ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20250120002283
Original scientific paper

Automatic Detection Method for Daytime Traffic Flow Based on Background Difference and Edge Extraction

Ke DAI

Abstract: Real-time monitoring and analysis of traffic flow have become critical as urban congestion intensifies. To address the current challenges of low accuracy and insufficient detection efficiency in daytime traffic monitoring, this paper introduces an advanced method utilizing background difference and edge extraction techniques. Specifically, it employs the Gaussian Mixture Model (GMM) for dynamic background modeling and Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance input images. Furthermore, the Faster Regional Convolutional Neural Network (Faster R-CNN) and an optimized Canny edge detection algorithm are utilized for effective target detection and tracking. Experimental evaluations indicate significant improvements over existing benchmarks, achieving accuracy, recall, and F1 scores of 0.99, 0.97, and 0.98, respectively. Practical application tests demonstrate an average vehicle detection accuracy of 99.49%, with robustness assessments confirming the model's reliability under diverse datasets and environmental conditions. This research provides substantial improvements in automatic traffic flow detection, supporting more efficient urban traffic management.

Keywords: background difference method; Canny; edge extraction; intelligent transportation; traffic flow detection

1 INTRODUCTION

The issue of daytime traffic congestion in cities has grown in importance as urbanization has accelerated, resulting in a number of issues such as energy waste, environmental pollution, and decreased travel efficiency [1]. Relatively speaking, the traffic flow in daytime environment is more complicated compared to nighttime. Traffic congestion in daytime environment originates from the contradiction between transportation demand and supply. In the daytime environment, urban traffic shifts from peak hour congestion to all-weather, normalized congestion. There were 430 million motor vehicles in China as of September 2023, with over a million automobiles in 90 cities [2]. Traffic flow data analysis is a technique of traffic engineering. Traffic data is collected, organized and parsed to better understand and optimize traffic operations [3-4]. These data include the number of vehicles, the speed at which they pass through a particular roadway, and the duration and extent of traffic congestion. By analyzing the traffic flow data, the pattern of daytime traffic operation can be better understood, which leads to more effective traffic planning [5]. However, there are still problems with poor adaptability to lighting and low edge detection accuracy in current traffic flow analysis and statistics [6]. Therefore, to enhance the detection accuracy and efficiency of daytime traffic flow, the study proposes an automatic daytime traffic flow detection (TFD) method based on background difference and edge extraction. This method solves the problem of image quality degradation caused by uneven lighting by using Contrast Limited Adaptive Histogram Equalization (CLAHE) with limited contrast, using Gaussian Mixture Model (GMM) to dynamically adapt to scene changes, and automatically selecting thresholds through an improved Canny edge detection operator (Canny) to preserve complete edge features. The innovation of the research lies in the combination of CLAHE, GMM, and improved Canny algorithm, which significantly improves the accuracy and robustness of automatic detection of daytime TFD through multi-level image processing and dynamic background adaptation.

2 LITERATURE REVIEW

Numerous academics both domestically and internationally have carried out a number of studies on the target detection (TD) and counting problem of automobiles. Using multi-scale deep convolutional neural networks, Hassaballah et al. presented a reliable vehicle detection and tracking technique for the severe weather vehicle detection challenge. Under extreme weather circumstances, the system beats the msost sophisticated vehicle recognition and tracking techniques currently in use, as confirmed by the experimental results [7]. However, this method can produce ghosting residues lasting over 15% during sudden changes in lighting such as cloud movement. The method proposed by the research institute integrates CLAHE preprocessing and GMM parameter dynamic adjustment mechanism, and enhances stable background features by limiting contrast, which is beneficial for reducing ghosting rate.

A lightweight deep learning (DL) approach based on infrared weak vehicle TD and recognition was created by Renhao et al. for the vehicle TD problem. A lightweight adaptation of the you only look once version 5 (YOLOv5) algorithm served as the foundation for the scheme, which enhanced real-time performance while reducing the model's structure. The experimental findings proved the method's viability and superiority in vehicle TD [8]. However, this method has a low recall rate for small targets such as motorcycles. The study used an improved Canny operator to enhance edge features, thereby improving the accuracy of small object detection.

