Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

Volume 07 Issue 02
2024

Research Article
https://doi.org/10.56979/702/2024

Real-Time Vehicle Detection with Advanced Machine Learning Algorithms

Syeda Saliha Kazmi[1], Shahbaz Shamshad[1], Fawad Nasim[1], Saira Hashim[1], Nisha Azam[1],* and Bushra Ambar[1]

[1]Faculty of Computer Science and Information Technology, Superior University, Lahore, 54000, Pakistan.
*Corresponding Author: Saliha Kazmi. Email: salihakazmi21@gmail.com

Received: March 21, 2024 **Accepted:** August 26, 2024 **Published:** September 01, 2024

Abstract: Vehicle detection not only increases road safety but also helps regulate traffic flow and improves the functioning of intelligent transportation systems. The model chosen was highly trained and tuned for an accuracy of 99%. It is far improved, considering that the detection accuracies in the literature were usually between 85% and 88%. It proposes a real-time vehicle detection system using a Convolutional Neural Network (CNN) model. The dataset "Vehicle Detection" consisted of 14,208 training images, 1,776 validation images, and 1,776 test images of size 1024x1024 pixels each and annotated with 20,000 bounding boxes. It is a 23-layer deep Convolutional neural networks model with 14.7 million parameters using ReLU and softmax as the activation functions, trained for 40 epochs using the Adamax optimiser and categorical cross-entropy loss. It used a custom callback for the hyperparameter tuning process, arriving at an initial learning rate of 0.001 and a batch size of 40. The model performed very well, with an accuracy of 99.4% on training, 99.3% on validation, and 99.4% on testing, with precision and recall of 99.3% and 99.4%, respectively.

Keywords: Autonomous Vehicle; Convolutional Neural Networks; Deep Learning; Vehicle Detection; Machine Learning.

1. Introduction

Vehicle detection involves identifying and localizing vehicles in any image or video frame with sensors and computer algorithms. It provides the basis for traffic monitoring, autonomous driving, and security systems [1]. Vehicle detection not only increases road safety but also helps regulate traffic flow and improves the functioning of intelligent transportation systems. The history of vehicle detection dates back to the middle of the 20th century when inductive loop sensors were first used to detect vehicles at intersections. Several decades later, technological improvements resulted in more advanced methods, including Convolutional neural networks and computer vision techniques [1]. Deep learning and artificial intelligence have created a real-time, high-accuracy identification of vehicles in complicated surroundings, compared to essential vehicle detection, within the past few years. These are innovations that have remained very relevant in modern transport systems. Real-time vehicle detection has become necessary in many applications, including autonomous driving, traffic management, surveillance systems, etc. The ever-increasing demand for intelligent transportation systems to work in dynamic environments efficiently and safely has pushed the development of deep learning techniques [2]. Indeed, it revolutionized the concept of vehicle detection, classification, and tracking for closer and quicker decision-making. The development of vehicle detection in the last few years has shifted from traditional computer vision methods to advanced deep-learning models. It has proved very effective in image recognition tasks, automatically learning and figuring out the features at different levels from raw input data. Its extraordinary faculty is instrumental in vehicle detection, where the significant variance of shapes, sizes, and orientations poses a considerable challenge. While bursting auspicious achievement in vehicle detection, deep learning contains numerous critical issues that need to be solved for the practical application of current deep learning-based vehicle detectors [4]. A highly critical issue for real-time vehicle detection systems is the trade-off between speed and accuracy. The current state-of-the-art models, such as faster Convolutional Neural Networks, achieve very high accuracies but lose them in scenarios with very fast-moving vehicles or a complex background and changeable environments [7]. Most of these models require a vast number of resources for computation, thereby being very hard to deploy on edge devices or in environments where real-time processing with low latencies is needed. The other pressing issues are the robustness in handling occlusions, partial visibility of vehicles, and varying weather conditions [3]. For instance, vehicles can be partially or fully blocked in heavy urban representations or under poor weather, and several cases of detection failures or lack of accuracy can be visually observed. Detection of vehicles is further complicated by the variability in vehicle appearance, which can be in different colors, shapes, and sizes-more so in situations where fast identification is critical [11]. Another huge challenge is the non-scalability and non-adaptability of current models. Deep learning model can be trained on datasets that might not have captured the entire variability present in real-world environments, impacting their detail over scenes they have not seen, an issue that creates a safety risk in implementation, ranging from autonomous driving to intelligent traffic management [4]. This approach tremendously reduces the need for abundant train data and computational resources to retrain the models in different environments and for tasks newly defined. The application of attention mechanisms and transformer networks holds promise in improving features that focus on relevance and filtering out irrelevant background noise [21].

