Article

Moving-Vehicle Identification Based on Hierarchical Detection Algorithm

Zhifa Yang ¹ , Yu Zhu ¹, Haodong Zhang ^{1,2}, Zhuo Yu ^{3,*} , Shiwu Li ¹ and Chao Wang ¹

¹ School of Transportation, Jilin University, Changchun 130022, China; yangzf@jlu.edu.cn (Z.Y.); zhuyu19@jlu.edu.cn (Y.Z.); yc18844146145@163.com (H.Z.); Yc_email0426@163.com (S.L.); 15554126295@163.com (C.W.)

² Department of Industrial Engineering, School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

³ School of Mathematics and Statistics, Changchun University of Technology, Changchun 130012, China

* Correspondence: zy15584318055@163.com

Abstract: The vehicle detection method plays an important role in the driver assistance system. Therefore, it is very important to improve the real-time performance of the detection algorithm. Nowadays, the most popular method is the scanning method based on sliding window search, which detects the vehicle from the image to be detected. However, the existing sliding window detection algorithm has many drawbacks, such as large calculation amount and poor real-time performance, and it is impossible to detect the target vehicle in real time during the motion process. Therefore, this paper proposes an improved hierarchical sliding window detection algorithm to detect moving vehicles in real time. By extracting the region of interest, the region of interest is layered, the maximum and minimum values of the detection window in each layer are set, the flashing frame generated by the layering is eliminated by the delay processing method, and a method suitable for the motion is obtained: the real-time detection algorithm of the vehicle, that is, the hierarchical sliding window detection algorithm. The experiments show that the more layers are divided, the more time is needed, and when the number of detection layers is greater than 7, the time change rate increases significantly. As the number of layers decreases, the detection accuracy rate also decreases, resulting in the phenomenon of a false positive. Therefore, it is determined to meet the requirements of real time and accuracy when the image is divided into 7 layers. It can be seen from the experiment that when the images to be detected are divided into 7 layers and the maximum and minimum values of detection windows are 30×30 and 250×250 , respectively, the number of sub-windows generated is one thirty-seventh of the original sliding window detection algorithm, and the execution time is only one-third of the original sliding window detection algorithm. This shows that the hierarchical sliding window detection algorithm has better real-time performance than the original sliding window detection algorithm.

Keywords: machine learning; moving vehicle recognition; sliding window detection algorithm; hierarchical sliding window detection algorithm

Citation: Yang, Z.; Zhu, Y.; Zhang, H.; Yu, Z.; Li, S.; Wang, C.

Moving-Vehicle Identification Based on Hierarchical Detection Algorithm. *Sustainability* **2022**, *14*, 264. <https://doi.org/10.3390/su14010264>

Academic Editor: Maxim A. Dulebenets

Received: 17 November 2021

Accepted: 24 December 2021

Published: 27 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

An intelligent transportation system can serve as a real-time, accurate and efficient integrated traffic management system. It relies on computer technology, information communication technology and automatic control technology to realize the unified scheduling of roads, vehicles and personnel in the traffic environment. Vehicle detection technology is an important part of an intelligent transportation system and has been widely studied in recent years. It mainly includes millimeter-wave radar, lidar, machine vision and so on [1]. However, because both millimeter-wave radar and lidar are difficult to use to distinguish detected targets, machine vision has a wide application range and low price, and can provide scene image information containing rich color and texture information.

Therefore, the vehicle detection method based on machine vision plays an important role in an intelligent transportation system.