Ahmed et al. used the sea lion optimized VRC-Sloder model with DL to create a new vehicle detection and classification system for the problem of automatic vehicle counting from aerial remote sensing photos in complex urban situations. The suggested model's robust vehicle detection and classification capacity was displayed by the experimental findings [9]. However, this The method has

issues with complexity and high computational requirements. The improved Canny edge detection operator and Fast Regional Convolutional Neural Network (Fast R-CNN) not only ensure the accuracy requirements, but also meet the real-time requirements.

Based on this, the study expands on past studies by employing a novel strategy that combines CLAHE, GMM, and improved Canny operator (ICO) to increase the accuracy, efficiency, and resilience of daytime TFD. This increases the model's robustness in the face of changing backgrounds and varied lighting circumstances, in addition to improving vehicle detection accuracy. In contrast, despite the local advantages of other methods under specific conditions, the research method is more reliable and practical in real-world environments, thus providing a more robust support for urban traffic management.

3 RESEARCH METHOD

To improve the accuracy, efficiency, and robustness of daytime TFD, a method for automatic daytime TFD based on background difference and edge extraction is proposed. This study enhances the input image using CLAHE and models the background using GMM. It also utilizes the ICO for edge detection, and finally applies Faster R-CNN and simple online and real-time tracking (SORT) to achieve object detection (OD) and object tracking (OT).

3.1 GMM-Based Background Model Approach

To detect the daytime traffic flow, TD of vehicles is required first. The study suggests a backdrop difference and edge extraction-based vehicle TD technique to solve the issue. The method first enhances the image by CLAHE algorithm for subsequent vehicle TD. Subsequently, the background model (BM) is established by GMM, and edge detection is performed on the image by the ICO. Finally, the vehicle detection is realized by the TD algorithm. The flowchart of vehicle TD proposed in the study is shown in Fig. 1.

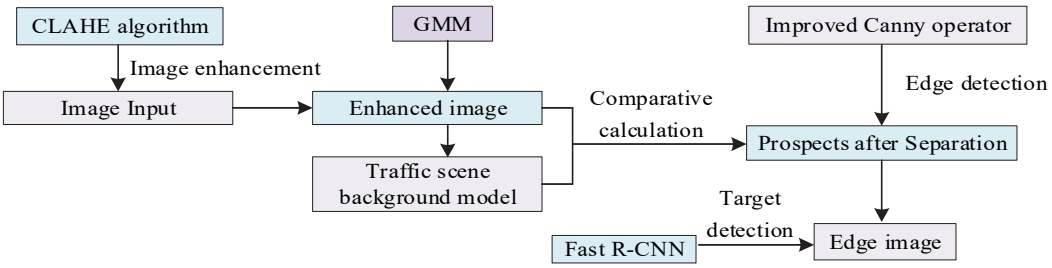


Figure 1 Flowchart of vehicle target detection

In Fig. 1, the vehicle TD uses the CLAHE algorithm to enhance the image's contrast and features, making the vehicle stand out more. The BM of the traffic scene is created by GMM for subsequent separation of foreground from the video frames. To distinguish the foreground from the video's CF, the foreground mask is calculated by comparing the current frame (CF) and the BM. The difference is then calculated by comparing the CF and the BM. To obtain edge features of foreground targets for later shape analysis and target recognition, an ICO is utilized for edge detection. During the daytime, vehicles in the image may appear blurred or have missing details due to lighting variations, weather conditions, shadows, etc. Therefore, the CLAHE algorithm is used in the study to improve the image's contrast and details. To reduce contrast and weaken the noise amplification problem, the CLAHE algorithm is based on the adaptive histogram equalization technique and incorporates a threshold [10, 11]. The CLAHE algorithm flowchart is shown in Fig. 2.

