Pattern Recognition Letters 158 (2022) 42–47

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Vehicle detection using improved region convolution neural network for accident prevention in smart roads

Youcef Djenouri[a], Asma Belhadi[b], Gautam Srivastava[c,f], Djamel Djenouri[d], Jerry Chun-Wei Lin[e,*]

[a] Mathematics and Cybernetics, SINTEF Digital, Oslo, Norway
[b] Department of Technology, Kristiania University College, Oslo, Norway
[c] Department of Math and Computer Science, Brandon University, Brandon, Canada
[d] Computer Science Research Centre, Department of Computer Science & Creative Technologies, University of the West of England, Bristol, UK
[e] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway
[f] Research Centre for Interneural Computing, China Medical University, Taichung, Taiwan

ARTICLE INFO

Article history:
Received 12 May 2021
Revised 4 March 2022
Accepted 9 April 2022
Available online 14 April 2022

Edited by: Maria De Marsico

MSC:
68T50
68U15
68U20
68T30

Keywords:
Deep learning
Vehicle detection
Region convolution neural network
Hyper-parameters optimization

ABSTRACT

This paper explores the vehicle detection problem and introduces an improved regional convolution neural network. The vehicle data (set of images) is first collected, from which the noise (set of outlier images) is removed using the SIFT extractor. The region convolution neural network is then used to detect the vehicles. We propose a new hyper-parameters optimization model based on evolutionary computation that can be used to tune parameters of the deep learning framework. The proposed solution was tested using the well-known *boxy vehicle detection data*, which contains more than 200,000 vehicle images and 1,990,000 annotated vehicles. The results are very promising and show superiority over many current state-of-the-art solutions in terms of runtime and accuracy performances.

© 2022 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

The rapid population growth in modern cities has increased the demand for smart technologies for environmental sustainability and safety. Road safety is one of the most critical issues in smart city development when it comes to intelligent mobility [5,13,22]. Accident prevention [16,17] is one of the hot topics in road safety, where the goal is to find an efficient mechanism in predicting accidents before they happened. Computer vision and deep learning are recent technologies for simulating the human visual system. Several technologies based on computer vision and deep learning have been developed for accident prevention problems in road safety environments [9,19]. This paper follows this direction and proposes an end-to-end framework for accident prevention in a road safety environment.

1.1. Motivation

Object detection is the process of identifying objects in a given sequence of images. Several deep learning solutions have been proposed for object detection [3,8,11,12,21]. This work addresses vehicle detection for accident prevention in a road safety environment, motivated by the effectiveness of object detection models in accurately detecting various objects. We believe that identifying arriving vehicles in real time reduces the number of accidents, which is particularly important given the steady increase in urban traffic.

1.2. Contributions

We developed in this work IRCNN-VD (Improved Region Convolution Neural Network), a comprehensive real-time vehicle detec-

* Corresponding author.
E-mail addresses: Youcef.Djenouri@sintef.no (Y. Djenouri), asma.belhadi@kristiania.no (A. Belhadi), SRIVASTAVAG@brandonu.ca (G. Srivastava), Djamel.Djenouri@uwe.ac.uk (D. Djenouri), jerrylin@ieee.org, chun-wei.lin@hvl.no (J. Chun-Wei Lin).

https://doi.org/10.1016/j.patrec.2022.04.012
0167-8655/© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

tion framework that incorporates preprocessing, deep learning, and evolutionary computation. This will first milestone towards preventing an accident in an intelligent transportation and road system. The vehicle data is first cleaned by discarding noises. i.e., outliers, from the overall process, using the SIFT extractor. The improved regional convolution neural network is then applied to find the closest vehicles to the given driver. Several strategies are developed to find the optimal bounding boxes of the vehicles. Evolutionary computation is integrated into the deep learning model to explore the hyper-parameters space of the IRCNN-VD framework. Holding these facts as notes, our key contributions in the paper can be concluded as follows:

1. We are developing a new cleaning algorithm based on the SIFT extractor that will be used to remove unwanted elements from the collected vehicle frames.
2. An accurate object detection model is proposed by adopting the G-RCNN (Granular Region Convolution Neural Network) algorithm [15] to process vehicle image data, by developing efficient strategies for finding the right and the optimal bounding boxes such as hard negative exploration, and multi-scale training.
3. To intelligently explore the hyper-parameters space of the IRCNN-VD, we develop an evolutionary approach. This significantly improves the accuracy of the developed IRCNN-VD in an acceptable period of time.
4. We evaluate the IRCNN-VD through intensive experiments on the challenging BOXY vehicle data. The results show that IRCNN-VD outperforms the baseline algorithms in both runtime and accuracy.

