applied sciences

MDPI

Article

An Improved CrowdDet Algorithm for Traffic Congestion Detection in Expressway Scenarios

Chishe Wang [1,2], Yuting Chen [1,*], Jie Wang [2] and Jinjin Qian [1]

[1] School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001, China
[2] Jinling Institute of Technology, Nanjing 210001, China
* Correspondence: cyt1101@aust.edu.cn

Abstract: Traffic congestion detection based on vehicle detection and tracking algorithms is one of the key technologies for intelligent transportation systems. However, in expressway surveillance scenarios, small vehicle size and vehicle occlusion present severe challenges for this method, including low vehicle detection accuracy and low traffic congestion detection accuracy. To address these challenges, this paper proposes an improved version of the CrowdDet algorithm by introducing the Involution operator and bi-directional feature pyramid network (BiFPN) module, which is called IBCDet. The proposed IBCDet module can achieve higher vehicle detection accuracy in expressway surveillance scenarios by enabling long-distance information interaction and multi-scale feature fusion. Additionally, a vehicle-tracking algorithm based on IBCDet is designed to calculate the running speed of vehicles, and it uses the average running speed to achieve traffic congestion detection according to the Chinese expressway level of serviceability (LoS) criteria. Adequate experiments are conducted on both the self-built Nanjing Raoyue expressway monitoring video dataset (NJRY) and the public dataset UA-DETRAC. The experimental results demonstrate that the proposed IBCDet outperforms the commonly used object detection algorithms in both vehicle detection accuracy and traffic congestion detection accuracy.

Keywords: traffic congestion detection; CrowdDet; vehicle detection; Chinese expressway; LoS

check for updates

Citation: Wang, C.; Chen, Y.; Wang, J.; Qian, J. An Improved CrowdDet Algorithm for Traffic Congestion Detection in Expressway Scenarios. *Appl. Sci.* **2023**, *13*, 7174. https://doi.org/10.3390/app13127174

Academic Editor: Seokwon Yeom

Received: 9 May 2023
Revised: 8 June 2023
Accepted: 13 June 2023
Published: 15 June 2023

Copyright: © 2023 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

Transportation is an integral part of modern cities' development and growth, which is closely linked to residents' mobility [1], freight transportation [2], environmental sustainability [3], and more. Expressways, as a fundamental element of the transportation system, play a vital role in urbanization. In recent years, China's expressway network has grown significantly, which led to the expansion of its scale. However, this growth has also led to the development of traffic congestion, which is becoming increasingly severe. Traffic congestion can adversely affect the level of serviceability (LoS) of the road network, leading to poor traffic flow efficiency, environmental degradation, economic losses, and road safety concerns. It is essential to detect congested sections of the road promptly so that the transportation management departments can take immediate measures to alleviate traffic congestion and resolve traffic incidents within a specific area. The above actions will prevent more extensive traffic congestion caused by chain reactions.

Traffic congestion detection involves evaluating the status and volume of vehicles on the road, which is typically determined indirectly using traffic flow parameters such as the number of vehicles, travel speed, queue length, and travel time. Many scholars from different disciplines and fields have analyzed how to use traffic flow parameters to detect traffic congestion and have achieved important research results in this area [4–13]. Based on the technology used to extract traffic parameters, traffic congestion detection techniques can be classified into three categories: traffic congestion detection technology based on sensors [4–6], traffic congestion detection technology based on the Internet of Things (IoT) [7,8],

and traffic congestion detection technology based on machine vision [9–13]. Sensor-based detection methods typically require fixed sensors to obtain vehicle speed, and they calculate the average speed over a specific time to estimate road conditions and determine congestion levels. Although the existing sensor technology can detect vehicle information on the road relatively well, the magnetic field in the detection area is susceptible to external factors. The IoT is an emerging field in modern communication networks that has a significant impact on the development of smart cities [14–16]. Some studies in the literature utilize IoT technologies such as a Vehicular AdHoc Network (VANET) to detect traffic congestion. A VANET is a self-organized, structurally open communication network between vehicles that provides decentralized, multi-hop forwarding data transmission capabilities to collect and aggregate real-time speed and location information related to individual vehicles [17]. However, deploying VANET is quite expensive, which requires that vehicles must be equipped with on-board units, and roadside units must be installed along the road. Additionally, the high mobility of vehicles and dynamic changes in related topologies make wireless transmission challenging. Machine vision-based traffic congestion detection technology is a promising and cost-effective solution for surveillance traffic flow. Cameras can capture high-quality video sequences and are widely available and easy to maintain [18,19]. With the growing deployment of cameras on expressways and the advancement of computer vision technology, machine vision-based traffic congestion detection has become feasible.

Currently, many machine vision-based traffic congestion detection methods are designed for urban roads. However, expressways are relatively more enclosed compared to urban roads, and surveillance cameras on expressways usually have a larger viewing angle, resulting in images with larger background areas. Therefore, it needs to be verified whether traffic congestion detection methods based on urban roads can be directly used for detecting expressway congestion. Issues include detecting small distant vehicles in expressway surveillance images captured by wide-angle lenses as well as the problems of reduced accuracy and increased false negative rates in vehicle detection due to vehicle size discrepancies and occlusions during traffic congestion. Therefore, a method using an improved CrowdDet algorithm is proposed to better detect expressway traffic congestion. Firstly, to better detect congested vehicles in expressway surveillance, the improved algorithm is proposed by using CrowdDet [20] as the baseline detector, introducing the Involution operator [21] with long-distance interaction and the bi-directional feature pyramid network (BiFPN) module [22] for multi-scale vehicle detection, and the improved algorithm is named IBCDet. This can provide an important basis for subsequent traffic congestion detection. Secondly, in order to quantify the degree of traffic congestion, the IBCDet algorithm adds tracking technology to measure the speed of vehicles, and it combines the Chinese expressway LoS criteria to divide the speed of vehicles and then analyze and determine the degree of congestion on the road section. Finally, in order to verify the performance of the proposed method, this paper uses the Nanjing Raoyue expressway surveillance video dataset (referred to as NJRY) to train and test the model. Experimental results show that compared with commonly used object detection algorithms, this method can achieve the best performance in both vehicle detection accuracy and traffic congestion detection accuracy. In summary, the contributions of this paper are as follows:

- The proposal of an improved version of the CrowdDet algorithm (IBCDet) that incorporates the Involution operator and BiFPN modules, leading to enhanced long-distance vehicles and occluded vehicles detection accuracy in expressway surveillance scenarios.
- The development of a vehicle-tracking algorithm based on IBCDet that calculates the running speed of vehicles and utilizes the Chinese expressway LoS criteria for traffic congestion detection.
- Extensive experiments conducted on the self-built NJRY dataset and the public UA-DETRAC dataset, which demonstrate the superior performance of the proposed IBCDet algorithm in both vehicle detection accuracy and traffic congestion detection accuracy compared to commonly used object detection algorithms.