This model better handles complex scenes with multiple vehicles and varying backgrounds by autonomously concentrating on different parts of the input data. They are recognized for extracting dependency over time and enhancing the better capture of cars in real time. Consider a simplified mathematical framework to interpret the effect of these advanced models [5]. Let x_i represent input data, e.g., image and distance measurements from different vehicle sensors I, and let y_i reduce the corresponding decision or action, such as the braking, steering angle, etc., which the Autonomous vehicles have to do. This decision-making process can be modelled with a function f (θ, x_i) = y_i, where θ denotes the model factor for machine learning.

$$\mathrm{E}\,(\theta) = \sum_{i=1}^{N} || \, y_i - f(\theta,\ x_i) \, ||^2 \tag{1}$$

It differs from the traditional model's objective to minimize the error over all training examples. However, due to its variability and complexity, this only sometimes holds in a real-world driving environment.

2. Literature Review

Recently, convolutional neural networks have been applied significantly in vehicle detection systems since they can capture spatial hierarchies and features from image data [6]. Many more studies have been published from 2022 to 2024, showing variable success rates, especially in real-time applications, where computational efficiency and accuracy are critical. This literature has mainly focused on assessing Convolutional Neural Networks' performance detecting vehicles within real-time environments [7]. It picks out research whose accuracy ranges from 85% to 87%, trying to point out difficulties or challenges contributing to such a threshold in performance. This scope is chosen to understand what factors incorporate suboptimal performance and how the contributions address these challenges, particularly in enhancing accuracy through model optimization [5]. One research dated back to 2022 reached an accuracy of 87%. However, it identified vehicles poorly in bad light conditions, such as at night or in shadowy areas. This exposed a possible limitation: the need for more efficient pre-processing techniques or other model architectures to function under these conditions [6]. One of the relevant studies, conducted in 2023, realized an accuracy of 85.5% using a self-designed Convolutional Neural Networks architecture, which suffered from overfitting due to too-small dataset size and a need for more data augmentation techniques. This study implies that improving the data's diversity may enhance generalization to an increased cost of training time and computational resources [7]. Another effort in the research reported an accuracy of 86.3% with a deeper Convolutional Neural Networks model as early as January 2024. However, all of this came at the cost of longer inference times with increased model complexity, a significant drawback in real-time applications [7]. Finally, sustained research into data augmentation techniques and synthetic data generation would result in more varied and representative training datasets. Approaches for achieving such precision are further detailed to measure appreciation for improved robustness of deep learning models under various simulation environments, vehicle appearances, and occlusions towards better performances in real-time scenes [8]. Much effort has gone into enabling real-time vehicle detection with deep learning. Still, many daunting challenges must be faced in other aspects, including high accuracy, speed, robustness, and scalability. Solutions reviewed here, such as generating lightweight models, incorporating multi-modal data, and using advanced network architectures, are promising areas for future research [9]. By further innovating and refining these developments, it can be counted on that real-time systems based on vehicle detection will prepare situations of high reliability and efficiency in complex, dynamic, real-world scenarios. Incidentally, there are criticisms in real-time vehicle detection for occlusion handling, high illumination variance, and the need to deal with models to function well in on-edge devices, considering profound resource implications in computation [10]. These open challenges need further innovative research in marrying appropriate algorithm design to data augmentation techniques and developing lightweight models while keeping the performance high at little resource consumption [12]. Therefore, one way to address the outlined challenges in the above project is to reduce the computational degree by developing lightweight deep learning models with high accuracy but low computation count [11].