1.3. Outline

The remainder of the paper is organized as follows. Section 2 reviews the main existing object and vehicle detection algorithms, followed by a detailed explanation of the framework proposed in this paper in Section 3. Section 4 presents the performance evaluation. Finally, Section 5 concludes the paper.

2. Related work

[14] developed a joint super-resolution network for solving the vehicle detection problem. The authors used an up-scaling strategy with the generative adversarial network to learn the hierarchical and discriminative features of vehicles with high resolution [18]. developed a convolution neural network to detect vehicles from urban traffic data by combining the traffic and vehicles features with the stream WaveNet [20]. developed a multi-source active fine-tuning algorithm for vehicle detection. The system is based on transfer and unsupervised learning, which integrates the fine-tuning network for annotating the unlabeled vehicle data [1]. provides a deep learning-based system for identifying construction vehicles. It extends the MobileNet model for vehicle detection problems aiming at deploying the model in embedded devices [6]. suggested the use of several object detection models based on Gaussian mixture model with and without Kalman filter and optical flow in deriving acceptable bounding boxes for vehicle data in both urban and highway areas.

The existing solutions for vehicle detection suffer from accuracy and runtime performance in particular for real setting scenarios. Motivated by the success of the recent object detection models in accurately capturing the different objects, this paper explores a novel intelligent framework for vehicle detection to mitigate accidents in intelligent transportation systems.

3. IRCNN-VD framework

3.1. Principle

The proposed IRCNN-VD framework is explained in this section. The method incorporates deep learning, the SIFT extractor, and evolutionary computation for vehicle detection in intelligent transportation systems. As illustrated in Fig. 1, the process in IRCNN-VD includes three steps. The first one is i) data reduction: it uses the sift extractor algorithm in discarding the non-relevant features. ii) Vehicle Detection: in which a faster RCNN algorithm with some improvement is applied to the urban traffic data for identifying the closest vehicle. iii) Hyper-parameters Optimization: in which evolutionary computation methods are performed to find the best hyper-parameters of the IRCNN-VD framework.

3.2. Data reduction

The images of intelligent transportation data are usually high-resolution data, where the number of pixels of each image ranges from 250,000 to more than 4,000,000 pixels. This generates a high number of region proposals, which can reach a billion regions. This yields to high time and memory footprint. To deal with this problem, we developed a data reduction-based strategy to prune the irrelevant pixels from the vehicle data images. The SIFT (Scale-Invariant Feature Transform) extractor [10] is used to figure out the relevant features, and then eliminate the parts in the image which do not contain relevant features. The scale-space function, $S(I,\sigma)$[1], is first defined, which is based on the Gaussian kernel, K. It is given with Eq. (1).

$$S(I,\sigma) = K(I,\sigma) \times I, \tag{1}$$

The Gaussian kernel takes the image I and makes it smaller based on the σ value. We then identify the spatial information of each candidate keypoint based on the interpolation procedure. The spatial interpolated information is computed, which assures the stability of the extracted features. The Taylor function $Y(I,\sigma)$ is defined as interpolation function, given by Eq. (2).

$$Y(I,\sigma) = D + \frac{dY^T}{dI} I + 0.5 I^T \frac{d^2Y}{dI^2} I \tag{2}$$

The descriptor vector for the key points is then calculated by generating different orientation histograms of 4×4 pixel neighborhoods. The next step is to minimize the number of pixels in each image using the SIFT features. The areas of each image that do not contain SIFT features are cropped and eliminated.