2. Related Works

In this section, this paper primarily reviewed the existing literature on traffic congestion detection methods. Vehicle congestion detection acquires various traffic flow parameters through relevant techniques to evaluate the vehicle congestion status. Due to the low cost and easy maintenance of cameras, which can provide high-quality video sequences, and the continuous development of image processing technology, more and more scholars have started to study machine vision-based traffic congestion detection. Lam et al. [9] use Haar-like features to extract the number of vehicles in the image, calculate the vehicle flow rate per unit time, and combine with a threshold to judge the vehicle congestion status. Tahmid and Hossain [10] proposed a vehicle congestion status evaluation system using texture analysis of images. The method uses the duration of vehicle density as the judgment condition for vehicle congestion detection, detects edge information in the image using the Canny edge detection algorithm, analyzes the edge information for vehicle target detection, and calculates the vehicle density on the road surface. However, hand-designed feature-based methods have high time complexity and are not conducive to real-time detection of traffic conditions. In order to overcome the limitations of the above traffic congestion detection methods, studies in the literature [12,13,23–25] use deep learning-based image classification methods for traffic congestion detection. Kurniawan et al. [24] use a convolutional neural network (CNN) to estimate the traffic state, adjust the size of the monitoring image and convert it into a 100 $\times$ 100 grayscale image as the input for CNN training and testing. The CNN model achieves an average classification accuracy of 89.50% on the monitoring image dataset. Chakraborty et al. [25] labeled images as congested or non-congested states and used YOLO [26], deep convolutional neural networks (DCNNs), and support vector machines (SVMs) for image classification to detect traffic congestion. The results show that the accuracies of YOLO and DCNN are 91.5% and 90.2%, respectively, and the accuracy of SVM is 85.2%. Jin et al. [12] proposed a method of extracting images from a monitoring camera at an ultra-low frame rate, considering that high frame rate videos are difficult to use in real-world situations. They used semantic segmentation to label common items such as vehicles, lanes, and road dividers at the pixel level. Based on the classification of traffic congestion images, they obtained the percentage of road occupancy by vehicles through image transformation to detect traffic congestion at the pixel level. Willis et al. [13] used GoogLeNet to classify images collected from monitoring cameras at intersections of urban roads in London and output the congestion level of the roads. Gao et al. [23] proposed an image-based traffic congestion estimation framework that integrates traffic parameters into the convolutional neural network layer to directly detect traffic congestion. Although classification-based methods perform well in detecting traffic congestion, this approach only classifies images into two states: congested and uncongested, ignoring the various complex scenes that occur during traffic congestion. In addition, the literature [11,27,28] proposed real-time discrimination methods for urban road traffic congestion. He et al. [11] used a road congestion index and network congestion index to, respectively, measure the degree of congestion on urban roads and road networks. Lam et al. [27] proposed the mIOU method, which detects traffic congestion by calculating the ratio of the overlap area between two images taken at a certain time interval and the union of multiple bounding boxes. Since the instantaneous mIOU proposed in the literature [27] performs poorly when the time interval is short, Liu et al. [28] proposed a new weighted mIOU method. First, the boundary box generated by YOLOv4 is used to automatically crop the image to generate the region of interest, and then, the weighted average of the current and previous instantaneous mIOU values is taken to improve the application of the mIOU method in traffic congestion detection.

In addition to the aforementioned machine vision-based traffic congestion detection methods, there are other studies [29,30] that approach traffic congestion detection from different perspectives. Zambrano-Martine et al. [29] characterized all streets in Valencia based on the driving times of vehicles under varying degrees of congestion, using a real traffic model obtained from previous work. They conducted simulation experiments

using SUMO to obtain the driving times of vehicles on different road segments under different traffic saturation levels. They used a regression strategy to adjust the curves and obtain an expression that describes the driving times. Costa et al. [30] utilized the Traffic Telco Big Data (Traffic-TBD) structure to provide micro-level road traffic modeling and prediction. All of the above methods are targeted at urban road traffic congestion, but due to the differences between highway and urban road conditions, traffic congestion detection methods based on urban roads cannot be directly applied to highways, so further research is needed. Considering the above issues, Oumaima et al. [8] utilized the traffic information provided by the Vehicular Ad Hoc Networks to propose a mobile model based on the Markov chain to solve the traffic congestion detection problem in multi-lane highways. Mu et al. [31] considered that images taken by cameras on highways have richer scenes and used two classic convolutional neural networks, AlexNet and GoogLeNet, to classify highway congestion states. Due to the fact that cameras on highways are usually located at higher positions and focus on a larger field of view, the background area of the generated images is larger and vehicle detection is more difficult. In addition, expressways have stronger closure than urban roads, and once congestion occurs, it lasts longer, requiring an efficient method to detect traffic congestion in real time. Unlike the above articles, this paper proposes a new method for detecting traffic congestion on expressways based on an improved CrowdDet algorithm.

3. Methodology

This paper proposes a novel traffic congestion detection method specifically designed for the surveillance scenario of expressways, considering challenges such as small vehicle sizes and vehicle occlusion. Firstly, we collected a dataset of surveillance videos from the Nanjing Raoyue expressway. Then, we improved the vehicle detection rate by introducing the Involution operator and BiFPN module into the CrowdDet algorithm, resulting in the IBCDet algorithm. To achieve more accurate traffic congestion detection, we incorporated a tracking algorithm into the IBCDet algorithm to measure vehicle speeds. By combining the measured speeds with the LoS criteria for the Chinese expressway, we categorized the vehicle speeds into congestion levels, enabling effective traffic congestion detection. The overall architecture of the traffic congestion detection method is illustrated in Figure 1.

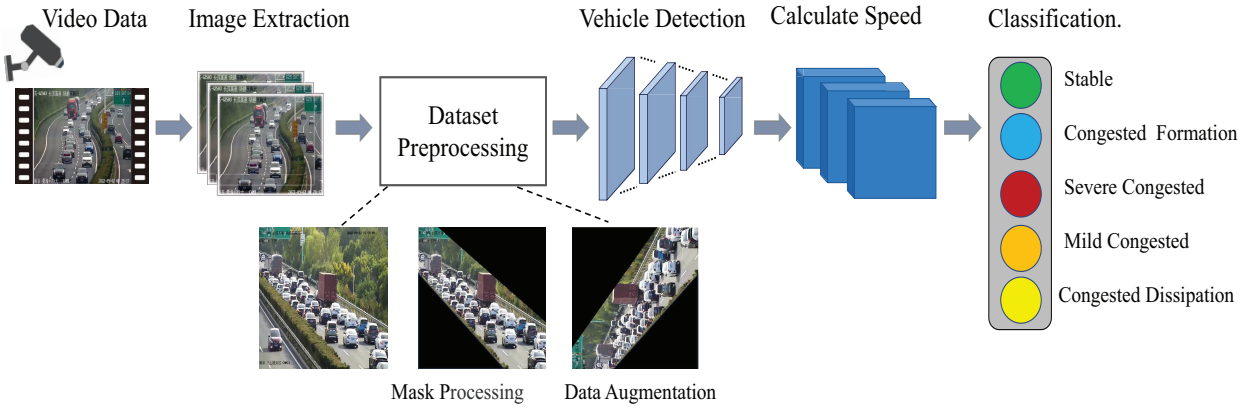


Figure 1. Traffic congestion detection architecture diagram.

This paper describes the generation of training and testing datasets for the detection of traffic congestion on one-way roads by extracting, filtering, and labeling frames from surveillance videos of the Nanjing Raoyue expressway. To improve the accuracy of the detection results, the original video images are cropped to eliminate irrelevant background information, and the areas of interest are manually delineated. In addition, uninteresting areas are masked to prevent interference from roadside disturbances on the detection results. To prevent overfitting, data augmentation techniques are used to expand the small number of samples.

3.1. Baseline Detector

The primary objective of this paper is to improve vehicle detection performance under severe occlusion conditions, ultimately enhancing the accuracy of traffic congestion detection. In traffic congestion scenarios, vehicle occlusion poses a problem similar to that of dense pedestrian detection. Therefore, this paper selected the CrowdDet algorithm [20] as the baseline detector, as it addresses the problem of detecting dense pedestrians. The detector has demonstrated good performance on the CoCo [32], CityPerson [33], and CrowdHuman [34] datasets, which contain lightly, moderately, and heavily occluded pedestrians, respectively. This is similar to vehicle detection under free-flow, slow-flow, and congestion conditions on expressways. The network architecture of the CrowdDet algorithm is shown in Figure 2.