At present, advanced optimization algorithms (such as heuristics and meta-heuristics) are widely used in various fields, such as online learning, scheduling, multi-objective optimization, vehicle routing, medicine, data classification and vehicle detection. Haitong Zhao et al. [2] proposed an online-learning-based evolutionary many-objective algorithm to improve the generalization ability. By introducing a learning automaton on the basis of a decomposition-based multi-objective optimization framework, it can acquire convincing performance in determining the convergence of PF. Maxim A et al. [3] proposed an adaptive polyploid memetic algorithm (APMA) to solve the CDT trucks scheduling problem, which improved the solution quality and truck scheduling. Zhi-Zhong Liu et al. [4] proposed the AnD algorithm, which has a simple structure, few parameters, and no complex operators; however, it can achieve high competitive benefits. J. Pasha et al. [5] focused on the transport of sub-assembly modules between suppliers and manufacturers. The authors developed an optimization model and solution algorithm to solve the second sub-problem similar to the vehicle routing problem. They used the CPLEX algorithm to solve the global optimality of the model, and used four metaheuristic algorithms: evolutionary algorithm, variable neighborhood search, tabu search and simulated annealing to solve large-scale problems, so that the new model algorithm is better than other metaheuristic algorithms. Maxim A et al. [6] used a global multi-objective optimization algorithm to solve Pareto front vessel schedules problem, and proposed a multi-objective mixed integer nonlinear optimization model, which can significantly reduce the total services route cost. For the medical field, for example, D'Angelo G. et al. [7] used a method based on machine learning to derive rules and formulas from known data sets to distinguish bacterial from viral meningitis. This method used genetic programming and decision trees to distinguish the types of meningitis and its sensitivity reached 100%. It can be seen that advanced optimization algorithms are applicable to all fields. In the process of vehicle detection, it also plays an important role.

The vehicle detection algorithms based on machine vision mainly include the method based on motion information detection [8–10], the method based on prior knowledge detection [11–13], the detection method based on stereo information [14–16] and the detection method based on machine learning [17–19]. Compared with other detection methods based on machine vision, detection methods based on machine learning are more outstanding in recognition performance and robustness [20]. The detection method based on machine learning is to use some feature descriptors to extract the feature information of the vehicle, and then use the method of machine learning to obtain the classifier, and finally use the classifier to complete the detection of the vehicle. There are many algorithms for vehicle recognition using machine learning. The main idea is to first determine the feature descriptor operators used to describe vehicle features (mainly including Haar-like, HOG, shadow feature, etc.), and then select appropriate machine learning algorithms to train the samples. The commonly used algorithms mainly include the Haar-like feature combined with the AdaBoost algorithm [21–23], HOG feature combined with the AdaBoost algorithm [24], Haar-like + HOG feature combined with the AdaBoost algorithm [25], shadow feature combined with AdaBoost algorithm, etc. [26]. For example, Southall et al. [17] proposed an active learning framework based on the combination of Haar features and Adaboost to achieve vehicle detection on expressways. Chang et al. [18] proposed a vehicle detection method combined with Haar feature and online boosting, which realizes vehicle detection in different environments. Niknejad et al. [19] proposed a variable vehicle model based on HOG features, which realizes the adaptive threshold vehicle detection under urban roads.

In the process of vehicle detection based on machine learning and machine vision, the most commonly used method is to use the sliding window detection algorithm to traverse the whole image and finally obtain the target to be detected. However, the sliding window detection algorithm has the disadvantages of large computation and low real time, which cannot realize the real-time detection of the target vehicle in the movement process. Therefore, based on the analysis of the principle of the existing sliding window detection

algorithm, this paper will improve the existing sliding window detection algorithm, and determine a more suitable algorithm for moving vehicle detection, finally realizing the real-time and accurate detection of vehicle detection.

The structure of this paper is as follows. Section 2 introduces the traditional sliding window detection algorithm. Section 3 introduces the improved hierarchical sliding window detection algorithm and eliminates the flash frame. Section 4 analyzes the performance of the hierarchical sliding window detection algorithm. Section 5 summarizes this paper.