3.3. Vehicle detection

After the data reduction step, the object detection algorithm is performed to identify vehicles that are close to the driver. We used the G-RCNN (Granular Region Convolution Neural Network) algorithm [15], which is a recent object detection algorithm that outperforms the state-of-the-art object detection solutions. It is based on the granulation process evolving the derivative information in an uncertain environment such as urban traffic data, where the vehicles can appear, and disappear at any time. Further to the existing G-RCNN, the network is pre-trained using the transfer learning process on the Imagenet dataset[2]. We also use some enrichment through hard negative exploration and multi-scale training. The improved G-RCNN includes the following steps:

[1] In our implementation, we set σ to 1.60.
[2] http://www.image-net.org/.

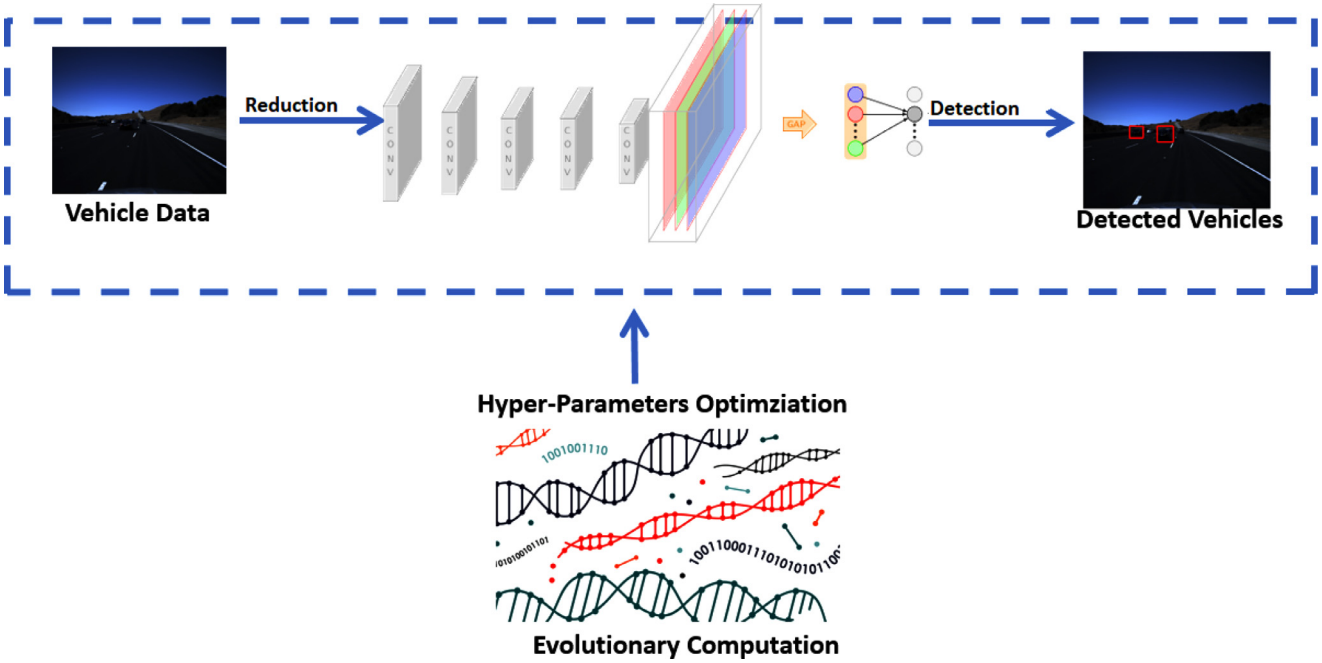

Fig. 1. IRCNN-VD Framework.