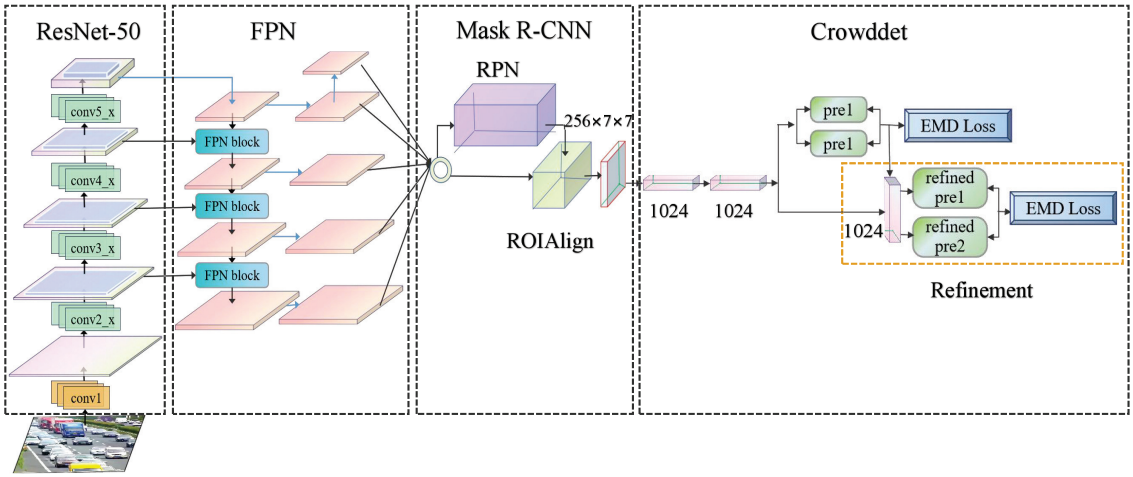


Figure 2. CrowdDet network architecture.

The CrowdDet algorithm mainly consists of the ResNet-50 [35] backbone network, the feature pyramid network (FPN) [36], the Mask R-CNN [37], and the Crowddet component used to simultaneously perform classification and regression tasks. The ResNet-50 backbone network is a residual network structure widely used for feature extraction in images due to its excellent performance. FPN is a feature pyramid network that combines high-level and low-level feature maps in a top–down structure to improve multi-scale detection accuracy. Mask R-CNN uses RPN to propose bounding boxes on the feature map, and then, it selects the corresponding window based on the proposal boxes on the feature map. As the window sizes are different, ROIAlign is used for standardization, resulting in a proposal box feature map of $256 \times 7 \times 7$.

The CrowdDet algorithm first considers that when multiple objects overlap severely, a proposal box corresponding to a single object will lead to a decrease in performance. Therefore, for each proposal box b_i, a set of ground-truth instances $G(b_i)$ is predicted instead of a single object. This is shown in Equation (1):

$$G(b_i) = \{g_i \in G \mid IOU(b_i, g_i) \geq \theta\} \tag{1}$$

where G is the set of all ground-truth boxes, and θ is the given IOU threshold.

For each proposal box b_i, most methods use a detection function to predict a pair (c_i, I_i) to represent the associated instance. However, the CrowdDet algorithm generates a set of predicted instances $P(b_i)$ by introducing K detection functions, as shown in Equation (2):

$$P(b_i) = \left\{ \left(c_i^{(1)}, I_i^{(1)}\right), \left(c_i^{(2)}, I_i^{(2)}\right), \ldots, \left(c_i^{K}, I_i^{K}\right) \right\} \tag{2}$$

where K is a given constant, representing a set of K pairs of predictions, where c_i^K and I_i^K are the class label confidence and relative coordinates of the K-th prediction of box b_i.

Then, in order to minimize the discrepancy between the predicted instance set $P(b_i)$ and the ground-truth instance $G(b_i)$ set corresponding to proposal box b_i, the EMD loss is designed, as shown in Equation (3):

$$L(b_i) = \min_{\pi \in \Pi} \sum_{k=1}^{K} \left[L_{cls}\left(c_i^{(k)}, g_{\pi_k}\right) + L_{reg}\left(I_i^{(k)}, g_{\pi_k}\right) \right] \tag{3}$$

where π represents a specific permutation of $(1, 2, \cdots, K)$, with the k-th element being π_k, and g_{π_k} is the ground-truth box corresponding to π_k. $L_{cls}(\cdot)$ and $L_{reg}(\cdot)$ represent the classification loss and the bounding box regression loss, respectively.

To better detect objects in occluded scenes, the CrowdDet algorithm proposes Set NMS. Set NMS inserts an extra detection to determine whether two boxes come from the same proposal before suppressing one with NMS. If they do, the suppression is skipped. Using Set NMS in conjunction with multi-instance prediction can achieve significant improvements in occlusion detection. The CrowdDet algorithm also introduces an optional Refinement module, which takes the prediction as input and performs a second round of prediction with the proposed features to correct possible errors. All methods in this paper use the Refinement module.

3.2. The Improved Vehicle Detection Network Model Based on CrowdDet Algorithm (IBCDet)

The cameras installed along expressways typically have a larger viewing angle, which can make it challenging to accurately detect small target vehicles at far distances. In such scenarios, the accuracy of vehicle detection is often low due to the small coverage area, blurred images, and limited feature information. In this paper, we address this issue by introducing the Involution operator [21] into CrowdDet. Involution is a novel type of neural network operator that differs from the traditional convolutional operator in terms of spatial sharing and channel specificity. It exhibits spatial specificity and channel sharing, which allows it to use larger convolution kernels to aggregate contextual information over a wider spatial range, thus overcoming the difficulties of long-range interactions in the model. Additionally, it can adaptively allocate different weights to different spatial positions, thereby enabling richer feature information extraction. As shown in Figure 3, first, the feature vector at a point on the input feature map is subjected to a fully connected operation and a reshape transformation to unfold into a K × K × G kernel shape, thus obtaining the Involution kernel corresponding to the coordinate point. Then, the Multiply–Add operation is performed with the feature vector in the neighborhood of this coordinate point on the input feature map to obtain the final feature vector. Involution can use larger 7 × 7 convolution kernels to capture long-range interactions and dynamically generate kernel parameters based on the input feature map at different positions, thereby facilitating the aggregation of contextual semantic information.

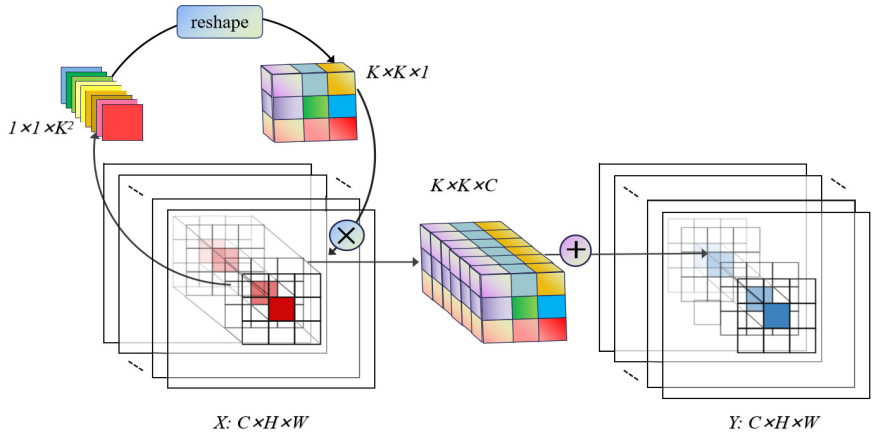


Figure 3. Schematic illustration of Involution.

On the other hand, vehicles on expressways have different sizes, and when traffic congestion occurs, vehicles are densely packed and prone to occlusion problems. In order to solve the occlusion vehicle detection problem under congested conditions and improve the feature extraction performance of occluded objects, the baseline detector CrowdDet algorithm uses the FPN proposed by Lin et al. [36], but the simple and crude fusion method of FPN has limitations on accuracy improvement. Tan et al. proposed BiFPN [22], which introduces learnable weights to learn the importance of different input features and repeatedly applies top–down and bottom–up multi-scale feature fusion, as shown in Figure 4.

