Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/33/20230227

Vehicle detection and tracking in intelligent transportation systems based deep learning

Jiongxiang Chen

The University of Huddersfield, School of Computing & Engineering, Huddersfield, England, UK

U2180170@unimail.hud.ac.uk

Abstract. This research paper focuses on the advancements and optimizations made to fundamental object detection algorithms in vehicle detection. The study explores integrating and reusing CNN (Convolutional Neural Networks) models with other techniques to enhance performance. Three main models, namely Faster R-CNN (Faster Region-based Convolutional Neural Network), Improved SSD (Single Shot Multibox Detector), and YOLOv4 (You Only Look Once v4), are analyzed, showcasing their incremental improvements in accuracy and overall detection performance. However, the increased computational complexity and time demands are trade-offs. The study also presents EnsembleNet, a model combining Faster R-CNN and YOLOv5, which achieves higher average precision values. Another approach involves fusing edge features with CNN models, resulting in faster and more accurate vehicle recognition. The paper predicts future deep learning trends, emphasizing the need for improved hardware capabilities to handle complex models. Integrating deep learning with sensor fusion and edge computing holds promise for intelligent transportation systems.

Keywords: convolutional neural network, vehicle detection, model optimization.

1. Introduction

Intelligent Transportation Systems (ITS) are crucial to modern urban traffic management. Their primary objective is to enhance traffic efficiency, reduce congestion, improve safety, and optimize the utilization of transportation resources through the integration of advanced technologies and intelligent approaches. Vehicle detection and tracking are pivotal in ITS, as they are vital for achieving intelligent and efficient traffic systems.

Currently, mainstream vehicle detection methods are categorized into Motion-Based Methods and Appearance-Based Techniques [1]. However, Motion-Based Methods exhibit clear limitations in adapting to complex scenarios, making them inadequate to meet the demands of modern vehicle detection and recognition. On the other hand, traditional Appearance-Based Techniques, such as Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Histogram of Oriented Gradients (HOG), and Haar-like features (Haar) feature extractors, require manual feature extraction, which is time-consuming and restricts their performance and robustness [2].

The rapid development of deep learning technology has brought revolutionary changes to vehicle detection and tracking. Object detection algorithms based on CNN, such as R-CNN (Region-based Convolutional Neural Network), YOLO, and SSD, have demonstrated remarkable performance and

DOI: 10.54254/2755-2721/33/20230227

effectiveness [3]. These methods leverage deep learning networks' powerful feature learning capabilities to automatically extract rich and sophisticated feature representations from raw images, eliminating the need for manual feature design. Consequently, they possess significant advantages in complex traffic scenarios, substantially enhancing vehicle detection and tracking accuracy and robustness. Moreover, these object detection algorithms continue to be improved and optimized, with noticeable advancements in processing speed and precision. Additionally, their integration with other technologies yields significant performance improvements.

This study will emphasize the advancements and optimizations made to fundamental object detection algorithms over the past few years. The research will thoroughly explore the background, methodology, and outcomes, aiming to analyze the prospective trends and directions for model development in the future.

Furthermore, this essay will delve into applying multiple basic CNN models in vehicle detection through repurposing or other techniques. The study will present the context in which these models were developed, the research procedures employed, and the resultant findings. Subsequently, a discussion will be conducted to evaluate their performance and propose suggestions for further advancements.

2. Optimization of fundamental deep learning models in vehicle detection

Vehicle recognition and detection are crucial and initial steps in intelligent transportation systems. In this field, convolutional neural networks and their variants, such as R-CNN, SSD , and YOLO, are widely used for vehicle detection, localization, and classification in modern ITS.

The mean average precision (mAP) is an evaluation metric that combines recall and precision for object detection. It measures the accuracy of the model. Another important metric in vehicle detection is the Average Processing Time, which reflects the model's computational complexity and time complexity.