2. Sliding Window Detection Algorithm

2.1. Principle of Sliding Window Detection Algorithm

The sliding window detection algorithm is used to obtain the sub-window by the multiple-scale scanning of images to be detected. Several Haar features, extracted in each sub-window [27], are input to the trained cascade classifier for final classification. Through sliding window detection algorithm diagram (See Figure 1), the working principle of algorithm can be understood. The steps are as follows: Ensure the image size is unchanged and the detection window is enlarged with a fixed ratio. Then slide the detection window after proportional enlargement from left to right and from top to bottom with a fixed step size. Record the sub-window to be detected corresponding to each position. After that, extract the Haar features from the sub-window of image to be detected and enter the trained cascade classifier. If the feature value of Haar features extracted in a sub-window passes through the entire cascade classifier, the sub-window considered the target to be detected is included (See Figure 2). Finally, record the sub-window.

Figure 1. Schematic diagram of sliding window detection algorithm.

Figure 2. Sub-window that passed the test.

2.2. Sub-Window Calculation for Sliding Window Detection Algorithm

The formula of sub-windows generated by sliding window detection algorithm is

$$sum_1 = \sum_{k=0}^{\frac{\min\{W_1, W_2\} - P}{L_2}} \left[\frac{W_2 - (P_i + kL_2) + 1}{L_1} \right] [W_1 - (P_i - kL_2)] \quad (1)$$

where sum_1 is the total number of generated sub-windows; i the i -th layer; n the total number of the layer; P_i the length of the minimum detection window of the i -th layer; p the length of the minimum detection window; L_1 the sliding step of the detection window; L_2 the magnification ratio of the detection window; W_1 the height of image to be detected; and W_2 the width of image to be detected.

2.3. Analysis of Sliding Window Detection Algorithm

The objects of detected image are searched by the sliding window detection algorithm in a multiple-scale and omnidirectional manner. With the given minimum detection window, the sliding is from left to right and from top to bottom, and then the detection window is enlarged to repeat the process until the detection window has the same size as the image to be detected. The algorithm is similar to the exhaustive algorithm, searching for the sub-region of image to be detected. Thus, wherever the object to be detected is, there is always a window that can detect it. The calculation of the algorithm is very large because it needs to intercept a large number of sub-windows in the image to be detected and process each sub-window. Through controlling the step size of each movement of sub-window and the magnification ratio of sub-window, the calculation is improved. However, the accuracy of detection is affected.

3. Hierarchical Sliding Window Detection Algorithm

3.1. Basic Idea of Hierarchical Sliding Window Detection Algorithm

The sliding window detection algorithm needs to traverse each position in the image to be detected, with each detection window of different specifications. However, it is not necessary to process the entire image captured by camera in most driving scenes (see Figure 3), such as the sky, hood of vehicle, fences or green belts.

Figure 3. Schematic diagram of driving scene.

Therefore, when the driving scene vehicle is identified, the ROI (region of interest) is extracted. It should be properly processed to exclude the effects of irrelevant content in the image to reduce the calculation of the algorithm and improve the detection speed.

In the sliding window detection algorithm, it needs to traverse each position in the image to be detected for each detection window. For a detection window of pixel 25×25 , it needs to traverse the entire image from left to right and from top to bottom from the upper left corner of image. It is necessary to extract the Haar features in each sub-window, which are judged by the cascade classifier after calculation.

In the actual image, the vehicle closer to the test vehicle occupies a larger space, and the farther vehicle has a smaller pixel. Therefore, through layering the ROI and limiting the maximum and minimum detection windows in each layer, the specific size detection window can be searched in specific area, which reduces the sub-windows generated by the detection window.

3.2. Principle of Hierarchical Sliding Window Detection Algorithm

The sizes, positions, and shapes of targets to be detected are different for running vehicles, so it should be included the total region of the target, which may appear when selecting the ROI. The red frame is the ROI (See Figure 4), and the region outside the ROI can be ignored, so the calculation can be reduced. The image in the ROI is divided into seven layers. Up to the bottom edge of detection window, the process of the moving detection window from A to B in the first layer is consistent with the sliding window detection algorithm, that is, the fixed-size detection window traverses the area to be detected on the first layer from left to right and from top to bottom. Different from the sliding window detection algorithm, the hierarchical sliding window detection algorithm sets the minimum and maximum values of detection window of each layer. Then, each layer is traversed by the same way, recording the sub-window with the target to be detected. In this way, the calculation can be reduced, with the operation of detection program improved.