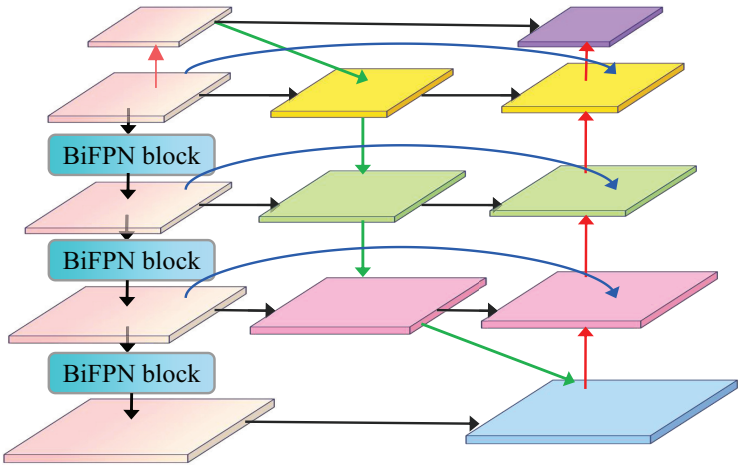


Figure 4. BiFPN module structure.

To this end, this paper modified the ResNet-50 backbone network of the CrowdDet algorithm, using Involution to replace the ordinary convolutional layer, and proposed InNet50. InNet50 keeps the first four parts of ResNet-50 unchanged, replaces the fifth convolutional layer with the Involution operator, denoted as Inv5, and replaces FPN in the CrowdDet algorithm with BiFPN. The proposed InNet50 network is combined with the BiFPN module. By introducing the Involution network and BiFPN module in the CrowdDet algorithm, the accuracy of vehicle detection in congested scenarios on expressways can be improved. This method is called IBCDet in this paper, and the specific network diagram of IBCDet is shown in Figure 5.

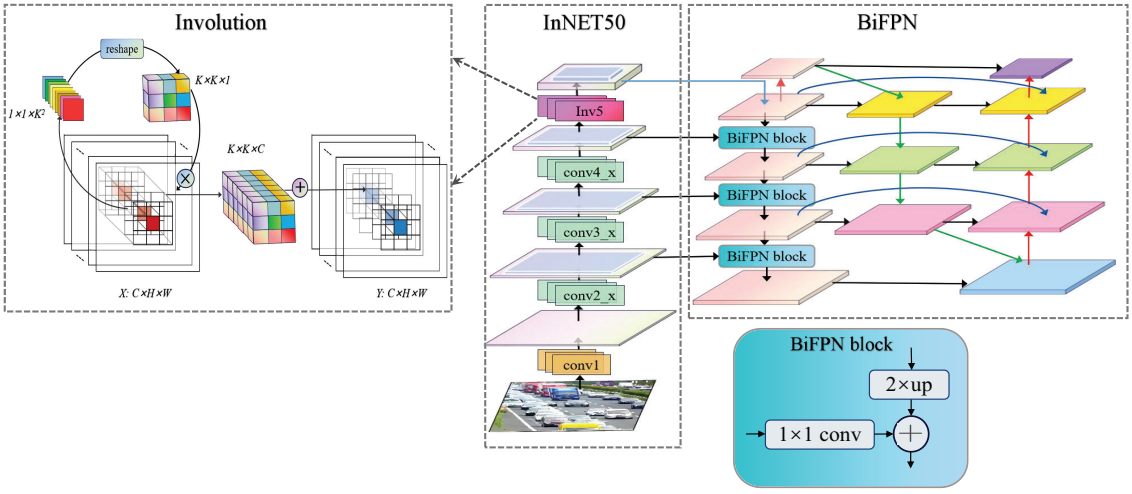


Figure 5. The IBCDet feature extraction and fusion network.

3.3. Traffic Congestion Detection Based on IBCDet

Kerner [38] proposed that the primary indicator of traffic congestion is the decrease in traffic capacity, which can be quantified by a reduction in vehicle speed. When traffic congestion occurs, the decrease in vehicle speed has a dynamic process of propagation from the front to the back, and this dynamic process is the occurrence of traffic congestion. In this study, this paper use the reduction in vehicle speed as a key feature point to identify the occurrence of traffic congestion. To be more specific, this paper adopt the vehicle running speed threshold division method to detect the status of traffic congestion.

Due to the use of vehicle speed thresholds for traffic congestion detection in this paper, the first step is to obtain speed information based on the detected number of vehicles. Currently, traffic monitoring systems used for speed measurement primarily rely on sensors. Although these sensors are widely used, they are complex to install, expensive in terms of equipment cost, and require frequent maintenance. Therefore, developing an economical speed measurement method is necessary in the field of traffic. Given that existing traffic systems are typically connected to video cameras and image processing techniques facilitate video analysis, this paper employs machine learning methods for speed estimation. Specifically, the IBCDet algorithm combined with DeepSort technology [39] is used to track the detected vehicles, as illustrated by the vehicle running trajectory diagram in Figure 6. Then, using the Euclidean distance, the distance traveled by the tracked vehicles between two consecutive frames is calculated based on the centroid points. The calculated distance is divided by the time difference between the frames to estimate the speed. The specific equation is shown in Equation (4):

$$V = \frac{\sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}}{t_2 - t_1} \tag{4}$$

where the vehicle is located at position (x_1, y_1) at time t_1 and at position (x_2, y_2) at time t_2.

The above equation calculates the speed of a single vehicle at a given moment. In order to more accurately reflect the speed of the vehicles in the video, we will calculate the average speed of all vehicles at a specific time, as shown in Equation (5).

$$V_s = \frac{\sum_{i=1}^{N} V_i}{N} \tag{5}$$

where N is the number of vehicles detected by the IBCDet algorithm in the video, and V_i represents the speed of the i-th vehicle. Reflecting on this equation, the higher the vehicle detection rate, the higher the accuracy of V_s to some extent.

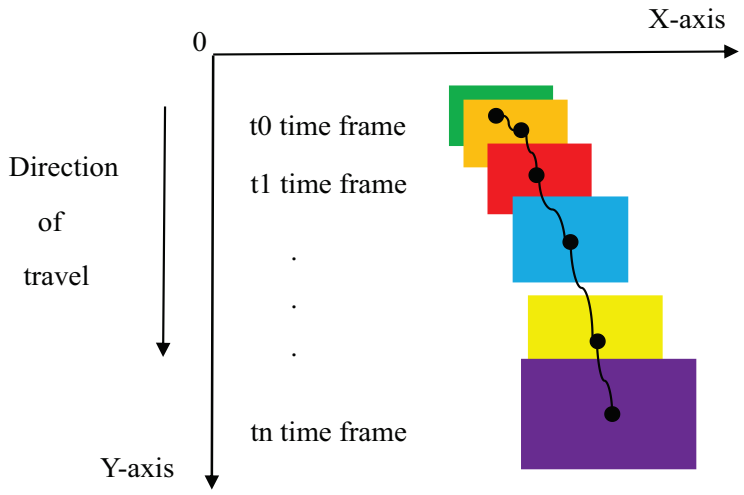


Figure 6. Vehicle running trajectory schematic diagram.

After obtaining the vehicle speeds, this paper primarily classifies the average vehicle speed V_s based on the LoS for expressways in the Chinese *Technical Standards for Highway Engineering* [40]. By observing the changes in average vehicle speed, the entire process of vehicle congestion is divided into five traffic states: stable state, congestion formation state, severe congestion state, mild congestion state, and congestion dissipation state. This study establishes a correlation between the changes in V_s and the five states of traffic congestion, providing clear criteria for their identification. This enables a more accurate assessment of the extent and dynamics of traffic congestion. The specific discrimination criteria are presented in Table 1. It should be noted that the specific speed thresholds may slightly vary across different regions and road sections. The discrimination criteria provided in this study are universal and derived from experimental observations. Although the scenarios and durations of expressway congestion can vary, the proposed method in this paper is applicable to most scenarios, allowing for an accurate classification of the entire process of traffic congestion formation and dissipation.

Table 1. Division of expressway congestion status.