2.1. Traditional method and its description

R-CNN: has a great object detection but a multi-stage architecture involving region proposals, feature extraction, and classification. This multi-stage process is computationally expensive and time-consuming, making it slow in real-time applications [4].

YOLO: is a one-stage object detection algorithm that performs bounding box and class probability predictions directly in a single pass through the network. It divides the input image into a grid and assigns the responsibility of predicting the target to each grid cell. YOLO is renowned for its real-time processing speed and efficiency [5].

SSD: is a popular object detection model that combines high accuracy with real-time processing speed. This algorithm is a one-step detection method that directly predicts object bounding boxes and class probabilities in a single forward pass of the neural network [6]. The SSD model divides the input image into grids and assigns the responsibility of predicting objects to each grid cell. This multi-scale approach lets the model detect objects of different sizes and aspect ratios. Furthermore, SSD utilizes a set of default anchor boxes at each grid cell to improve localization accuracy. With its efficient design and effective feature extraction capability, the SSD model has been widely used in various applications including vehicle detection. Its ability to balance precision and speed makes it suitable for real-time scenarios.

2.2. Faster R-CNN

In 2022, Mohamed Othmani reported an advanced vehicle detection and tracking model using deep neural network technology based on the Faster R-CNN model [7]. They modeled a specific general detector based on Faster R-CNN and trained it with Region Proposal Network (RPN) for object proposals and relevant features extracted from a specific CNN architecture. This approach efficiently searches for vehicle object instances in traffic videos. They integrated RPN and Fast R-CNN into a coherent model by leveraging the convolutional advantages of the current neural network formulas. Although the architecture combines various types of layers, including convolutional, pooling,

DOI: 10.54254/2755-2721/33/20230227

rectification, dropout, and normalization layers, it is still based on the Faster R-CNN model and continuously improved.

The primary training data was collected from the GTI vehicle image dataset and represented two classes: vehicles and non-vehicles. 70% of the data was used as the training set, and the remaining data was used as the test set and validation set to evaluate the training performance. The training algorithm of the network used stochastic gradient descent with momentum (SGDM) configured with an initial learning rate of 0.001.

In performance testing, the model achieved a processing speed of 0.54 seconds per frame with an accuracy of 99.24%. The improved model was significantly faster than the R-CNN model, which had an accuracy of 74.45% and a processing time of 57.9 seconds per frame. The improved model's faster performance makes it suitable for real-time object detection and tracking in videos.

2.3. Improved SSD

In 2021, Jingwei Cao et al. reported a smart vehicle detection algorithm utilizing an improved SSD model, and targeted improvements were made to the SSD network model [8]. The KITTI Vision Benchmark Suite was used for vehicle detection experiments, with a dataset of 7,481 sample images with corresponding label files. Of these, 5,985 images were used as the training set and 1,496 as the test set. The optimization process utilized stochastic gradient descent, where the weight parameters of the training network were iteratively updated through the backpropagation algorithm. The network training process involved setting a maximum number of iterations and dynamically adjusting the learning rate during training. The learning rate was progressively reduced at specific intervals to facilitate stable convergence and prevent overshooting. The L2 regularization was applied as the loss function to prevent overfitting and ensure the generalization of the learned features from the training set.

The team conducted performance testing using a custom-built vehicle dataset. The mAP values of the improved SSD under different weather conditions were higher than those of the original SSD. Regarding mean processing time for each frame, the original SSD had a value of 28, while the improved SSD achieved a value of 15. The following two images illustrate the accuracy of the improved SSD in Fig. 1 [8].