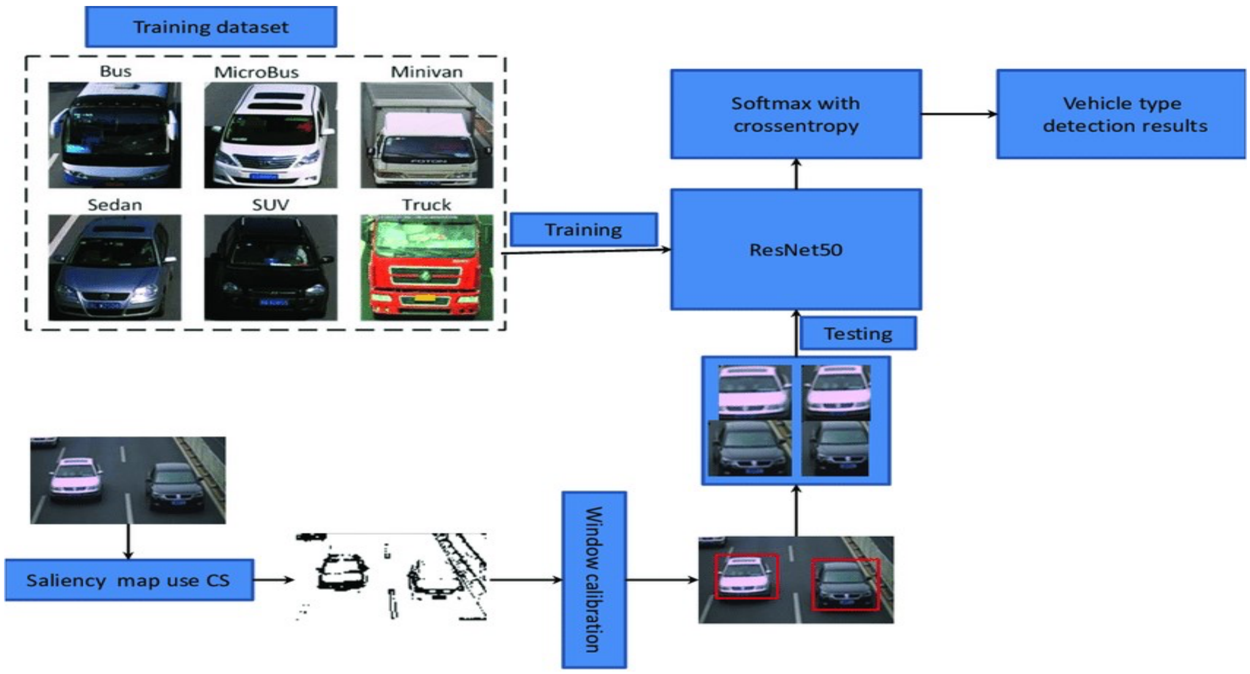

Figure 1. Showing comparison in vehicles.

In this line, methods including but not limited to model pruning, quantization, and knowledge distillation could be leveraged for crafting efficient models deployable on resource-constrained edge devices. This paper concludes that an appropriate trade-off between model depth and computing efficiency is required for the practical applications of Convolutional Neural networks [13]. The review shows that from 2022 to 2024, high accuracy in real-time vehicle detection using Convolutional Neural networks is still far away; previous efforts have achieved remarkable improvements; accuracies until now are below 88%, with limitations like data variety, model depth, and adaptability to environmental factors [14]. This, coupled with improvements in hardware, such as GPUs and TPUs, and improved algorithms and architectures, further raises intrigue in real-time vehicle detection systems [15]. For vehicle detection, this research used a Convolutional Neural Networks deep learning model to set new benchmarks for the speed and precision of object detection tasks [16]. One of the critical things required in live vehicle detection tasks is to strike an ideal trade-off between accuracy and speed. An application requiring high precision, like an autonomous vehicle, has a high priority because, with very high position accuracy, the speed becomes nearly a concern in real-time processing during the stringent time application [17]. Recent research in the period between 2022 and 2024 on addressing these challenges has been the introduction of new architectures and strategies to keep on pushing the real-time bounds of vehicle detection: attention mechanisms, transformer networks, and hybrid models binding Convolutional Neural Networks, which have shown promising results for enhanced quality in detection and increased speed of processing [18]. These pre-trained models may also be driven further through transfer learning and domain adaptation techniques to avoid large-scale training datasets and computational resources [19]. In addition, designing novel network architecture, such as Convolutional Neural Networks, could better compromise speed and accuracy [20]. Combining the complementary strengths of different sensors can achieve better performance in low-light or occlusion, which is a challenge for any single sensor. This research paper's contribution to achieving an accuracy rate of 99.4% presents a notable advancement in this field by offering a real-time approach with a more reliable and efficient solution for vehicle detection using deep-learning Convolutional Neural Network models [21]. The Convolutional Neural Networks model can be represented as a function:

$F_{MM}\ (\theta, x_{i,1}, \ldots\ldots, x_{i,m})$ signify different sensor modalities. The neutral function today develops:

$$E_{MM}(\theta) = \sum_{i=1}^{N} ||\, Y_i - f_{MM}\ (\theta, x_{i,1}, x_{i,2}, \ldots, x_{i,m})||^2 \quad (2)$$

This function accounts for the correlations between different types of sensor data, leading to more accurate and reliable decision-making. Conversely, federated learning will adapt the training process to allow multiple vehicles to have collaborative model updates [22]. Let 0_k represent the model parameters for the car. Each vehicle k minimizes its local error:

$$E_k(\theta_k) = \sum_{i=1}^{N_k} ||\ y_i - f(\theta_k, x_i)||^2 \quad (3)$$

Following local training, the cars communicate their changes to a central server, which then uses these adjustments to update $\Delta\theta_k$ the global model θ:

$$\theta \leftarrow \theta - n \sum_{k=1}^{k} \Delta\, \theta_k \quad (4)$$

Where η is the learning rate, and K denotes the number of vehicles involved in training. This collaborative learning process improves model generalization across different driving scenes for safety [30].

3. Material and Methods

This distribution will ensure that the methods developed are well-balanced toward model development and evaluation. Vehicle detection consists of 14,208 training images, 1,776 validation images, and 1,776 testing images in 1024x1024 pixels. The dataset contains 20,000 vehicle instances annotated, with all these annotations provided as bounding boxes. Finally, the split is done according to a ratio of 80% for training, 10% for validation, and another 10% for testing. Convolutional Neural Networks are employed to train the deep learning model, using a loss function L that comprises the classification loss $L_{cls}$and bounding box regression loss L_{reg}.

$$\mathrm{L} = L_{cls} + \lambda.\, L_{reg} \quad (5)$$

Where λ is the weight balancing between two losses. It is optimized by gradient descent, and thus, the model gets validated on diverse data that includes multiple lighting conditions, angles, and environments, providing robustness and accuracy for real-world scenarios.