Status	Discrimination Standard	Vehicle Speed
Stable state	Smooth traffic flow, with larger gaps between vehicle.	80 km/h $< V_s \leq$ 120 km/h
Congestion formation state	Smaller gaps between vehicles, with a sharp decrease in speed in a short period of time and traffic gradually slows down.	40 km/h $< V_s \leq$ 80 km/h
Severe congestion state	Stop-and-go traffic, with very small gaps between vehicles.	$V_s \leq$ 25 km/h
Mild congestion state	Very small gaps between vehicles, with significant impact on traffic flow, but vehicles are still able to move forward.	25 km/h $< V_s \leq$ 40 km/h
Congestion dissipation state	Smaller gaps between vehicles, with a rapid increase in speed and traffic gradually becoming smooth again.	40 km/h $< V_s \leq$ 80 km/h

4. Experimental Results and Performance Analysis

To evaluate the traffic congestion detection performance of the proposed IBCDet algorithm in this paper, we first conducted experiments on a self-built Nanjing Raoyue expressway dataset using the IBCDet algorithm. The main goal was to test the vehicle detection performance of different improvement schemes on the self-built dataset. Next, experiments were conducted on the UA-DETRAC traffic public dataset captured using a Cannon EOS 55D camera, which showed that the proposed model has robustness in vehicle detection. Finally, the IBCDet algorithm was combined with a tracking algorithm to measure the vehicle speed in a monitored video segment of the Nanjing Raoyue Expressway, and the congestion level was determined based on the Chinese expressway LoS criteria, enabling the evaluation of the traffic congestion detection performance based on the IBCDet algorithm.

4.1. Dataset

This paper include a self-built dataset called the Nanjing Raoyue expressway dataset (NJRY) and a public dataset called UA-DETRAC. The NJRY dataset is provided by the funding of the project, it comes from the surveillance video of Nanjing Raoyue expressway, and this paper has obtained the right to use it. It consists of road surveillance videos cap-

tured on the Nanjing Raoyue expressway under good lighting conditions. The videos were processed by extracting frames at one-second intervals, resulting in a total of 3081 images. These images were then annotated using labeling tools, and data augmentation techniques were applied to expand the dataset to 6162 images, which were used as the experimental dataset for algorithm evaluation. The dataset was divided into training, testing, and validation sets in a ratio of 7:2:1.

The UA-DETRAC dataset consists of images from urban area roads. The samples in the dataset represent various types of vehicles allowed to operate in the city, including sedans, SUVs, small trucks, and various types of passenger vehicles, but they are lacking in data samples of large vehicles such as trucks, semi-trailers, and tankers. Because of the high similarity between adjacent frames in the dataset, this paper sampled 5605 images using an equidistant sampling method. The dataset was also split into training, testing, and validation sets in a ratio of 7:2:1.

4.2. Evaluation Metrics of Vehicle Detection

To quantitatively analyze the performance of the proposed vehicle detection model, this paper adopts the same evaluation metrics as in the literature [20], which mainly include the following three metrics:

Average Precision (AP). Precision is the proportion of true positive samples predictions among all positive samples predictions, while Recall is the proportion of true positive samples predictions among all actual positive instances. The calculation equation is shown in Equations (6) and (7), respectively:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

where TP represents true positive samples, FP represents false positive samples, and FN represents false negative samples. AP is the area under the Precision–Recall curve, and the higher the AP value, the better the model's accuracy and performance.

Log Average Miss Rate (MR^{-2}) [41]. The vehicle detection index is measured by simultaneously calculating the Miss Rate (MR) and the number of False Position Per Image ($FPPI$). The equation for MR is shown in Equation (8):

$$MR = 1 - TP/N \tag{8}$$

where TP represents true positive samples. By plotting the MR–$FPPI$ curve, the MR^{-2} is the average value of MR calculated for nine $FPPI$ values. The logarithmic intervals of the nine points are averaged samplings within the range of $[10^{-2}, 100]$. MR^{-2} represents the Miss Rate of the vehicle detector at a specified false positive rate, and the lower the value, the better the detection performance.

Jaccard Index (JI). The equation for JI is shown in Equation (9):

$$JI(D, G) = \frac{|IOUMatch(D, G)|}{|D| + |G| - |IOUMatch(D, G)|} \tag{9}$$

where D is a set of detection boxes, and G is a set of ground-truth boxes. JI is more suitable for detection tasks in dense scenes. JI represents the degree of overlap between the predicted box and the ground-truth box, and the higher the value, the better the detection performance.

4.3. Performance Analysis

The PyTorch framework is used in this paper to implement the IBCDet algorithm, and we trained and tested it on an NVIDIA P100 GPU. The experiment uses the training

weights provided by the CrowdDet algorithm as the pre-training weights for the IBCDet algorithm. The proposed model was trained using stochastic gradient descent (SGD) with a momentum value of 0.9, a decay factor of 1e-4, a batch size of 2, and an initial learning rate of 1.25×10^{-3}. The IBCDet algorithm was trained for 30 epochs until convergence.

4.3.1. Experimental Results on NJRY Dataset

This paper presents the results of testing the IBCDet algorithm on the NJRY dataset. Four experiments were designed and compared with current mainstream object detection algorithms, including Faster RCNN [42], SSD [43], YOLOv3 [44], YOLOv5, YOLOv7 [45], YOLOV3 and median filtering [27], and CrowdDet [20].

Firstly, Table 2 shows the performance comparison between our proposed method and the aforementioned object detection algorithms on the NJRY dataset. The results in Table 2 show that the IBCDet algorithm has better performance than the other methods. The IBCDet algorithm achieves an AP of 95.30%, MR^{-2} of 24.44%, and JI of 76.35%, which, respectively, improve the baseline network by 1.3, 19.23, and 2.1 percentage points in terms of AP, MR^{-2}, and JI evaluation metrics as well as achieve the best performance on the NJRY dataset. Additionally, it can be seen from the experimental results that the MR^{-2} improvement effect is most significant, indicating that our algorithm can effectively reduce the Miss Rate of vehicle detection.

Table 2. Comparison of results of different methods on the NJRY dataset. The best results are highlighted in bold.

Method	AP%	MR^{-2}%	JI%
Faster RCNN	75.89	68.00	-
SSD	83.24	62.00	-
YOLOv3	85.13	61.00	-
YOLOv3+median filtering	85.33	57.00	-
YOLOv5	92.31	42.00	-
YOLOv7	93.38	41.00	-
CrowdDet	94.04	30.26	74.76
IBCDet (Ours)	**95.30**	**24.44**	**76.35**

To verify the effectiveness of Involution in the backbone network, the third, fourth, and fifth convolutional layers of ResNet-50 were replaced with Involution, and their performance was evaluated, as shown in Table 3. Since the images have a higher resolution and contain more detailed information in the first two layers of ResNet-50 and the features have less spatial variation, Involution was not used to replace the first two layers. From Table 3, it can be seen that the improvement is the most significant when only the fifth layer convolution is replaced: the AP is improved by 1.1%, the MR^{-2} is reduced by 17.2%, and the JI is improved by 3.3% compared to the CrowdDet algorithm. As Involution dynamically generates different convolution kernels at different positions, it produces different levels of attention, and by using a 7×7 convolution kernel, it obtains more long-range information. Therefore, the Involution operator can achieve better performance than ordinary convolution, and it can improve the detection accuracy of distant vehicles.

To verify the ability of BiFPN to capture contextual semantic information, this study compared the performance of CrowdDet using different feature fusion methods. The experimental results are presented in Table 4. Since BiFPN adds edges to connect contextual information on the basis of FPN and multiplies corresponding weights, it can more effectively acquire contextual semantic information and has better detection performance for vehicles under traffic congestion. As shown in Table 4, the method using BiFPN has better performance than the method using FPN.

To test the impact of Involution and BiFPN used in this article on the overall detection results of the model, two modules were verified through ablation experiments. Table 5 shows the experimental results, which indicate that using both modules simultane-

ously achieves the best results. This demonstrates that the IBCDet algorithm can accurately detect traffic congested vehicles on an expressway.