1. **Generation and Classification of Region Proposal Candidates:** It has the goal to first generate the regions of interests represented by bounding boxes. It uses a more efficient method based on the convolution neural network for bounding box generation. The regions of images are then classified into vehicles, where the refinement of the results is done by the regression process.
2. **Hard Negative Exploration:** It has the goal to derive the hard negatives, part of images for which the model produces an error in the detection process. These regions are handled again by the model using the reinforcement learning process. This mechanism allows to enhance the performance of the training process and reduce the errors in the learning phase. This also allows the model to learn complex configurations that are frequent in urban traffic data. This step considerably helps for accident prevention. The hard negatives are obtained if its intersection with the ground truth-region is less than 25%.
3. **Multi-Scale Training:** Traditional RCNN is based on a fixed scale for bounding boxes determination. Different scales are integrated into the proposed framework enabling thus to generate bounding boxes with different sizes, which permits identifying vehicles in a real setting scenario. Different batches of bounding boxes are created, each of which contains the bounding boxes of the same number of pixels. Therefore, the region proposal determination is independently applied for each group of bounding boxes.

3.4. Hyper-parameters optimization

This section attempts to identify the best hyper-parameters of the G-RCNN algorithm. Let us consider $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2 \dots \mathcal{P}_{|P|}\}$ to be the set of parameters, where $|P|$ is the number of parameters of the G-RCNN framework. We also define the domain value of each parameter $\mathcal{P}_i$, noted $D(\mathcal{P}_i)$, which contains all possible values of $\mathcal{P}_i$. The configuration space $\mathcal{C}$ is defined by the set of all possible configurations. In this research work, we consider various parameters for optimization such as epochs, batch size, error rate, image size, and kernel size. Every possible configuration is a vector of possible values in $D(\mathcal{P}_i)$ for all $\mathcal{P}_i$ in $\mathcal{P}$. To find the best values for all parameters that yield high accuracy performance of IRCNN-VD, the entire configuration space in $\mathcal{C}$ must be examined. This is high time consuming where the number of possible configuration is the combination of all possible values of the parameters in $\mathcal{P}$, which is determined by $\prod_{i=1}^{|P|} |D(\mathcal{P}_i)|$. The configuration space is huge. For instance, if we only consider 100 possible values for epoch parameter, 10 possible value for error rate and 5,000 possible values for the number of bounding boxes, the size of the configuration space is 5 million configurations. Therefore, traditional search-based strategies such as branch and bound [7], and A* [4] became inefficient for such a high number of configurations. Therefore, the evolutionary computation approaches are applied to accurately find a configuration that is as close to the optimal one as possible. In the following, we define the basic operators of the developed evolutionary computation:

1. **Solution Encoding:** Each solution is the set of values ranging in the domain values of the parameters in $\mathcal{P}$. Each generation in this evolutionary computation contains a fixed number of solutions that should be in a heterogeneous space for better exploration of different regions of the configuration space. The initial generation is recursively created by generating diversified solutions, where the diversification criteria are determined by the distance between solutions.
2. **Crossover:** Crossover operators allow the intensification process, i.e., it intensively explores one region in the configuration space. The following crossover operators are applied for each couple of individuals of the current population. The crossover point is randomly selected from 1 to $|P|$ which allows to divide each individual into two parts, *left side*, and *right side*. The first child takes the left side of the first individual, and the right side of the second individual, while the second child takes the right side of the first individual, and the left side of the second individual.
3. **Mutation:** Mutation operators allow for the diversification process and thus generating solutions far from the current region.

A given parameter of each individual is randomly selected and updated.

First, the initial population is randomly generated, where each individual is created based on the population initialization. The crossover and mutation operators are then applied for exploring the configuration space. To maintain consistent population size, every individual is evaluated making use of the object detection accuracy, while keeping the best individuals and removing the rest. This process is repeated in multiple iterations until the max number of iterations is reached.