Figure 4. Schematic diagram of hierarchical sliding window detection algorithm.

3.3. Flashing Frame Elimination

In hierarchical detection, the detection frame may generate flashing. Among them, the flashing phenomenon of the detection frame refers to the phenomenon that the error detection frame is generated due to recognition errors in one or more frames of images in the detection process, there is no target to be detected in the detection frame, and the error detection frame flashes due to the small number of images in the detection frame. The reason is that fewer images appear in the detection frame. The flashing out means that a certain detection target is in accurate recognition; however, one or more frames of images cannot be recognized. The specific process of eliminating the flashing frame is as follows:

Step 1: The first frame of images acquired by camera is identified by the hierarchical sliding window detection algorithm, and then all the detected targets are numbered. The specific information of each detection frame is saved, including the coordinates of the upper left corner of each detection frame in the pixel coordinate system, the coordinates of the geometric center point of the detection frame, the height and width of the detection frame, etc.

Step 2: The second frame of images acquired by camera is identified by the hierarchical sliding window detection algorithm, and all detection frames including the target to be identified by camera are stored. The repetition rate between detection frames in the first frame image and all the frames in the second frame image is calculated.

Step 3: A threshold is set (set to 0.8) and the entire repetition rate obtained in the second step is compared with the threshold. If it is bigger than the threshold, the target in the first frame image still exists in the second frame image; otherwise, it does not. If both repetition rates are more than the threshold, the detection frame information in the first frame image is replaced by the second, which saves the detection frame information. If both repetition rates are less than the threshold, the information of the detection frame is no longer saved in the second frame image.

Step 4: The same way is used to calculate the repetition rate of each detection frame in the third frame image and all the detection frames in the frame image. When the repetition

rate is bigger than the threshold value, the first frame image is replaced; otherwise, the third frame image is not saved, until the camera stops collecting images.

Step 5: When the number of continuous detection frame is less than 5, the total number of occurrences is more than 5. The inside of the detection frame contains the target to be detected, which is displayed in the image.

Step 6: If the current frame does not appear in the previous frame, it should detect whether the detection frame is eliminated at this position in the previous frame image. If eliminated, it is a flashing frame that needs to be added; if not, the target in detection frame is a new one, and then the target is processed according to Steps 1 to 5. The above method can eliminate the flashing detection frame. Compared with the sliding window detection algorithm, the hierarchical sliding window detection algorithm can better eliminate the flashing frame (see Figure 5).

Figure 5. Eliminated flashing box. (a) When the program is running, three detection windows appear when detecting pictures (b) After the program is processed, we can see that there is only one detection window, which eliminates the flashing of the detection window.

3.4. Sub-Window Calculation for Hierarchical Sliding Window Detection Algorithm

After selection, the ROI is layered, with the maximum and minimum values of the detection window of each layer set. Therefore, we can obtain the number of sub-windows generated in the i -th layer in the first layer by the hierarchical sliding window detection algorithm and the total number of sub-windows.

The formula of the number of sub-windows generated by the first layer is shown as

$$sum_2^1 = \sum_{k=0}^{k=\frac{Q_1-P_1}{L_2}} G_1 \left[\frac{W_1 - (P_1 + kL_2) + 1}{L_1} \right] \quad (2)$$

The formula of the number of sub-windows generated by the i -th layer is shown as

$$sum_2^i = \sum_{k=0}^{k=\frac{Q_i-P_i}{L_2}} G_i \left[\frac{W_1 - (P_i + kL_2) + 1}{L_1} \right] \quad (3)$$

The formula of the total number of sub-windows is shown as

$$sum_2 = \sum_{i=1}^n \sum_{k=0}^{k=\frac{Q_i-P_i}{L_2}} G_i \left[\frac{W_1 - (P_i + kL_2) + 1}{L_1} \right] \quad (4)$$

where sum_2 is the total number of the generated sub-windows; i is the i -th layer; n is the total number of layers; P_i is the length of the minimum detection window of some layer; Q_i is the length of the maximum detection window of some layer; L_1 is the sliding step size of the detection window; L_2 is the magnification of the detection window; w_1 is the

width of the image to be detected; and G_i is the height of the i -th layer. The performance analysis of the hierarchical sliding window detection algorithm follows.