To visually demonstrate the excellent performance of the proposed method, this paper provides visual comparison results of different methods, as shown in Figure 7. The IBCDet algorithm can achieve satisfactory vehicle detection performance, indicating that the IBCDet algorithm can effectively utilize long-distance interaction information and contextual semantic information for vehicle detection.

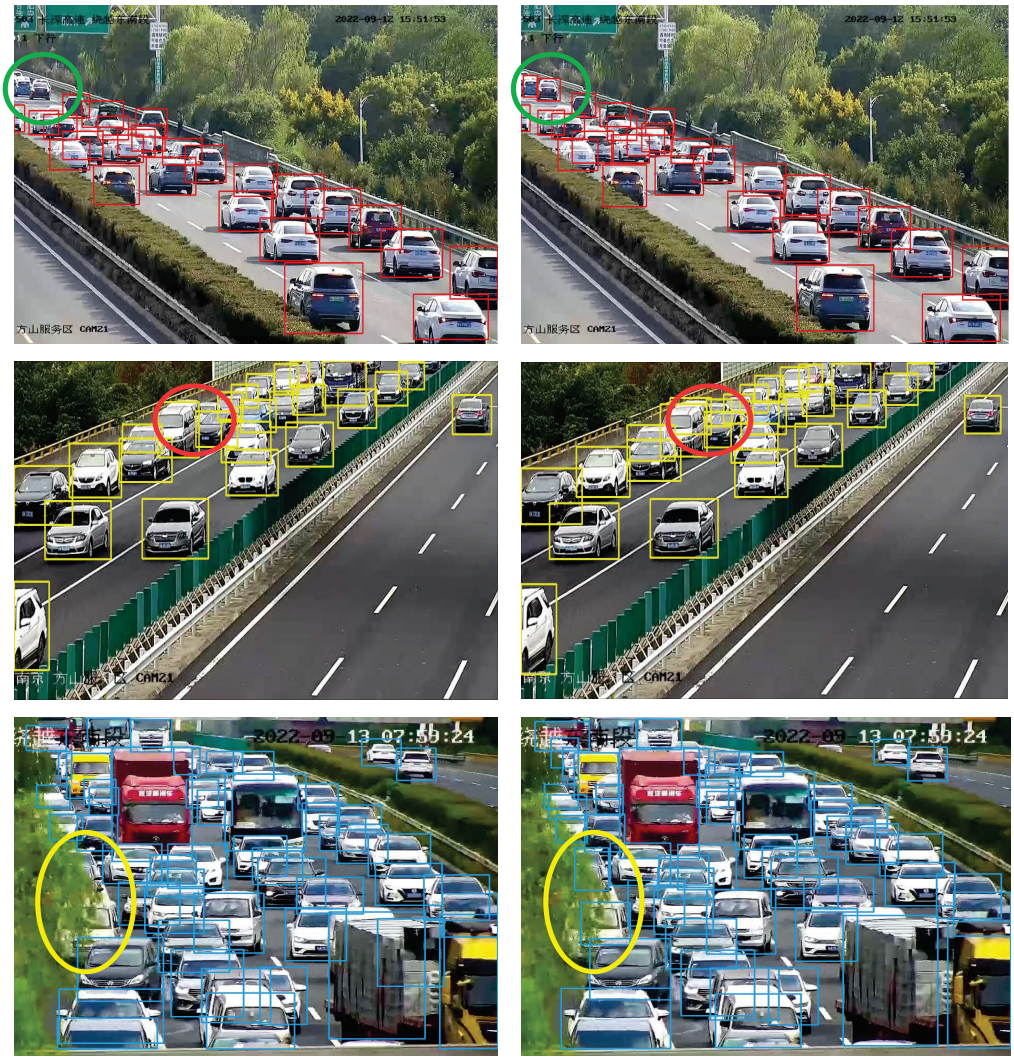


Figure 7. Visual comparison of our method (IBCDet) with baseline. The left is the baseline and the right is ours.

Table 3. Comparison experiment results of replacing convolutional layers at different positions of ResNet-50 with Involution. The best results are highlighted in bold.

Method	AP%	MR^{-2}%	JI%
Baseline	94.04	30.26	74.76
The third layer	93.35	30.16	70.71
The fourth layer	92.56	29.81	70.85
The fifth layer	**95.12**	**25.03**	**77.25**

Table 4. Comparison experiments of different feature fusion methods. The best results are highlighted in bold.

Method	AP%	MR^{-2}%	JI%
ResNet+FPN	94.04	30.26	74.76
ResNet+BiFPN	**95.19**	**27.20**	**75.07**

Table 5. Ablation experiments. The best results are highlighted in bold.

Method	AP%	MR^{-2}%	JI%
Baseline	94.04	30.26	74.76
w/o BiFPN	95.12	25.03	77.25
w/o Involution	95.19	27.20	75.07
IBCDet (Ours)	**95.30**	**24.44**	**76.35**

4.3.2. Experimental Results on the UA-DETRAC Dataset

To validate the performance of the proposed vehicle detection algorithm on the public dataset UA-DETRAC, we compared the IBCDet algorithm with baseline detectors and commonly used object detection algorithms. As shown in Table 6, the IBCDet algorithm achieved the best results on the UA-DETRAC dataset with an AP of 97.49%, MR^{-2} of 14.38%, and JI of 90.43%. Compared to the baseline detector, our method improved the AP, MR^{-2}, and JI by 0.7%, 7%, and 1.9%, respectively, achieving the best performance. This indicates that the IBCDet algorithm not only performs well in detecting vehicles in heavily congested traffic conditions but also has good performance in general vehicle detection scenarios, demonstrating the good robustness of the IBCDet algorithm.

Table 6. Comparison of results between different methods on UA-DETRAC. The best results are highlighted in bold.

Method	AP%	MR^{-2}%	JI%
SSD	89.30	54.00	-
Faster RCNN	90.36	49.00	-
YOLOv5	94.91	34.00	-
YOLOv7	95.69	30.00	-
CrowdDet	96.78	15.47	88.73
IBCDet (Ours)	**97.49**	**14.38**	**90.43**

4.4. Using IBCDet to Implement Traffic Congestion Detection

In traffic congestion detection, obtaining the number of vehicles is crucial. From the above experiments, it can be observed that the proposed IBCDet algorithm in this paper can more accurately detect vehicles under traffic congestion. Additionally, measuring vehicle speed is an essential parameter in traffic congestion detection. To validate the effectiveness of the proposed traffic congestion detection method based on the IBCDet algorithm, a specific segment of the Nanjing Raoyue expressway was selected for verification. This segment is a dual six-lane road with a design speed of 120 km/h. The monitoring video of the downstream section of this road was chosen for experimental validation within a two-hour timeframe. First, to validate the accuracy of speed measurement, this paper combines the YOLOv5 algorithm and the IBCDet algorithm with the DeepSort algorithm for vehicle tracking. The average speed of vehicles at various time intervals in the surveillance video is calculated using Equation (5) proposed in this paper. The accuracy of calculating is evaluated, and the results are presented in Table 7. It can be observed that the IBCDet algorithm achieves the best performance in terms of both vehicle count and vehicle speed accuracy. Further explanation is provided to demonstrate the effectiveness of our proposed traffic congestion detection method compared to other methods.

Table 7. Comparison results of vehicles speed detection accuracy using different methods. The best results are highlighted in bold.

Method	Number of Vehicles		Average Speed of Vehicles
	AP%	MR^{-2}%	V_s Accuracy%
Speed detection algorithm on YOLOv5	92.31	42.00	89.46
Speed detection algorithm on IBCNet(Ours)	**95.30**	**24.44**	**91.28**

Subsequently, the average speed of vehicles at different time points within the surveillance video is calculated by combining the IBCDet algorithm with the DeepSort algorithm, and Figure 8 is plotted accordingly. Based on Table 1, the entire process of vehicle congestion in Figure 8 is divided into five traffic states. The speed distribution and speed confidence intervals of the road segment are analyzed to determine the congestion threshold. Finally, based on the intensity of speed variations at each moment and the obtained speed threshold, the occurrence and dissipation of traffic congestion are determined.