3.5. Developed algorithm

Algorithm 1 shows the pseudo-code of the DCNN-TFO algorithm. The input data is the set of n video frames used for training. This information is accompanied with ground truth. The bounding boxes of all detected vehicles in the associated frame are displayed by each ground truth. We used a set of m video frames with the ground truth to test the models we trained. In this way, the accuracy of the developed model can be calculated. The best trained model and the set of bounding boxes of the detected cars from the testing set are the outputs. Using the SIFT method, the process starts by reducing the dimensionality of the images and extracting only the important visual features. The models are then trained, with the evolutionary process generating the parameters for each model. As a result of the training phase, the weights of the best trained model, called *best_model*, are stored. In the inference step, the propagation of the weights of *best_model* is performed for all test data to identify the bounding boxes of the detected vehicles. The algorithm returns both the best model and the set of bounding boxes of the testing data. We note that the training phase, which is performed only once regardless of the amount of data in the inference, is a time-consuming process with multiple optimizations. The inference stage, on the other hand, involves only one loop and relies on straightforward propagation of the learned models from the training phase.

Algorithm 1 IRCNN-VD Algorithm.

Input:$V = \{V_1, V_2, \ldots, V_n\}$: the set of n video frames containing vehicles used for the training stage.
$V' = \{V'_1, V'_2, \ldots, V'_m\}$: the set of m video frames containing vehicles used in the inference stage.
IMAX: Maximum number of generations of the evolutionary algorithm.
Output: *best_model*: the best model generated in the training phase.
Boxes: the set of bounding boxes of vehicles in V'.
$V \leftarrow SIFT(V)$;
$HNM \leftarrow HardNegativeMining(V)$;
$FC \leftarrow FeatureConcatenation(V)$;
$MST \leftarrow MultiScaleTraining(V)$;
$best\_model \leftarrow \emptyset$;
for iter in IMAX **do**
  $current\_model \leftarrow FasterRCNN(HNM, FC, MST, iter)$;
  $best\_model \leftarrow Best\_IoU(best\_model, current\_model)$;
end for
Boxes $\leftarrow \emptyset$;
for $V'_i \in V'$ **do**
  $B \leftarrow \emptyset$;
  $result \leftarrow best\_model(V'_i)$;
  $Boxes \leftarrow Boxes \cup result$;
end for
return $(best\_models, Boxes)$.

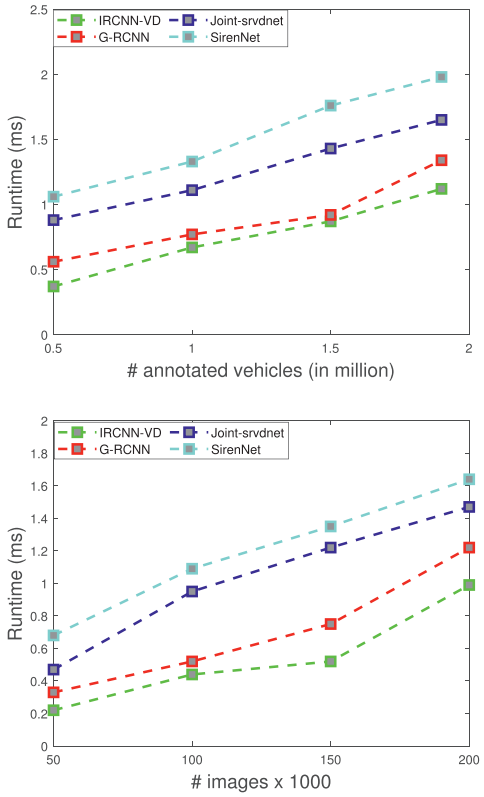

Fig. 2. Runtime of IRCNN-VD versus the state-of-the-art vehicle detection solutions using different number of annotated vehicles and images.

4. Performance evaluation

The performance of the proposed framework IRCNN-VD is analyzed through intensive experimental evaluation. The BOXY vehicle data is used for this purpose [2]. It is a large vehicle detection dataset with almost two million annotated vehicles for training and evaluating object detection methods for self-driving cars on freeways. The computing runtime and accuracy represented by mAP (mean Average Precision) are measured as metrics of comparison. The mAP is largely used to test object detection systems. It can be defined by,

$$mAP = \frac{\sum_{i=0}^{n} AvgP(i)}{n}, \tag{3}$$

where n is considered as the detected objects among all objects, and $AvgP(i)$ is calculated as the precision results at i-rank. For example, the first i-ranked object is then taken into the consideration but ignored others.