4. Performance Analysis of Hierarchical Sliding Window Detection Algorithm

4.1. Parameter Determination of Hierarchical Sliding Window Detection Algorithm

The work used a camera with a resolution of 1080×720 , and the camera was placed on the position of the driving recorder.

(1) Identification of the ROI

For cameras with different road sections and resolutions, the pixels occupied by the ROI are different. For the training set of this study, driving-related regions can be included in the coordinate regions of (380, 0), (380, 1080), (580, 0) and (580, 1080), respectively, in the pixel coordinate system during the whole video. It can effectively exclude objects, such as the sky above the image, the hood below the image, and the fence or green belt on both sides. Therefore, for this training set, the ROI is enclosed by four points of coordinates—(380, 0), (380, 1080), (580, 0), and (580, 1080);

(2) Determination the layer of the ROI

In this study, the layered processing of the ROI is carried out (see Figure 6). We compare the sliding window detection algorithm of different layers (1–10 layers) to detect 5 images, and obtain the execution time consumed by each frame of image detection for different layers. When the number of layers is lower than 7, even though the execution time is short, there is a phenomenon of a false positive (see Figure 7). When the number of layers is larger than 7, the detection accuracy hardly changes, but as the number of layers increases, the execution time also increases sharply. Therefore, when the ROI is divided into 7 layers, the real-time and accuracy requirements of the program can be met. Finally, for the training set and test set, the ROI is divided into 7 layers.

Figure 6. The execution time of different layered programs.

Figure 7. The phenomenon of false positive.

(3) Determination the size of detection window

Through the study of the training set, the minimum pixel point occupied by vehicles in the images to be checked is 30×30 and the maximum is 250×250 . The minimum and maximum detection windows are tested through the test set; the accuracy meets the detection requirements. Therefore, the size of the minimum detection window of the first layer is set to 30×30 and the maximum detection window is set to 60×60 . The minimum detection window of the seventh layer is set to 200×200 , and the maximum detection window is set to 250×250 . Table 1 shows the specific parameter settings. In the pixel coordinate system, all the hierarchical settings and detection window sizes are discussed, with the unit of pixel. The sliding step of the detection window L_1 is 1; the amplification ratio of detection window L_2 is 1; other parameters can be selected by default.

Table 1. Parameter setting of hierarchical sliding window detection algorithm.

Layers	Layered Setup	Minimum Detection Window	Maximum Detection Window
The first layer	380~410	30×30	60×60
The second layer	410~440	60×60	90×90
The third layer	440~460	90×90	130×130
The fourth layer	460~480	130×130	150×150
The fifth layer	480~500	150×150	175×175
The sixth layer	500~520	175×175	200×200
The seventh layer	520~550	200×200	250×250

4.2. Comparison of Number of Sub-Windows

According to Formula (1) and Table 1, the total number of sub-windows generated by the sliding window detection algorithm is about 196.28 million. According to Formula (2) and Table 1, the sub-windows generated by each layer of the hierarchical sliding window detection algorithm can be acquired (shown as Table 2).

Table 2. Number of sub-windows generated by the hierarchical sliding window detection algorithm.

Layers	The First Layer	The Second Layer	The Third Layer	The Fourth Layer	The Fifth Layer	The Sixth Layer	The Seventh Layer
Number (ten thousand)	96.348	93.558	79.622	39.522	47.762	46.462	130.968

The hierarchical sliding window detection algorithm is used to generate a total of 5.34 million sub-windows, and the number of sub-windows generated by the sliding window detection algorithm is approximately 37 times that of the hierarchical sliding window detection algorithm.