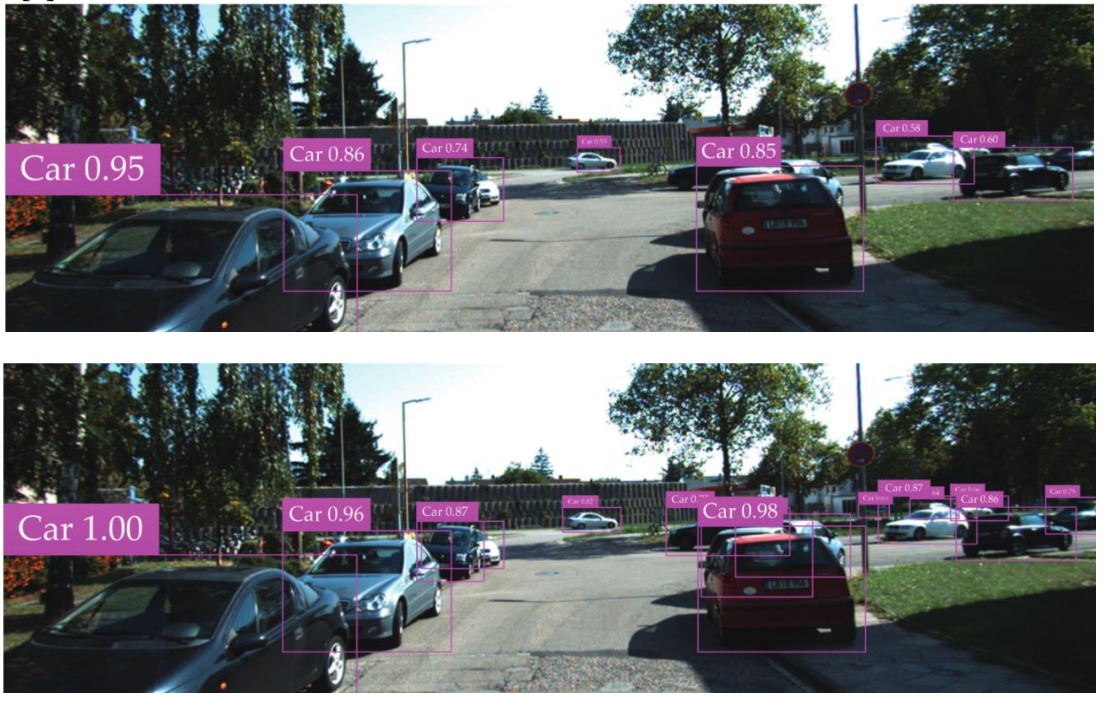

Figure 1. Comparison of SSD before and after improvement [8]. (a) Original; (b) Improved.

DOI: 10.54254/2755-2721/33/20230227

This model exhibited accelerated convergence, effectively tackling the challenge of imbalanced sample data. The performance of the improved SSD network showed significant improvements, primarily attributed to the improvements made to the SSD architecture and loss function. By introducing novel techniques and refining existing components, the proposed vehicle detection algorithm showcased remarkable robustness and adaptability to complex traffic environments and diverse road scenarios, consequently leading to substantial improvements in detection accuracy. However, it should be noted that the computational complexity and time complexity were relatively high, as evidenced by the average processing time values.

2.4. YOLOv4

In 2021, Muhammad Azhad Bin Zuraimi et al. published a YOLO model for real-time vehicle detection, an object detection algorithm, and compared it with previous models, specifically YOLOv4[8]. The Google Open Images dataset, which consists of 600 classes and over 1,700,000 images, was used for training. The validation dataset was split at a ratio of 30% for all the datasets used. The team collaborated using Google Colab and GPU to train the YOLO model.

In the performance evaluation on the Custom dataset, YOLOv4 achieved an mAP score of 82.08%, while YOLOv3 achieved a score of 80.32%. Additionally, the team tested the frame rates for video processing of both YOLO versions on the same hardware setup. The frame rate for YOLOv4 was measured at 14.12 fps, slightly lower than the 16.99 fps achieved by YOLOv3. The conclusion was that YOLOv4 offers higher accuracy than YOLOv3 but comes with increased model and computational complexity. These findings provide insights for future predictions and applying YOLO models in more suitable scenarios.

