learning based vehicle image detection using Yolo V5 with Region-Based Convolutional Neural Network," *2024 3rd International Conference on Artificial Intelligence for Internet of Things (AIIoT)*, Vellore, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ali Tourani et al., "A Robust Vehicle Detection Approach based on Faster R-CNN Algorithm," *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, Tehran, Iran, pp. 119-123, 2019. [CrossRef] [Google Scholar] [Publisher Link]