4.3. Running Time Comparison Analysis

Based on the relevant source code of the sliding window detection, it is modified and rewritten on the OpenCV platform to obtain a program suitable for the hierarchical sliding window detection algorithm and tested. The consumed time can be obtained by the OpenCV internal function `getTickCount` through the use of different layers (1–10), the sliding window detection algorithm and the traditional sliding window detection algorithm to detect 10 different images to be detected, we obtain the average time consumed to detect a frame of an image (see Figure 8), where SM represents the traditional sliding window detection algorithm. It can be concluded that when the number of layers is 1, the execution time is the least, and the detection of a frame of an image is about 1/5 that of the traditional sliding window detection algorithm. As mentioned above, the lower the number of layers, the less time is consumed. However, the accuracy of detection cannot be guaranteed, and a

false positive will occur. Therefore, the image to be detected is divided into 7 layers to meet the requirements of accuracy and real-time detection.

Figure 8. Execution time of different layers.

This is the average time consumed by the sliding window detection algorithm and the hierarchical sliding window detection algorithm (the number of layers is 7) to detect one frame of an image (see Figure 9). The resolution of each frame detected by the sliding window detection algorithm is 1080×720 . It takes about 110 ms, and the time required for the hierarchical sliding window detection algorithm is about 35 ms. The detection speed of the hierarchical sliding window detection algorithm is about three times that of the sliding window detection algorithm, with the real-time performance of the system improved.

```

execution time = 35.1873ms
execution time = 34.8758ms
execution time = 35.0078ms
execution time = 35.1486ms
execution time = 34.6575ms
execution time = 34.0271ms
execution time = 34.0005ms
execution time = 34.3613ms
execution time = 34.384ms
execution time = 33.7011ms
execution time = 34.9265ms
execution time = 34.3585ms
execution time = 35.0695ms
execution time = 34.1915ms

```

(a)

```

execution time = 111.845ms
execution time = 112.478ms
execution time = 113.668ms
execution time = 123.101ms
execution time = 114.54ms
execution time = 113.39ms
execution time = 111.197ms
execution time = 111.216ms
execution time = 107.532ms
execution time = 110.602ms
execution time = 108.883ms
execution time = 108.091ms
execution time = 109.815ms
execution time = 109.798ms
execution time = 107.816ms

```

(b)

Figure 9. Time comparison between two detection algorithms. (a) Execution Time required for sliding window detection, it takes about 35 ms (b) Execution time required for hierarchical sliding window detection algorithm, it takes about 110 ms it can be seen that the execution time required by the hierarchical sliding window detection algorithm is 1/3 of that of the original algorithm.

5. Conclusions

The work analyzed the principle of the existing sliding window detection algorithm, finding that there are many disadvantages, such as large calculation and poor real-time performance. The hierarchical sliding window detection algorithm is performed by extracting the ROI of the image, layering, and setting the maximum and minimum values of the detection window of each layer, based on the sliding window detection algorithm, to construct a real-time detection algorithm suitable for moving vehicles. Moreover, a sub-window calculation model of the hierarchical sliding window detection algorithm was derived. By training the training set, the ROI of the image could be obtained. We layered

the ROI, compared the different layers of sliding window detection algorithm to detect a frame of the average consumed time, and combined with precision, and obtained the best number of the layer as 7. By comparison, the number of sub-windows generated by the hierarchical sliding window detection algorithm was about one-thirty-seventh of the original sliding window detection algorithm, which greatly reduced the hierarchical sliding window detection algorithm execution time. At the same time, the minimum pixel point occupied by the vehicle in the image to be detected was 30×30 and the maximum was 250×250 , so the sizes of the maximum and minimum detection window were determined.

This paper also made a detailed study on the flashing frame problems during the detection process. In order to eliminate the flashing box, the method in this paper was as follows: calculate the overlap rate of two consecutive frames, replace it when the overlap rate is greater than the threshold value, stop saving it when the overlap rate is less than the threshold value, and so on until the camera stops collecting images.