3. Reuse and integration of CNN models with other techniques in vehicle detection

In recent years, the reuse and integration of CNN models with other techniques in the field of vehicle detection have been a prominent area of research. Incorporating other techniques or reusing multiple CNN models can lead to improved performance compared to using a single model alone.

3.1. Faster R-CNN and YOLO models

In 2021, Usha Mittal et al. reported a deep learning model called EnsembleNet, which integrates two popular single models, Faster R-CNN and YOLOv5, for vehicle detection [10]. The research team analyzed this model. Four datasets were collected, including thermal images, RGB images, 10K videos, and nighttime photos, and data annotation was performed for different vehicle categories in the images. Before inputting the images into the model, image preprocessing techniques such as sharpening and data augmentation were applied to enhance image quality. During the prediction process, the predictions made by two basic model were recorded in separate variables. When the difference between the coordinates was below or equal to the threshold, it signified agreement in predicting the presence of a specific vehicle. In such cases, both models' predictions were considered valid, contributing to the final combined output. Specifically, if Faster R-CNN's confidence score was higher than that of YOLOv5, the detection provided by Faster R-CNN was deemed more reliable and was adopted as the final prediction. Conversely, if YOLOv5 exhibited a higher confidence score, its detection was accepted, and the prediction of Faster R-CNN was rejected. In this ensemble model, Faster R-CNN and YOLOv5 still make individual predictions according to their respective models, but the final prediction is selected based on their respective confidence scores. The flowchart of this chapter is shown in Fig. 2 [10].

DOI: 10.54254/2755-2721/33/20230227

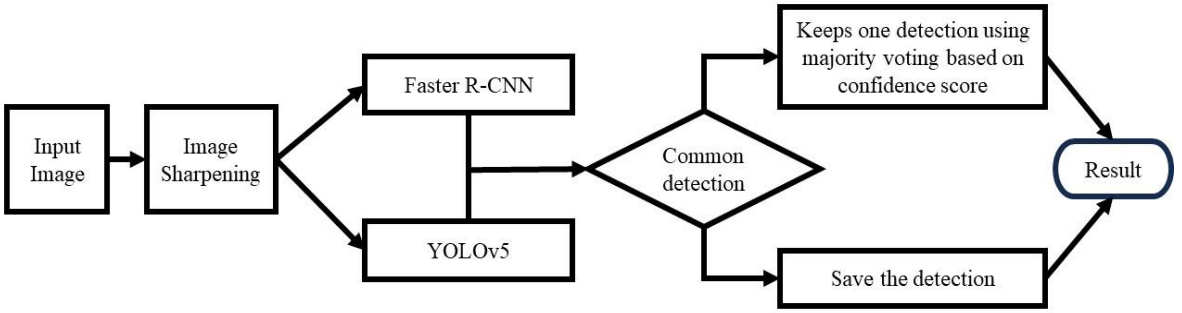

Figure 2. Flow chart of the processing in vehicle detection [10].

In terms of performance on the four datasets, EnsembleNet consistently achieved higher average precision than its base models, with values ranging from 94% to 97%. The ensemble-based deep learning architecture improved prediction performance, function approximation, and classification model accuracy through majority voting, enhancing overall predictions. Additionally, the overall detection performance and accuracy were significantly improved. However, there are also drawbacks associated with the approach. Using two deep learning models increased computational time, resulting in higher time and computational complexity. Therefore, there is room for improvement in runtime efficiency and computational complexity.