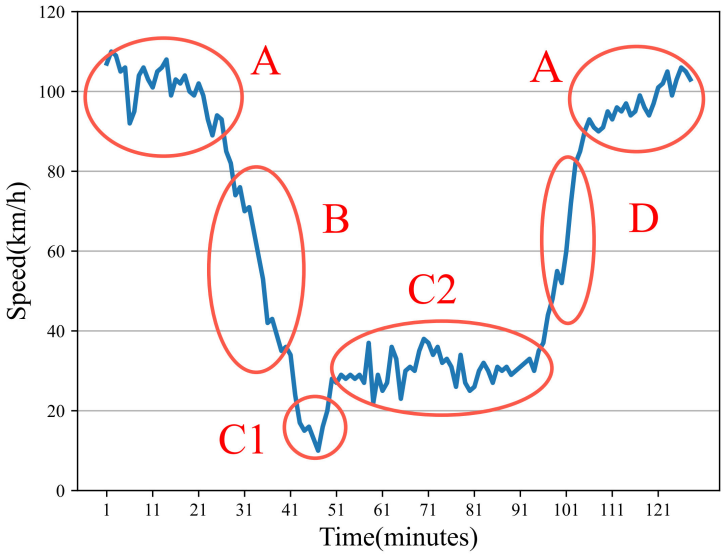


Figure 8. Traffic state diagrams under different speeds.

Among these states, State A represents the stable state, where vehicles can freely pass through with high speeds. State B indicates the congestion formation state, which is characterized by a rapid decrease in speed within a short period. During this state, some vehicles ahead decelerate, causing a backward propagation of the congestion wave. Rear vehicles alternate deceleration, and traffic flow gradually becomes disordered. When the speed falls below a certain threshold, the traffic flow is considered congested. The duration of congestion varies depending on the actual traffic volume. States C1 and C2 represent severe congestion and mild congestion, respectively. In State C1, the threshold is below 23.9 km/h, and the vehicle spacing is extremely small. In State C2, the threshold is below 38.5 km/h, and the vehicle spacing is very small, although the traffic flow can still proceed normally but with reduced mobility. State D represents the congestion dissipation state, where the speed quickly increases, and the vehicle mobility increases, gradually returning to the stable state. From Figure 8, it is evident when traffic congestion occurs and dissipates. This demonstrates that our proposed method using speed thresholds accurately detects traffic congestion.

To further demonstrate the effectiveness of the traffic congestion detection method depicted in Figure 8, we present in Figure 9 the identified congestion scenes from the surveillance videos of the Nanjing Raoyue expressway. Figure 9a shows a severe congestion state characterized by frequent stops and starts. Figure 9b shows a mild congestion state with a lower vehicle occupancy rate compared to the severe congestion state.

From Figure 9, it can be concluded that the congestion detection method for expressway traffic proposed in this article can accurately analyze the congestion status of a road segment. To further verify the accuracy of our congestion detection method, Figure 10 shows the number of vehicles corresponding to each speed detected using the IBCDet algorithm on the specific road segment. It can be seen from Figure 10 that the number of vehicles is inversely proportional to the vehicle speed. When traffic congestion occurs, the vehicle speed on the road segment decreases significantly, while the number of vehicles increases. When the traffic congestion status dissipates, the vehicle speed increases and the number of vehicles decreases. This result verifies the accuracy of vehicle detection and vehicle speed calculation in the traffic congestion algorithm based on IBCDet.



(**a**) (**b**)

Figure 9. Traffic congestion images from Nanjing Raoyue expressway: (**a**) severe congestion; (**b**) mild congestion.

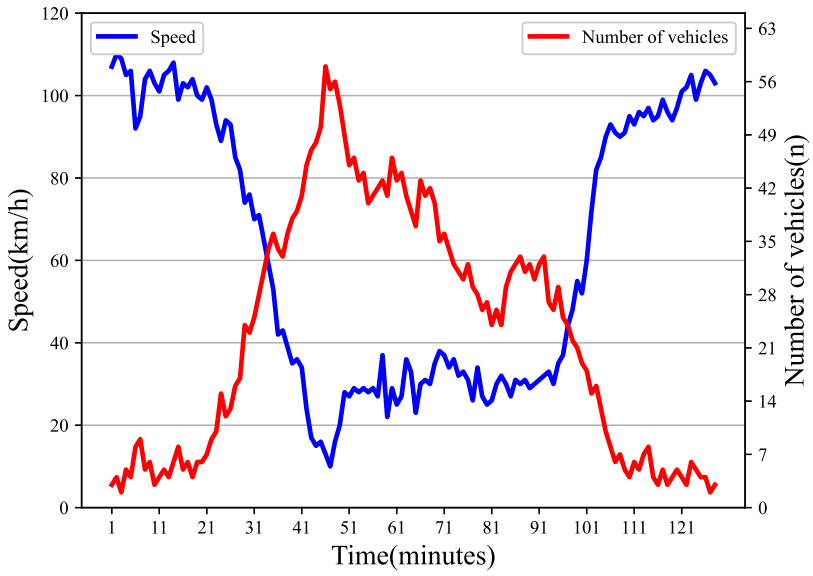


Figure 10. The relationship between speed and vehicle count during the entire duration of congestion.

5. Conclusions

To enhance the vehicle detection accuracy and traffic congestion detection accuracy in expressway surveillance scenarios, this paper proposes an improved vehicle detection algorithm based on CrowdDet, which is called IBCDet. Then, a tracking algorithm based on IBCDet is designed to calculate the running speed of vehicles, and the average running speed is used to achieve traffic congestion detection based on China expressway LoS criteria. The self-built NJRY dataset and the public dataset UA-DETRAC are used to verify the performance of the proposed algorithm. Experimental results demonstrate that our algorithm outperforms commonly used object detection algorithms in terms of both vehicle detection accuracy and traffic congestion detection accuracy. The proposed algorithm provides an effective solution for detecting vehicles and detecting traffic congestion in expressway surveillance scenarios.

The proposed method in this paper can effectively detect traffic congestion on a dataset collected under good lighting conditions. However, its performance is not satisfactory under complex weather conditions and low-light conditions. In the future, we plan to expand our research to detect traffic congestion in different scenarios and under various weather conditions in order to further validate and improve the robustness and adaptability of our algorithm.

Author Contributions: Formal analysis, C.W.; proiect administration, C.W.; software, Y.C.; writing—original draft, Y.C.; writing—review, J.W.; investigation, Y.C. and J.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Ministry of Transport's Industry Key Science and Technology Project (Project No. 2020-ZD3-029), and 2021 Nanjing Municipal Industry and Information Technology Development Special Fund Project (Project name: Construction of 5G-based Application Scenarios for Digital Operation and Control of Intelligent Transportation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are not publicly available due to involve a certain degree of privacy.