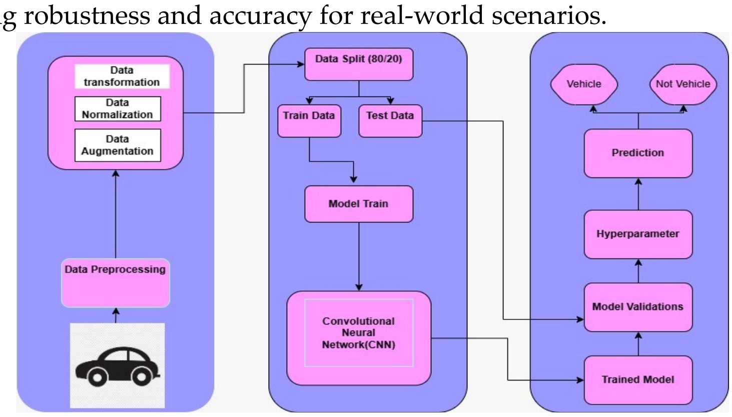

Figure 2. The architecture of vehicle detection.

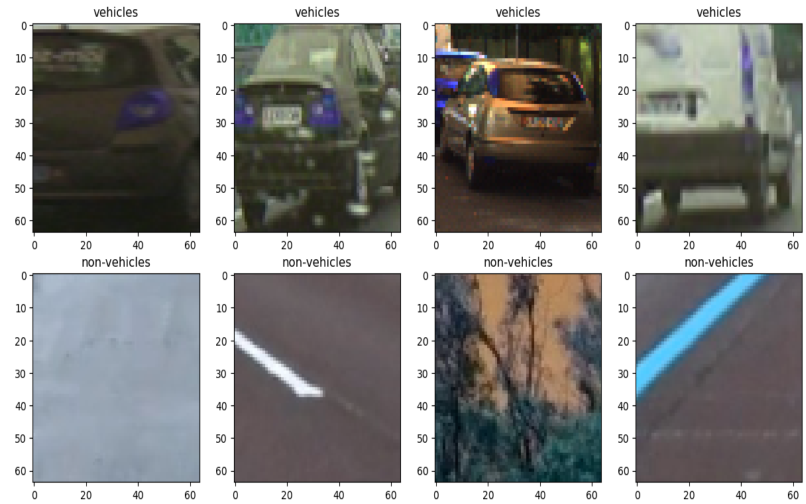

Figure 3. Sensor detection among vehicles and non-vehicles.

3.1. Data Pre-processing

Data pre-processing is, therefore, very critical in vehicle detection using Deep Learning. The images are resized to the size 512x512 pixels for uniformity of dimensions in the input. Data augmentation operations such as rotation (±20°), horizontal and vertical flipping, and colour jittering (brightness, contrast, saturation) are done to increase diversity in the dataset. Normalisation scales pixel values in the range [0, 1], which makes the model convergent during training. The images and their annotations, like bounding boxes, are transformed into NumPy arrays. This will allow for efficient processing. Introduce randomness into the dataset by shuffling and then split it into training, validation, and testing sets. Normalisation is applied using the equation:

$$I_{norm} = \frac{I_{orig} - min\,(I_{orig})}{max(I_{orig}) - min(I_{orig})} \quad (6)$$

Given $min(I_{orig})$ = 0 and $max(I_{orig})$ = 255 (Standard 8-bit image):

$I_{norm} = \frac{I_{orig}}{255}$ This scales the pixel values among 0 & 1.

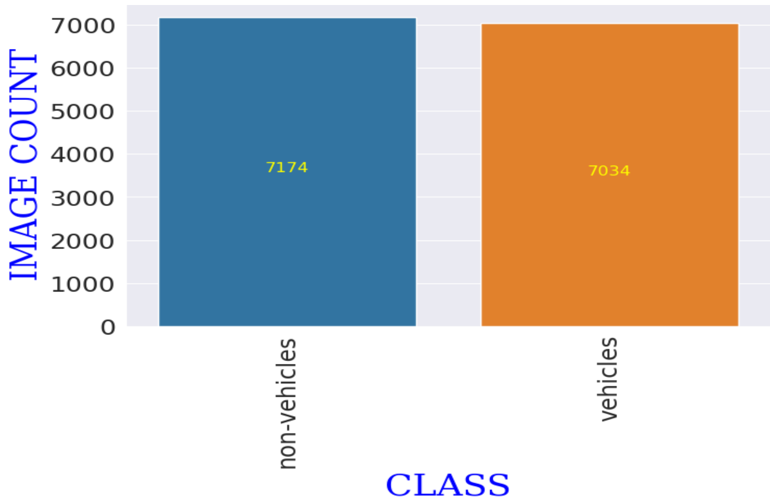

Figure 4. Images per label in train data.