The relevant source code based on sliding window detection algorithm was modified and rewritten on the OpenCV platform to obtain a program suitable for the hierarchical sliding window detection algorithm. By comparing with the existing sliding window detection algorithm, the results showed that the detection of one frame resolution was 1080×720 , and the time of hierarchical sliding window detection algorithm was only one-third that of the original sliding window detection algorithm. Therefore, the new hierarchical sliding window detection algorithm can improve the image detection and meet the demand for real-time vehicle detection.

The hierarchical sliding window detection algorithm can meet the real-time requirements of moving vehicle detection without huge computing resources and expensive equipment costs. It is very suitable for occasions requiring high real-time performance. In future research, it can be optimized. Using it for multi-objective recognition of the road in front of the vehicle can better help the driving assistance system and reduce the risk of accidents.

Author Contributions: Methodology, Z.Y. (Zhifa Yang); formal analysis, Y.Z.; data curation, H.Z.; writing—original draft preparation, H.Z.; review and editing, S.L.; handle consultation regarding the paper, Z.Y. (Zhuo Yu); supervision, Z.Y. (Zhifa Yang); project administration, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: The authors fully acknowledge the Jilin province Science and Technology Development Project (20200201225JC), the National Natural Science Foundation of China (Grant No. 51978310), and the Science and Technology Project in the 13th Five-Year Plan Period of Education Bureau of Jilin Province (JJKH20200686KJ, JJKH20201007KJ) for the approved fund for this research, which makes this important research viable and effective.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We are very grateful to everyone involved in the process of publishing a scientific work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, Z.; Bebis, G.; Miller, R. On-road vehicle detection: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 694–711. [[CrossRef](#)] [[PubMed](#)]
2. Zhao, H.; Zhang, C. An online-learning-based evolutionary many-objective algorithm. *Inf. Sci.* **2019**, *509*, 1–21. [[CrossRef](#)]
3. Dulebenets, M.A. An Adaptive Polyploid Memetic Algorithm for scheduling trucks at a cross-docking terminal. *Inf. Sci.* **2021**, *565*, 390–421. [[CrossRef](#)]
4. Liu, Z.-Z.; Wang, Y.; Huang, P.-Q. AnD: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation. *Inf. Sci.* **2018**, *509*, 400–419. [[CrossRef](#)]