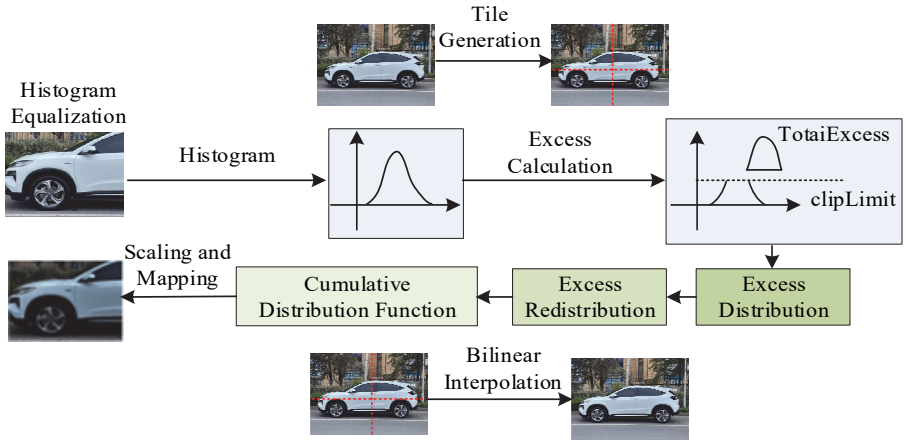


Figure 2 Flowchart of CLAHE algorithm

In Fig. 2, the CLAHE algorithm process is to input the image to be processed and divide the input image into multiple small image blocks, and perform histogram equalization independently for each image block to achieve

local contrast enhancement. When calculating the histogram of each image block, in an attempt to obtain a smoother histogram distribution, oversampling is performed for each gray level. The oversampled histogram is cropped to ensure that the number of pixels in each grey level does not exceed a set threshold to avoid over-enhancement of the image. Cumulative distribution function (CDF) is calculated for the histogram of each image block. By calculating the CDF, the new grey value corresponding to each grey value can be obtained, thus equalizing the histogram [12, 13]. Based on the computed CDF, each image block is processed for scaling and mapping. Redistribution process is performed on the processed image blocks to eliminate the boundary effect between blocks. After the above processing steps, the final enhanced image is obtained. Based on the enhanced image obtained, it is studied to separate the moving vehicles from the image by background difference method. One of the most influential factors on the background difference results is the selection of the BM. To adapt to the lighting changes and dynamic interference in daytime traffic scenes, GMM was used to construct the background [14. 15]. The core is to simulate the color value distribution of each pixel point through a linear combination of 3-5 Gaussian distributions, as shown in Eq. (1).

$$P(X_t)=\sum_{k=1}^{K} w_k \cdot \eta(X_t, \mu_k, \Sigma_k) \tag{1}$$

In Eq. (1), K is the number of Gaussian distributions, w_k is the weight of the k-th distribution, μ_k and $\sum_k$ are the mean and covariance matrices, respectively. Calculate the Mahalanobis distance between the current frame pixel and all Gaussian distributions. If $|X_t-\mu_k|<2.5\sqrt{\sum_k}$ is satisfied, it is judged as a match. The weight parameters of the GMM background model are updated as shown in Eq. (2).

$$\prod^{(l)}=(1-a)\prod^{(l-1)}+a \tag{2}$$

In Eq. (2), a denotes the learning rate. The mean value is updated as shown in Eq. (3).

$$M^{(l)}=(1-z)M^{(l-1)}+zX_l \tag{3}$$

In Eq. (3), X_l denotes the first pixel value of the image. Eq. (4) displays the computational formula for z.

$$z=ap \tag{4}$$

The covariance is updated as shown in Eq. (5).

$$\Sigma=(1-z)\Sigma^{(l-1)}+z\left(X^{(l)}-M^{(l)}\right)^2 \tag{5}$$

The background modeling process using GMM is shown in Fig. 3.