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

References

1. Pei, J.; Li, J.; Zhou, B.; Li, Y.; Li, J. A Recommendation Algorithm about Choosing Travel Means for Urban Residents in Intelligent Traffic System. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 26–28 March 2021; pp. 2553–2556.
2. Wang, L. Significance Analysis of Influencing Factors of Highway Freight Transportation in China and Multi-Variable Grey Prediction for Its Development. *Intell. Fuzzy Syst.* **2021**, *41*, 1237–1246. [CrossRef]
3. Enescu, F.M.; Bizon, N.; Serban, G.; Vatuiu, T.-B.; Istrate, D.-C. Environmental Protection-Blockchain Solutions for Intelligent Passenger Transportation of Persons. In Proceedings of the 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 24–26 June 2021; pp. 1–6.
4. Zhang, C.; Shen, S.; Huang, H.; Fang, S.; Zhao, H.; Zhao, X. Estimation of the Vehicle Speed Using Cross-Correlation Algorithms and MEMS Wireless Sensors. *Sensors* **2021**, *21*, 1721. [CrossRef] [PubMed]
5. Zhang, L.; Lu, Y. Distributed Consensus-Based Boundary Observers for Freeway Traffic Estimation with Sensor Networks. In Proceedings of the American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 4497–4502.
6. Tampubolon, H.; Yang, C.L.; Chan, A.S.; Ciptadi, A. Optimized CapsNet for Traffic Jam Speed Prediction Using Mobile Sensor Data under Urban Swarming Transportation. *Sensors* **2019**, *19*, 5277. [CrossRef] [PubMed]
7. Qureshi, K.N.; Abdullah, A.H.; Altameem, A. Road Aware Geographical Routing Protocol Coupled with Distance, Direction and Traffic Density Metrics for Urban Vehicular Ad Hoc Networks. *Wireless Pers. Commun.* **2017**, *92*, 1251–1270. [CrossRef]
8. Oumaima, E.J.; Jalel, B.O.; Véronique, V. A Stochastic Traffic Model for Congestion Detection in Multi-lane Highways. In Proceedings of the Ad Hoc Networks: 12th EAI International Conference (ADHOCNETS 2020), Paris, France, 17 November 2020; Springer International Publishing: Cham, Switzerland, 2021; pp. 87–99.
9. Lam, C.T.; Gao, H.; Ng, B. A Real-Time Traffic Congestion Detection System Using On-Line Images. In Proceedings of the IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, China, 27–30 October 2017; pp. 1548–1552.

10. Tahmid, T.; Hossain, E. Density Based Smart Traffic Control System Using Canny Edge Detection Algorithm for Congregating Traffic Information. In Proceedings of the 3rd International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 28–30 December 2017; pp. 1–5.
11. He, F.; Yan, X.; Liu, Y.; Han, L.; Li, G. A Traffic Congestion Assessment Method for Urban Road Networks Based on Speed Performance Index. *Procedia Eng.* **2016**, *137*, 425–433. [CrossRef]
12. Jin, Y.; Hao, W.; Wang, P.; Wang, Z. Fast Detection of Traffic Congestion from Ultra-Low Frame Rate Image Based on Semantic Segmentation. In Proceedings of the 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), Xi'an, China, 19–21 June 2019; pp. 528–532.
13. Willis, C.; Harborne, D.; Tomsett, R.; Guo, Y. A Deep Convolutional Network for Traffic Congestion Classification. In Proceedings of the NATO IST-158/RSM-010 Specialists' Meeting on Content Based Real-Time Analytics of Multi-Media Streams, Bucharest, Romania, 16–18 May 2017; pp. 1–11.
14. García-García, L.; Jimenez, J.M.; Taha, M.; Lloret, J. Wireless Technologies for IoT in Smart Cities. *Netw. Protoc. Algorithms* **2018**, *10*, 23–64. [CrossRef]
15. Rehman, G.; Ghani, A.; Zubair, M.; Ghayyure, S.A.K.; Muhammad, S. Honesty based democratic scheme to improve community cooperation for Internet of Things based vehicular delay tolerant networks. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4191. [CrossRef]
16. Rehman, G.; Ghani, A.; Zubair, M.; Saeed, M.I.; Singh, D. SOS: Socially Omitting Selfishness in IoT for Smart and Connected Communities. *Int. J. Commun. Syst.* **2020**, *36*, e4455. [CrossRef]
17. Rasheed, A.; Gillani, S.; Ajmal, S.; Ahmad, I. Vehicular Ad Hoc Network (VANET): A Survey, Challenges, and Applications. In Proceedings of the Vehicular Ad-Hoc Networks for Smart Cities: Second International Workshop, Lahore, Pakistan, 7–8 August 2016; Springer: Singapore, 2017; pp. 39–51.
18. Wang, H.; Quan, W.; Ochieng, W.Y. Smart Road Stud Based Two-Lane Traffic Surveillance. *J. Intell. Transp. Syst.* **2020**, *24*, 480–493. [CrossRef]
19. Ding, D.; Tong, J.; Kong, L. A Deep Learning Approach for Quality Enhancement of Surveillance Video. *J. Intell. Transp. Syst.* **2020**, *24*, 304–314. [CrossRef]
20. Chu, X.; Zheng, A.; Zhang, X.; Ma, J.; Qi, H. Detection in Crowded Scenes: One Proposal, Multiple Predictions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 13–19 June 2020; pp. 12214–12223.
21. Li, D.; Hu, J.; Wang, C.; Shi, B.; Zhuang, Y.; Lin, Y. Involution: Inverting the Inherence of Convolution for Visual Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 12321–12330.
22. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
23. Gao, Y.; Li, J.; Xu, Z.; Zhao, X.; Cheng, Y.; Wang, Y.; Wang, X.; Zhang, B. A Novel Image-Based Convolutional Neural Network Approach for Traffic Congestion Estimation. *Expert Syst. Appl.* **2021**, *180*, 115037. [CrossRef]
24. Kurniawan, J.; Syahra, S.G.S.; Dewa, C.K. Traffic Congestion Detection: Learning from CCTV Monitoring Images Using Convolutional Neural Network. *Procedia Comput. Sci.* **2018**, *2672*, 222–231. [CrossRef]
25. Chakraborty, P.; Adu-Gyamfi, Y.O.; Poddar, S.; Ahsani, V.; Sharma, A.; Sarkar, S. Traffic Congestion Detection from Camera Images Using Deep Convolution Neural Networks. *Transp. Res. Rec.* **2018**, *144*, 291–297. [CrossRef]
26. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
27. Lam, C.-T.; Ng, B.; Chan, C.-W. Real-Time Traffic Status Detection from On-Line Images Using Generic Object Detection System with Deep Learning. In Proceedings of the 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 16–19 October 2019; pp. 1506–1510.
28. Liu, B.; Lam, C.T.; Ng, B.K. Improved Real-Time Traffic Congestion Detection with Automatic Image Cropping using Online Camera Images. In Proceedings of the 2021 IEEE 21st International Conference on Communication Technology (ICCT), Xi'an, China, 28–31 October 2021; pp. 1117–1122.
29. Zambrano-Martinez, J.L.; Calafate, C.T.; Soler, D.; Cano, J.C.; Manzoni, P. Modeling and Characterization of Traffic Flows in Urban Environments. *Sensors* **2018**, *18*, 2020. [CrossRef]
30. Costa, C.; Chatzimilioudis, G.; Zeinalipour-Yazti, D.; Mokbel, M.F. Towards Real-Time Road Traffic Analytics using Telco Big Data. In Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics, Munich, Germany, 28 August 2017; pp. 1–5.
31. Cui, H.; Yuan, G.; Liu, N.; Jiang, F.; Wu, J.; Zhang, J. Convolutional Neural Network for Recognizing Highway Traffic Congestion. *J. Intell. Transp. Syst.* **2020**, *24*, 279–289. [CrossRef]
32. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
33. Zhang, S.; Benenson, R.; Schiele, B. CityPersons: A Diverse Dataset for Pedestrian Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3213–3221.
34. Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; Sun, J. CrowdHuman: A Benchmark for Detecting Human in a Crowd. *arXiv* **2018**, arXiv:1805.00123.

35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
37. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
38. Kerner, B.S. Three-Phase Traffic Theory and Highway Capacity. *Phys. A Stat. Mech. Appl.* **2004**, *333*, 379–440. [CrossRef]
39. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 3645–3649.
40. China Committee of Highway Engineering Standardization. *Technology Standard of Highway Engineering*; China Communications Press Co., Ltd.: Beijing, China, 2014.
41. Dollar, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 743–761. [CrossRef] [PubMed]
42. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
43. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer International Publishing: Cham, Switzerland, 2016; Part I; pp. 21–37.
44. Farhadi, A.; Redmon, J. Yolov3: An Incremental Improvement. In *Computer Vision and Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 1804, pp. 1–6.
45. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.