5. Pasha, J.; Dulebenets, M.A.; Kavooosi, M.; Abioye, O.F.; Wang, H.; Guo, W. An Optimization Model and Solution Algorithms for the Vehicle Routing Problem with a “Factory-in-a-Box”. *IEEE Access* **2020**, *8*, 134743–134763. [\[CrossRef\]](#)
6. Dulebenets, M.A. A comprehensive multi-objective optimization model for the vessel scheduling problem in liner shipping. *Int. J. Prod. Econ.* **2018**, *196*, 293–318. [\[CrossRef\]](#)
7. D’Angelo, G.; Pilla, R.; Tascini, C.; Rampone, S. A proposal for distinguishing between bacterial and viral meningitis using genetic programming and decision trees. *Soft Comput.* **2019**, *23*, 11775–11791. [\[CrossRef\]](#)
8. Arrospide, J.; Salgado, L.; Nieto, M. Multiple Object Tracking Using an Automatic Variable-Dimension Particle Filter. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 49–52. [\[CrossRef\]](#)
9. Jazayeri, A.; Cai, H.; Zheng, J.Y.; Tuceryan, M. Vehicle Detection and Tracking in Car Video Based on Motion Model. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 583–595. [\[CrossRef\]](#)
10. Heisele, B.; Ritter, W. Obstacle Detection Based on Color Blob Flow. In Proceedings of the Intelligent Vehicles ’95. Symposium, Detroit, MI, USA, 25–26 September 1995; pp. 282–286. [\[CrossRef\]](#)
11. Wang, H.; Lu, R.; Wu, X.; Zhang, L.; Shen, J. Pedestrian Detection and Tracking Algorithm Design in Transportation Video Monitoring System. In Proceedings of the 2009 International Conference on Information Technology and Computer Science, Kiev, Ukraine, 25–26 July 2009; pp. 53–56. [\[CrossRef\]](#)
12. Zhang, J.; Guo, X.; Zhang, C.; Liu, P. A vehicle detection and shadow elimination method based on greyscale information, edge information, and prior knowledge. *Comput. Electr. Eng.* **2021**, *94*, 107366. [\[CrossRef\]](#)
13. Katsilieris, F.; Charlish, A. Knowledge Based Anomaly Detection for Ground Moving Targets. In Proceedings of the 2018 IEEE Radar Conference (RadarConf18), Oklahoma City, OK, USA, 23–27 April 2018; pp. 786–791. [\[CrossRef\]](#)
14. Li, P.; Chen, X.; Shen, S. Stereo R-CNN Based 3D Object Detection for Autonomous Driving. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7636–7644. [\[CrossRef\]](#)
15. Chen, J.; Xu, W.; Peng, W.; Bu, W.; Xing, B.; Liu, G. Road Object Detection Using a Disparity-Based Fusion Model. *IEEE Access* **2018**, *6*, 19654–19663. [\[CrossRef\]](#)
16. Xiao, Z.; Dai, B.; Li, H.; Wu, T.; Xu, X.; Zeng, Y.; Chen, T. Gaussian process regression-based robust free space detection for autonomous vehicle by 3-D point cloud and 2-D appearance information fusion. *Int. J. Adv. Robot. Syst.* **2017**. [\[CrossRef\]](#)
17. Southall, B.; Bansal, M.; Eledath, J. Real-Time Vehicle Detection for Highway Driving. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 541–548. [\[CrossRef\]](#)
18. Chang, W.-C.; Cho, C.-W. Online Boosting for Vehicle Detection. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2009**, *40*, 892–902. [\[CrossRef\]](#)
19. Niknejad, H.T.; Takeuchi, A.; Mita, S.; McAllester, D. On-Road Multivehicle Tracking Using Deformable Object Model and Particle Filter with Improved Likelihood Estimation. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 748–758. [\[CrossRef\]](#)
20. Wen, X.; Shao, L.; Xue, Y.; Fang, W. A rapid learning algorithm for vehicle classification. *Inf. Sci.* **2015**, *295*, 395–406. [\[CrossRef\]](#)
21. Lou, Q. Research and Implementation of Vehicle Identification System Based on AdaBoost Algorithm. Master’s Thesis, Northeastern University, Shenyang, China, 2009. Available online: <https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CMFD2012&filename=1012299069.nh> (accessed on 15 June 2021).
22. Wen, X.; Fang, W.; Zheng, Y. A Vehicle Identification Algorithm Based on Haar-like Features and Improved AdaBoost Classifier. *Electron. J.* **2011**, *39*, 1121–1126.
23. Jin, L.; Wang, Y.; Hua, J. Daytime ahead vehicle detection based on Adaboost algorithm. *J. Jilin Univ. (Eng. Technol. Ed.)* **2014**, *44*, 1604–1608.
24. Bensrhair, A.; Bertozzi, M.; Broggi, A.; Miche, P.; Mousset, S.; Toulminet, G. A Cooperative Approach to Vision-Based Vehicle Detection. In Proceedings of the 2001 IEEE Intelligent Transportation Systems (ITSC 2001), Proceedings (Cat. No.01TH8585), Oakland, CA, USA, 25–29 August 2001; pp. 207–212. [\[CrossRef\]](#)
25. Yang, H. Vehicle Detection Based on HOG and Haar-like Fusion Features. Master’s Thesis, Nanjing University of Posts and Telecommunications, Nanjing, China, 2013.
26. Li, Y.; He, K.; Jia, P. Forward vehicle detection system based on shadow features and Adaboost. *J. Tsinghua Univ. (Sci. Technol.)* **2007**, *47*, 1713–1716. [\[CrossRef\]](#)
27. Viola, P.; Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; pp. 511–518. [\[CrossRef\]](#)