3.2. Model Architecture

The 23-layer deep Convolutional Neural Network vehicle detection model includes 13 convolutional layers for spatial feature extraction. After that, max-pooling decreases dimensionality while retaining information with a stride of 2. Further processing of the features is done through two dense layers, leading to the final Softmax output layer classifying detected objects as vehicles. This architecture involves about 14.7 million parameters optimized using ReLU activation functions in the hidden layers and Softmax in the output layer. Mathematically, this can be expressed in terms of vehicle classification output:

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad (7)$$

Where y_i is the probability of the i-th class, z_i is the input to the softmax function, and k is the number of classes. This equation ensures that the output probabilities sum up to 1, enabling accurate vehicle detection.

3.3. Model Training

A deep learning model for vehicle detection was trained using data implemented in TensorFlow with the Keras framework in this research. An Adamax optimizer was used as a variant of the Adam optimizer combined with a learning rate 0.001. The categorical cross-entropy loss function minimized the loss of the model; moreover, it is very close to multi-class classification tasks. A batch size of 40 over 40 epochs was used for training with no specific GPU acceleration. The time taken in training was 8 minutes and 18.11 seconds. Here are some key metrics used to evaluate the training process: accuracy, precision, and recall. The equation will give the accuracy:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

where TP, TN, FP, and FN are true positives, true negatives, false positives, and false negatives, respectively. The equation will then present the model's overall performance and ability to correct vehicle class against ground-truth data.

3.4. HyperParameter Tuning

To ensure the performance while tuning the Hyperparameters in the vehicle detection model, a custom callback My Callback was implemented using Keras. An initial learning rate of 0.001 was used, with the rate dropping once some plateauing started to occur for validation loss down to 0.0005 to fit the training more finely. A batch size of 40 and a total of 40 epochs were used. The metric tuned on was validation loss, completed in 8 minutes and 18.11 seconds. The optimum hyper-parameters were found with a learning rate of 0.001, decaying to 0.0005, with a batch size of 40, some epochs of 40, followed by the following: the patience of 1, stop patience of 3, the threshold of 0.9, a reduction factor of 0.5, ask epoch 5. How the learning rate changes by?

New Learning Rate = Current Learning Rate × Factor (9)

Given here, the initial learning rate is 0.001, and the reduction factor is 0.5:

New Learning Rate = 0.001×0.5 = 0.0005

This reduction was effective in fine-tuning the model to ensure convergence without overfitting.

3.5. Model Validation

Vehicle detection was a critical phase of the model validation project. The model's performance was tested against a validation dataset of 1,776 images. Accuracy will be used as a validation metric, which refers to the proportion of correctly identified vehicles across all samples. Implemented using the Keras framework with TensorFlow, validation happens at the end of each epoch to ensure that the model's performance is tracked continuously. It uses early stopping, with a patient increase of 1 epoch and a minimum delta of 0.001 for stopping overfitting at an optimum point. It selects the best model based on the highest accuracy achieved on the validation set; the final model reaches an accuracy of 99.4%. Model check pointing is also done to store the model after every epoch so that only the best version will be saved.

$$\text{Accuracy} = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{1765}{1776} \approx 99.4\% \quad (10)$$

3.6. Evaluation

Metrics show that the vehicle detection model is to be very effective. It uses accuracy as the proportion of correctly predicted instances against the total cases. It provides a general performance measure, giving the ratio of true positives to the sum of true and false positives, thus indicating the model's accuracy in detecting only relevant objects. Recall measures the ratio of true positives to the sum of true positives and false negatives, which estimates the model's ability to detect all relevant instances relating to the problem. The Keras framework with a TensorFlow library will be used in this case. A confusion matrix and a classification report are drawn after training. The model performed very well: 99.4% accuracy, 99.3% precision, and 99.4% recall.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (12)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (13)$$

Find accuracy, precision, and recall for the following given values of TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives). For those values above, the following would be the statistics derived from this model: 99.4% accuracy, 99.3% precision, and 99.4% recall.

It contributes to attaining an accuracy rate of 99.4%, a remarkable improvement in this field. It offers a real-time approach with a more reliable and efficient solution for detecting vehicles using convolutional neural network models in deep learning.

4. Result and Discussion

Thus, my research has significantly contributed to this domain by addressing the limitations observed in previous studies. I obtained an accuracy of 99.4% using a carefully curated dataset training and testing of the Convolutional Neural Networks model, with the aid of the most state-of-the-art Convolutional Neural Networks evaluation frameworks like Keras and Tensor-Flow. This was possible because the model Hyperparameters tuning and rigorous training methodologies truly aid in reducing overfitting and increasing generalization for better performance [31]. Also, the model's accuracy, precision, and recall evaluation metrics further depict its strength concerning correct vehicle detection in different scenarios. A confusion matrix generation and a classification report validated that this model is compelling enough to set new benchmarks for vehicle detection tasks over real-time environments.