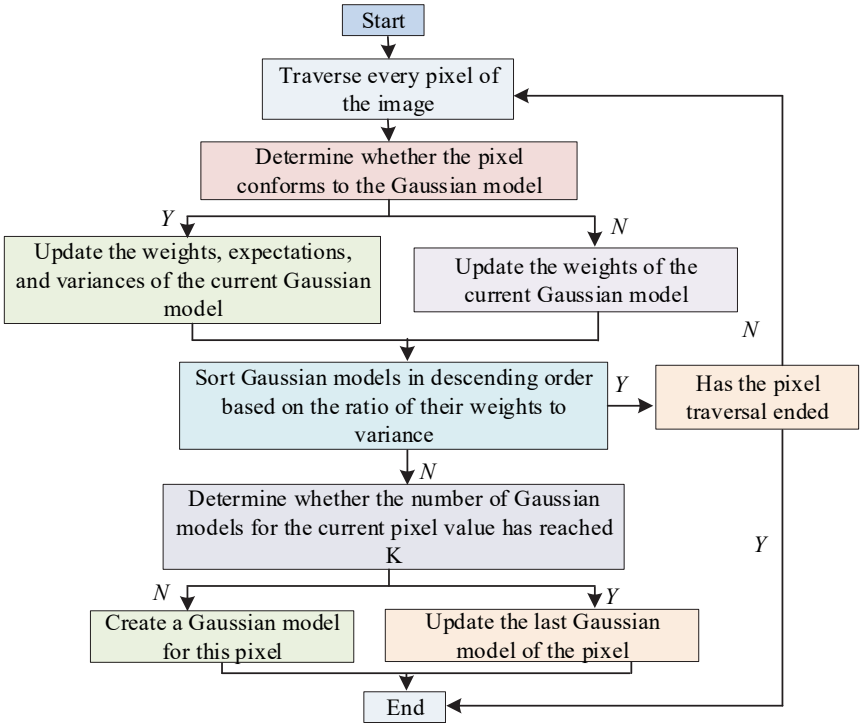


Fig. 3 Flowchart of GMM-based background model

In Fig. 3, each new frame pixel is judged by the established GMM BM. The probability of that pixel value under each GD is calculated. If a pixel value is very close to the mean value of the BM, the pixel can be considered to belong to the background. If it does not, it is recognized as foreground.

3.2 Edge Detection Algorithm Based on ICO

To improve the effectiveness of TFD in daytime environments, edge detection is also required after BM to make subsequent TD and classification more effective. The

edge detection method used in the study is the ICO. Canny operator is a multistage edge detection algorithm designed to provide an optimal edge detection algorithm to ensure that the detection of edges is as accurate and efficient as possible [16, 17]. The main steps of Canny operator are shown in Eq. (6).

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{6}$$

In Eq. (6), σ denotes the standard deviation of the Gaussian function. $G(x, y)$ is the Gaussian value at coordinate (x, y). Eq. (7) illustrates how to use the gradient operator to determine the gradient magnitude and direction of each pixel in the image.

$$G(x,y)=\sqrt{G_x^2+G_y^2} \tag{7}$$

In Eq. (7), G_x and G_y denote the image gradient in the x and y directions, respectively. The gradient magnitude is refined to suppress edge responses that are not local maxima. Two thresholds are used to identify strong edges, weak edges, and non-edges. Pixels with a gradient magnitude higher than the high threshold are regarded as edges in this context. Pixels with a gradient magnitude between the low and high thresholds could be considered edges. It is regarded as an edge if it is attached to a strong edge. If not, it is repressed. Pixels are not regarded as edges if their gradient magnitude is less than the low threshold [18, 19]. To address the problems of sensitive parameter selection and computational complexity of the traditional Canny operator, the study has improved it. Fig. 4 displays the revised method's schematic diagram.

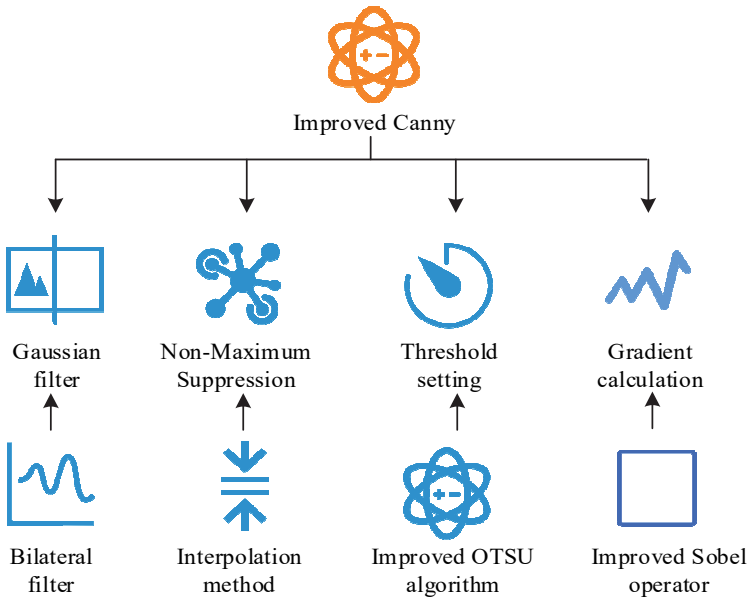


Figure 4 Flowchart of the ICO

In Fig. 4, the ICO utilizes bilateral filtering instead of Gaussian filtering, which causes blurring of edge information during denoising. Bilateral filtering is able to maintain the edge information while smoothing the image. This improvement enhances the clarity of the edges while removing the noise, making the gradient calculation more accurate. The improved Sobel operator is utilized for obtaining more accurate gradient magnitude maxima. Points on both sides of the gradient direction are obtained by interpolation for more accurate edge localization to enhance detection of fine edges or edges with small slopes. The improvement in gradient computation is shown in Eq. (8).