The models are implemented on a machine fitted with an Intel-Core i7 processor and combined with NVIDIA GeForce GTX 1070 GPU. The IRCNN-VD is compared with the recent vehicle detection solutions under a varied number of annotated vehicles in databases, as well as a varied number of images.

4.1. Runtime

The purpose of the first experiments is to evaluate the runtime of the IRCNN-VD compared to the baseline vehicle detection solutions; the G-RCNN [15], the Joint-srvdnet [14], and the SirenNet [18]. By varying the number of annotated vehicles from 0.5 to 1.9 million, and the number of images from 50,000 to 200,000 images, Fig. 2 shows that IRCNN-VD outperforms the three baseline algorithms in terms of processing runtime. The runtime of the IRCNN-

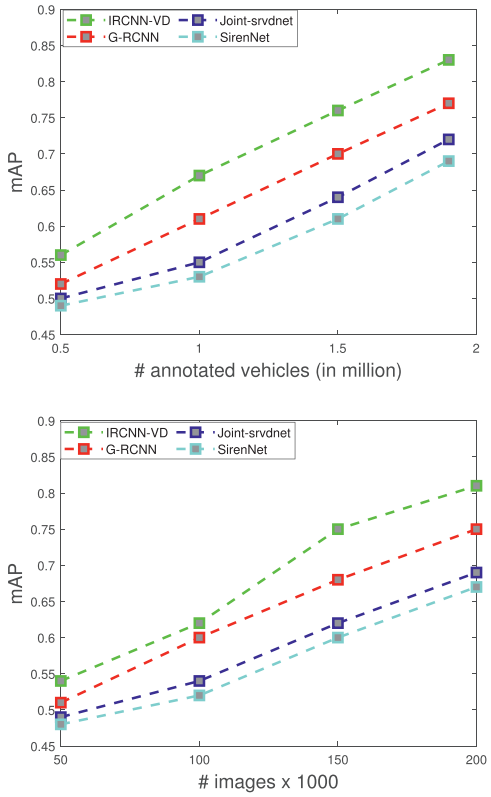

Fig. 3. Accuracy of IRCNN-VD, and state-of-the-art vehicle detection solutions using different number of annotated vehicles, and different number of images.

VD does not exceed 1 mile seconds for processing 1.9 annotated vehicles and 200,000 images, whereas the runtime for the other algorithms reached 2 mile seconds for handling the same number of annotated vehicles and images. These results are the outcome of an efficient exploration of bounding boxes under consideration, as well as a cleaning mechanism that helps remove irrelevant regions during the search.

4.2. Accuracy

The second experiment aims to evaluate the accuracy of the IRCNN-VD compared to the previously mentioned baseline vehicle detection solutions. Fig. 3 shows that IRCNN-VD outperforms the three baseline algorithms in terms of mean average precision. The *mAP* of the IRCNN-VD reached 0.85 for handling 1.9 annotated vehicles and 200,000 images, whereas the *mAP* for the other algorithms goes under 0.75 for dealing with the same number of instances. The proposed strategies during vehicle detection and the efficient procedure for the hyperparameter optimization phase contributed to these results.

4.3. Case study

The last experiment aims to illustrate a real case study for the IRCNN-VD for accident prevention. Fig. 4 presents the original image at the top and the results of both G-RCNN, and IRCNN-VD on the bottom. As shown, the IRCNN-VD can properly detect the closest vehicles to the driver. However, G-RCNN has difficulty capturing such vehicles in real-time. With IRCNN-VD, the driver/robot can have a clear picture of the overall environment and can easily prevent accidents by avoiding the detected vehicles.