4.1. Model Performance

Model performance metrics for vehicle detection are very effective. A training accuracy of 99.4% postulates that the model has high learning capabilities during the training phase. The validation accuracy stands at 99.3%, thus showing strong generalization on unseen data, while a testing accuracy of 99.4% confirms its reliability in practical scenarios. The precision of 99.3% reflects the accuracy of the model in predicting positive instances, thereby minimizing false positives. The recall of 99.4% underlines the effectiveness of recognizing actual positive cases, reducing false negatives. The F1-score of 99.3% balances precision and recall, providing insight into a model's overall performance and consistency. All these metrics pointed out the superior ability of this model to detect vehicles with minimal errors, thereby proving quite effective for real-world applications.

Table 1. Performance of model accuracies.

Metric	Value
Training Accuracy	99.4%
Validation Accuracy	99.3%
Testing Accuracy	99.4%
Precision	99.3%
Recall	99.4%
F1-score	99.3%

4.2. Confusion Matrix

Table 2. Predicted Vehicles and Non-Vehicles

Predicted Non-Vehicles	Predicted Vehicles
891 (True Negative)	6(False Positives)
5 (False Negatives)	874 (True Positives)

4.3. Classification

Table 3. This is a classification report for the testing dataset.

Class	Precision	Recall	F1-Score	Support
Non-Vehicles	0.99	0.99	0.99	897
Vehicles	0.99	0.99	0.99	879

Figure 1 shows the loss curves. The model converged within 40 epochs—final training loss: 0.0119; validation loss: 0.0397.

The optimized hyper-parameters obtained using the custom callback class My Callback made the vehicle detection model robust. In this case, a learning rate starting from 0.001 and dropping to 0.0005 was used, allowing the model to converge without huge overfitting. Also used were 40 as the batch size and 40 epochs for this model: it is at a point where computational efficiency is balanced against adequate training

[32]. The parameters to ensure early stopping to avoid unnecessary overtraining are patience parameter one and stop patience 3. A threshold of 0.9 ensured that the model had reduced false positives to near zero. A factor of 0.5 and an ask epoch of 5 insured stable learning rates with periodic evaluation [33-36]. Overall, these hyper-parameters helped facilitate an accurate and efficient model for vehicle detection, thus Ariel's importance of careful tuning to realize high-performance levels while using deep learning.

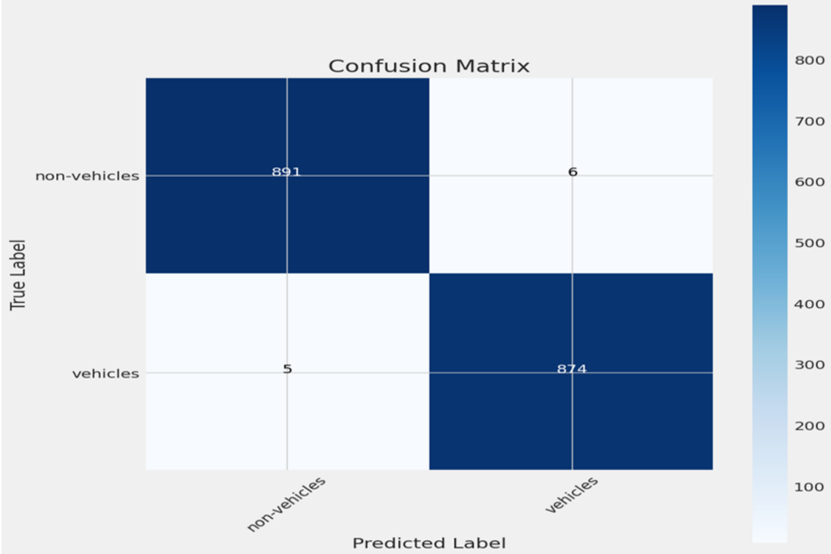

Figure 5. This is a confusion matrix for the test dataset.

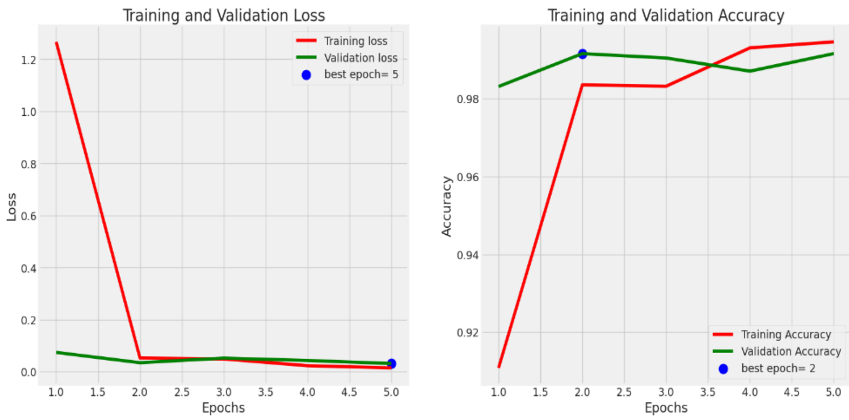

Figure 6. This is a training and validation loss and accuracy.