3.2. CNN with fused edge features

In recent years, numerous CNN models have been combined with specialized algorithms in vehicle recognition. Traditional vehicle detection models often have complex training processes and overlook edge features. In contrast, edge features carry crucial structural information in images and play a key role in recognizing object boundaries and contours. In 2021, Linrun Qiu et al. proposed a vehicle recognition algorithm based on a convolutional neural network with fused edge features (FE-CNN) [11]. FE-CNN automatically adjusts the parameters of each layer in the CNN during the training process, effectively extracting and fusing features. The Lanczos algorithm was employed for multi-stage image interpolation to preprocess the images and ensure consistent sample sizes for training FE-CNN. The paper extracted vehicle edge images and fused them with the original images to accelerate model convergence and improve recognition accuracy. A traffic training dataset of 6000 vehicle images and 8000 background images was selected. Table 2 presents the annotation results and classification strategy for the dataset. The performance was validated by comparing it with Haar-like vehicle detection model and GoogleLeNet model, and real-world classification tests were conducted using an onboard camera.

Regarding performance in vehicle detection and tracking tasks, FE-CNN achieves a precision rate of 92.35%, a recall rate of 92.13%, and a response time of 134ms. Compared to other models, it is faster and more accurate. This indicates that the FE-CNN model has significant advantages in real-time and complex scenarios.

4. Conclusions

Based on the literature and results from the optimizations of these three basic CNN architectures, it is evident that significant improvements and advancements have been made in vehicle detection over the past few years. Each basic model, namely Faster R-CNN, Improved SSD, and YOLOv4, has demonstrated incremental enhancements in mean Average Precision (mAP) and overall detection performance compared to their predecessors. However, it is worth noting that this progress comes with the inevitable trade-off of increased computational and time complexity, as evident from the observed Average Processing Time. These conclusions also offer valuable insights into the future deep learning trends in vehicle detection. It is anticipated that the accuracy of future deep learning models will continue to improve, further enhancing the capabilities of vehicle detection systems. However, this

DOI: 10.54254/2755-2721/33/20230227

progress will be accompanied by the necessity for continuous advancements in hardware capabilities to accommodate the growing computational demands of these sophisticated models.

In the reuse and integration of CNN models with other techniques in the field of vehicle detection, the future trend in vehicle detection is expected to continue the pursuit of higher accuracy and real-time performance. Researchers will likely focus on refining ensemble-based approaches to minimize computational complexity and develop more efficient hardware solutions to accommodate advanced models. Moreover, exploring novel algorithms that exploit structural information, like edge features, in CNN models will remain a promising direction. Integrating deep learning with other technologies, such as sensor fusion and edge computing, will further enhance the capabilities of vehicle detection systems for the future's intelligent transportation landscape.

References

[1] Yang Z and Pun-Cheng L S C 2018 *IMAVIS* **69** 143–54
[2] Mahaur B, Singh N and Mishra K K 2022 *Multimed Tools Appl* **81** 14247–82
[3] Sharma P, Singh A and Dhull A 2022 *Proceedings of Academia-Industry Consortium for Data Science ed G Gupta, L Wang, A Yadav, P Rana and Z Wang (Singapore: Springer Nature Singapore)* pp 307–21
[4] Girshick R 2015 *2015 ICCV(Santiago, Chile: IEEE)* pp 1440–8
[5] Jiang P, Ergu D, Liu F, Cai Y and Ma B 2022 *Procedia Comput. Sci.* **199** 1066–73
[6] Nagrath P, Jain R, Madan A, Arora R, Kataria P and Hemanth J 2021 *Sustain. Cities Soc.* **66** 102692
[7] Othmani M 2022 *Multimed Tools Appl* **81** 28347–65
[8] Cao J, Song C, Song S, Peng S, Wang D, Shao Y and Xiao F 2020 *Sensors* **20** 4646
[9] Bin Zuraimi M A and Kamaru Zaman F H 2021 *2021 ISCAIE (Penang, Malaysia: IEEE)* pp 23–9
[10] Mittal U, Chawla P and Tiwari R 2023 *NEURAL COMPUT APPL* **35** 4755–74
[11] Qiu, L., Zhang, D., Tian, Y. Najla Al-Nabhan 2021 *J Supercomput* **77**, 11083–11098.