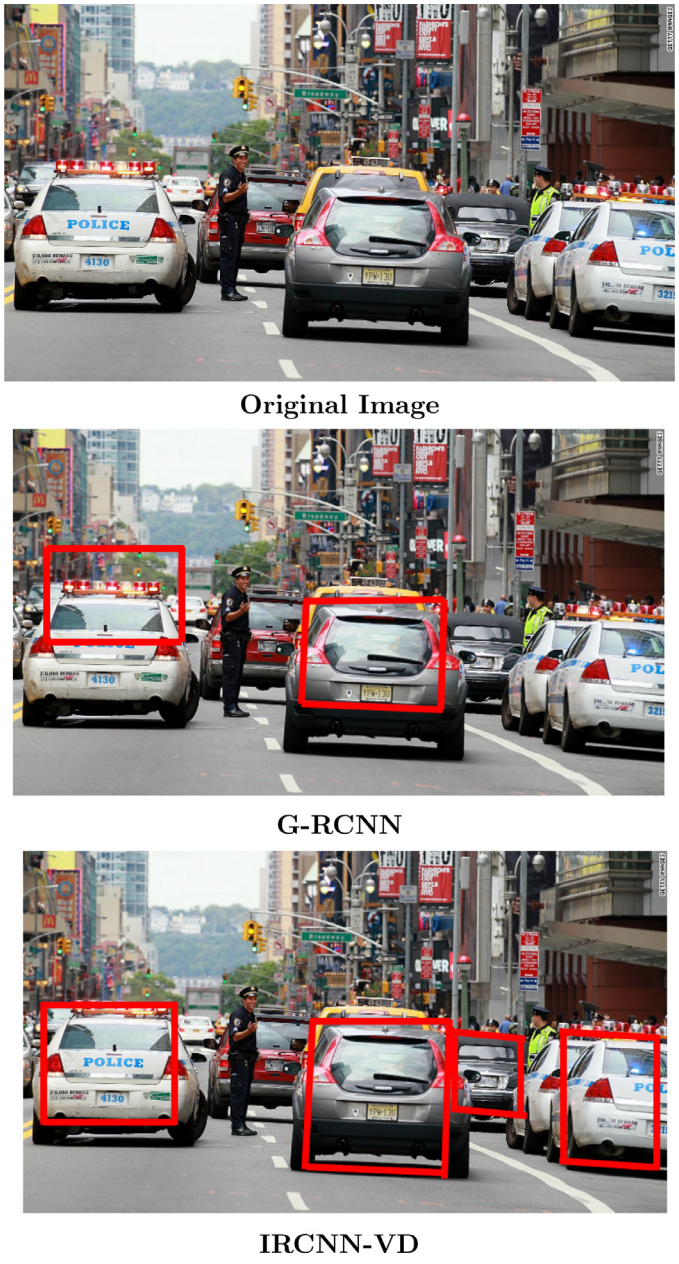

Fig. 4. Demonstration results of IRCNN-VD and the G-RCNN algorithms.

5. Conclusion

This paper introduced a novel approach for real vehicle detection by exploring an improved region convolution neural network. Preprocessing is first performed by reducing the number of pixels of each image using the SIFT extractor. The region convolution neural network is then established to identify vehicles in different scale sizes. The detection process is refined by employing different strategies such as hard negative exploration, and multi-scale training. To accurately find the best parameters of the proposed framework, evolutionary computation is incorporated by developing intelligent genetic operators in exploring the configuration space. Experimental evaluation is executed to validate the applicability of the proposed framework using the well-known *boxy vehicle detection data*. The results are very promising and show the proposed solution outperforms the baseline vehicle detection solutions in terms of runtime and accuracy in detection. In the future, we plan

to extend the proposed solution for handling large and big vehicle data in real time by exploring high-performance computing tools. This could be realized by exploring GPU computing. One possible optimization is to split the images into GPU blocks and perform the training using the massive thread of the GPU hardware. In addition, processing 3D object data of vehicles is a promising direction, with the idea of extending IRCNN-VD in processing various 3D models such as point clouds and voxels. Exploring other evolutionary computations to tune the hyper-parameters of the proposed model is also on our future agenda.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