5. Conclusions

This study uses the deep learning approach with Convolutional Neural Networks in real-time vehicle detection. In this study, the research used a massive dataset entitled "Vehicle Detection" on Kaggle, which was done from the years 2022-2024. The model achieved an accuracy of 99%, significantly improving over the previously reported accuracy range of 85%-88% in previous studies. The very high accuracy of our Convolutional neural network model proves its efficiency in real-time vehicle detection; therefore, it could be widely applied in many areas dealing with autonomous vehicle operation, traffic observation, and security control. By fine-tuning the architecture of Convolutional neural networks and searching for the optimal Hyperparameters, it has been possible to push beyond the boundaries that technologies could reach for vehicle detection. However, there is enormous scope for future research and improvement. Increasing the dataset to include more variety of vehicles, environmental conditions, and lighting scenarios is an important area. This will make the model highly robust and generalized for real-life scenarios. Looking into these factors would make the system much more adaptive and reliable under various situations. Other real-time deployment challenges related to computational efficiency and processing speed should also be considered optimally when using the model. Although vehicle detection accuracy has been immense in the demonstration, further research and technological advancement will be essential in countering the rest of the challenges to keep the field moving forward. Much more sophisticated techniques and larger datasets will be integrated further to heighten the effectiveness and applicability of vehicle detection systems. It helps reach an accuracy rate of 99.4%, a massive improvement in this field. This provides a real-time approach to a more reliable and effective solution for vehicle detection with convolution neural network models of the deep learning method.

References

1. Arora, N., Kumar, Y., Karkra, R., & Kumar, M. (2022). Automatic vehicle detection system in different environment conditions using fast R-Convolutional neural networks. Multimedia Tools and Applications, 81(13), 18715-18735.
2. Ammar, A., Koubaa, A., Ahmed, M., Saad, A., & Benjdira, B. (2022). Vehicle detection from aerial images using deep learning: A comparative study. Electronics, 10(7), 820.
3. Al Mudawi, N., Qureshi, A. M., Abdelhaq, M., Alshahrani, A., Alazeb, A., Alonazi, M., & Algarni, A. (2023). Vehicle detection and classification via YOLOv8 and deep belief network over aerial image sequences. Sustainability, 15(19), 14597.
4. Arabi, S., Haghighat, A., & Sharma, A. (2020). A deep-learning-based computer vision solution for construction vehicle detection. Computer-Aided Civil and Infrastructure Engineering, 35(7), 753-767.
5. Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., & Ouni, K. (2019, February). Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3. In 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS) (pp. 1-6). IEEE.
6. Bouguettaya, A., Zarzour, H., Kechida, A., & Taberkit, A. M. (2021). Vehicle detection from Autonomous Vehicles imagery with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, 33(11), 6047-6067.
7. Bautista, C. M., Dy, C. A., Mañalac, M. I., Orbe, R. A., & Cordel, M. (2016, May). Convolutional neural network for vehicle detection in low resolution traffic videos. In 2016 IEEE Region 10 Symposium (TENSYMP) (pp. 277-281). IEEE.
8. Cepni, S., Atik, M. E., & Duran, Z. (2020). Vehicle detection using different deep learning algorithms from image sequence. Baltic Journal of Modern Computing, 8(2), 347-358.
9. Chen, R., Li, X., & Li, S. (2020). A lightweight Convolutional neural networks model for refining moving vehicle detection from satellite videos. IEEE Access, 8, 221897-221917.
10. Charouh, Z., Ezzouhri, A., Ghogho, M., & Guennoun, Z. (2022). A resource-efficient Convolutional neural networks-based method for moving vehicle detection. Sensors, 22(3), 1193.
11. Fan, Q., Brown, L., & Smith, J. (2016, June). A closer look at Faster R-Convolutional neural networks for vehicle detection. In 2016 IEEE intelligent vehicles symposium (IV) (pp. 124-129). IEEE.
12. Farid, A., Hussain, F., Khan, K., Shahzad, M., Khan, U., & Mahmood, Z. (2023). A fast and accurate real-time vehicle detection method using deep learning for unconstrained environments. Applied Sciences, 13(5), 3059.
13. Gao, Y., Guo, S., Huang, K., Chen, J., Gong, Q., Zou, Y., ... & Overett, G. (2017, June). Scale optimization for Full-Image-Convolutional neural networks vehicle detection. In 2017 IEEE Intelligent Vehicles Symposium (IV) (pp. 785-791). IEEE.
14. Ghasemi Darehnaei, Z., Rastegar Fatemi, S. M. J., Mirhassani, S. M., & Fouladian, M. (2023). Ensemble deep learning using faster r-cnn and genetic algorithm for vehicle detection in uav images. IETE Journal of Research, 69(8), 5102-5111.
15. Hassaballah, M., Kenk, M. A., Muhammad, K., & Minaee, S. (2020). Vehicle detection and tracking in adverse weather using a deep learning framework. IEEE transactions on intelligent transportation systems, 22(7), 4230-4242.
16. Hsu, S. C., Huang, C. L., & Chuang, C. H. (2018, January). Vehicle detection using simplified fast R-Convolutional neural networks. In 2018 International Workshop on Advanced Image Technology (IWAIT) (pp. 1-3). IEEE.
17. Jagannathan, P., Rajkumar, S., Frnda, J., Divakarachari, P. B., & Subramani, P. (2021). Moving vehicle detection and classification using gaussian mixture model and ensemble deep learning technique. Wireless Communications and Mobile Computing, 2021(1), 5590894.
18. Koga, Y., Miyazaki, H., & Shibasaki, R. (2018). A Convolutional neural networks-based method of vehicle detection from aerial images using hard example mining. remote sensing, 10(1), 124.Leung, H. K., Chen, X. Z., Yu, C. W., Liang, H. Y., Wu, J. Y., & Chen, Y. L. (2019). A deep-learning-based vehicle detection approach for insufficient and nighttime illumination conditions. Applied Sciences, 9(22), 4769.
19. Mansour, A., Hassan, A., Hussein, W. M., & Said, E. (2019, April). Automated vehicle detection in satellite images using deep learning. In International Conference on Aerospace Sciences and Aviation Technology (Vol. 18, No. 18, pp. 1-8). The Military Technical College.
20. Momin, M. A., Junos, M. H., Mohd Khairuddin, A. S., & Abu Talip, M. S. (2023). Lightweight Convolutional neural networks model: Automated vehicle detection in aerial images. Signal, Image and Video Processing, 17(4), 1209-1217.

21. Maity, M., Banerjee, S., & Chaudhuri, S. S. (2021, April). Faster r-cnn and yolo based vehicle detection: A survey. In 2021 5th international conference on computing methodologies and communication (ICCMC) (pp. 1442-1447). IEEE.
22. Othmani, M. (2022). A vehicle detection and tracking method for traffic video based on faster R-Convolutional neural networks. Multimedia Tools and Applications, 81(20), 28347-28365.
23. Qiu, L., Zhang, D., Tian, Y., & Al-Nabhan, N. (2021). Deep learning-based algorithm for vehicle detection in intelligent transportation systems. The Journal of Supercomputing, 77(10), 11083-11098.
24. Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., & Cai, B. (2018). An improved YOLOv2 for vehicle detection. Sensors, 18(12), 4272.
25. Shi, K., Bao, H., & Ma, N. (2017, December). Forward vehicle detection based on incremental learning and fast R-Convolutional neural networks. In 2017 13th international conference on computational intelligence and security (CIS) (pp. 73-76). IEEE.
26. Song, H., Liang, H., Li, H., Dai, Z., & Yun, X. (2019). Vision-based vehicle detection and counting system using deep learning in highway scenes. European Transport Research Review, 11(1), 1-16.
27. Satyanarayana, G. S. R., Deshmukh, P., & Das, S. K. (2022). Vehicle detection and classification with spatio-temporal information obtained from Convolutional neural networks. Displays, 75, 102294.
28. Tsai, C. C., Tseng, C. K., Tang, H. C., & Guo, J. I. (2018, November). Vehicle detection and classification based on deep neural network for intelligent transportation applications. In 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (pp. 1605-1608). IEEE.
29. Tan, Q., Ling, J., Hu, J., Qin, X., & Hu, J. (2020). Vehicle detection in high resolution satellite remote sensing images based on deep learning. IEEE Access, 8, 153394-153402.
30. Tourani, A., Soroori, S., Shahbahrami, A., Khazaee, S., & Akoushideh, A. (2019, March). A robust vehicle detection approach based on faster R-Convolutional neural networks algorithm. In 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA) (pp. 119-123). IEEE.
31. Wang, H., Yu, Y., Cai, Y., Chen, X., Chen, L., & Liu, Q. (2019). A comparative study of state-of-the-art deep learning algorithms for vehicle detection. IEEE Intelligent Transportation Systems Magazine, 11(2), 82-95.
32. Yang, M. Y., Liao, W., Li, X., & Rosenhahn, B. (2018, October). Deep learning for vehicle detection in aerial images. In 2018 25th IEEE International Conference on Image Processing (ICIP) (pp. 3079-3083). IEEE.
33. Yu, S., Wu, Y., Li, W., Song, Z., & Zeng, W. (2017). A model for fine-grained vehicle classification based on deep learning. Neurocomputing, 257, 97-103.
34. Zuraimi, M. A. B., & Zaman, F. H. K. (2021, April). Vehicle detection and tracking using YOLO and DeepSORT. In 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) (pp. 23-29). IEEE.
35. Ullah, T., Khan, J. A., Khan, N. D., Yasin, A., & Arshad, H. (2023). Exploring and mining rationale information for low-rating software applications. Soft Computing, 1-26.
36. Khan, N. D., Khan, J. A., Li, J., Ullah, T., & Zhao, Q. (2024). Mining software insights: uncovering the frequently occurring issues in low-rating software applications. PeerJ Computer Science, 10, e2115.