References

[1] S. Arabi, A. Haghighat, A. Sharma, A deep-learning-based computer vision solution for construction vehicle detection, Comput.-Aided Civ. Infrastruct. Eng. 35 (7) (2020) 753–767.
[2] K. Behrendt, Boxy vehicle detection in large images, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019.
[3] R.R. Boukhriss, E. Fendri, M. Hammami, Moving object detection under different weather conditions using full-spectrum light sources, Pattern Recognit. Lett. 129 (2020) 205–212.
[4] Z. Bu, R.E. Korf, A*+ bfhs: A hybrid heuristic search algorithm, arXiv preprint arXiv:2103.12701 (2021).
[5] C. Chen, B. Liu, S. Wan, P. Qiao, Q. Pei, An edge traffic flow detection scheme based on deep learning in an intelligent transportation system, IEEE Trans. Intell. Transp. Syst. 22 (3) (2021) 1840–1852.
[6] A. Chetouane, S. Mabrouk, I. Jemili, M. Mosbah, Vision-based vehicle detection for road traffic congestion classification, Concurrency Comput. (2020) e5983.
[7] S. Coniglio, F. Furini, P. San Segundo, A new combinatorial branch-and-bound algorithm for the knapsack problem with conflicts, Eur. J. Oper. Res. 289 (2) (2021) 435–455.
[8] H. Feng, L. Zhang, X. Yang, Z. Liu, Incremental few-shot object detection via knowledge transfer, Pattern Recognit. Lett. (2022).
[9] J. Guerrero-Ibañez, J. Contreras-Castillo, S. Zeadally, Deep learning support for intelligent transportation systems, Trans. Emerging Telecommun. Technol. 32 (3) (2021) e4169.
[10] S. Gupta, K. Thakur, M. Kumar, 2D-human face recognition using sift and surf descriptors of faces feature regions, Vis. Comput. (2020) 1–10.
[11] D.K. Jain, et al., An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery, Pattern Recognit. Lett. 120 (2019) 112–119.
[12] M.A. Khan, T. Akram, Y.-D. Zhang, M. Sharif, Attributes based skin lesion detection and recognition: a mask rcnn and transfer learning-based deep learning framework, Pattern Recognit. Lett. 143 (2021) 58–66.
[13] C.-j. Li, Z. Qu, S.-y. Wang, L. Liu, A method of cross-layer fusion multi-object detection and recognition based on improved faster r-cnn model in complex traffic environment, Pattern Recognit. Lett. 145 (2021) 127–134.
[14] M. Mostofa, S.N. Ferdous, B.S. Riggan, N.M. Nasrabadi, Joint-srvdnet: joint super resolution and vehicle detection network, IEEE Access 8 (2020) 82306–82319.
[15] A. Pramanik, S.K. Pal, J. Maiti, P. Mitra, Granulated rcnn and multi-class deep sort for multi-object detection and tracking, IEEE Trans. Emerg. Topics Comput. Intell. (2021).
[16] R.B. Rajasekaran, S. Rajasekaran, R. Vaishya, The role of social advocacy in reducing road traffic accidents in india, J. Clin. Orthop. Trauma 12 (1) (2021) 2–3.
[17] M. Sangare, S. Gupta, S. Bouzefrane, S. Banerjee, P. Muhlethaler, Exploring the forecasting approach for road accidents: analytical measures with hybrid machine learning, Expert Syst. Appl. 167 (2021) 113855.
[18] V.-T. Tran, W.-H. Tsai, Acoustic-based emergency vehicle detection using convolutional neural networks, IEEE Access 8 (2020) 75702–75713.
[19] S. Uma, R. Eswari, Accident prevention and safety assistance using iot and machine learning, J. Reliable Intell. Environ. (2021) 1–25.
[20] X. Wu, W. Li, D. Hong, J. Tian, R. Tao, Q. Du, Vehicle detection of multi-source remote sensing data using active fine-tuning network, ISPRS J. Photogramm. Remote Sens. 167 (2020) 39–53.
[21] F. Xiaolin, H. Fan, Y. Ming, Z. Tongxin, B. Ran, Z. Zenghui, G. Zhiyuan, Small object detection in remote sensing images based on super-resolution, Pattern Recognit. Lett. 153 (2022) 107–112.
[22] Q. Zhou, Z. Qu, C. Cao, Mixed pooling and richer attention feature fusion for crack detection, Pattern Recognit. Lett. 145 (2021) 96–102.