$$G_0=\begin{bmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{bmatrix}\cdot k,\ G_{90}=\begin{bmatrix}-1&-2&-1\\0&0&0\\1&2&1\end{bmatrix}\cdot k \tag{8}$$

Eq. (8) represents the calculation of partial derivatives of gradient magnitude over 0° and 90°. Among them, k denotes the input image. The partial derivatives on 45° and 135° are calculated as shown in Eq. (9).

$$G_{45}=\begin{bmatrix}0&1&2\\-1&0&1\\-2&-1&0\end{bmatrix}\cdot k,\ G_{135}=\begin{bmatrix}-1&-2&0\\-1&0&1\\0&2&1\end{bmatrix}\cdot k \tag{9}$$

The gradient in k is calculated as shown in Eq. (10).

$$G=\sqrt{G_0^2+G_{45}^2+G_{90}^2+G_{135}^2} \tag{10}$$

The gradient direction for each pixel in k is calculated as shown in Eq. (11).

$$\begin{cases}G_x=G_0+\frac{\sqrt{2}}{2}G_{45}+\frac{\sqrt{2}}{2}G_{135}\\G_y=G_{90}+\frac{\sqrt{2}}{2}G_{45}+\frac{\sqrt{2}}{2}G_{135}\end{cases} \tag{11}$$

The gradient direction is shown in Eq. (12).

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{12}$$

For non-maximal value suppression, the interpolation method is utilized to obtain points on both sides of the gradient direction as shown in Eq. (13).

$$\begin{cases} v_1 = w \times G[i+1, j-1] + (1-w) \times G[i, j-1] \\ v_2 = w \times G[i-1, j+1] + (1-w) \times G[i, j+1] \end{cases} \tag{13}$$

In Eq. (13), v_1 and v_2 denote the gradient magnitude interpolation results in the x-axis direction when $|G_x| \cdot |G_y| > 0$, respectively. w is the weight. The result of $|G_x| \cdot |G_y| < 0$ in the x-axis direction is the same as the interpolation result in the y-axis. The generation of pseudo edge points can be effectively avoided by comparing the gradient magnitude at the center point with the calculated magnitude in the adjacent gradient direction. To address the problem of the traditional Canny operator threshold selection is more difficult, the study uses the improved maximum inter-class variance method (Nobuyuki Otsu, OTSU) algorithm to optimize it. Specifically, the average gray level of the interclass variance is replaced by using the variance information, as shown in Eq. (14).

$$\varphi_1^2 = (\mu_0 - \mu)^2, \ \varphi_2^2 = (\mu_1 - \mu)^2 \tag{14}$$

In Eq. (14), $\varphi_0{}^2$, $\varphi_1{}^2$, $\varphi_2{}^2$ are denoted as the interclass variance of the global, foreground, and background regions, respectively. φ^2 is the interclass variance for foreground and background. μ denotes the mean gray value. The foreground and background region gray values are μ_0 and μ_1, respectively. $\varphi_0{}^2$ is calculated as shown in Eq. (15).

$$\varphi_0^2 = \sum_{i=0}^{255} P_i \cdot (i-\mu)^2 \tag{15}$$

In Eq. (15), P_i denotes the probability of occurrence of a pixel point. The φ^2 calculation formula is shown in Eq. (16).

$$\varphi^2 = \omega_0 \cdot \left(\varphi_1^2 - \varphi_0^2\right) + \omega_1 \cdot \left(\varphi_2^2 - \varphi_0^2\right) \tag{16}$$

The optimal threshold is shown in Eq. (17).

$$\overline{P} = \sum_{-o}^{o} P_{(T-o)} \tag{17}$$

In Eq. (17), $\overline{P}$ denotes the optimal threshold. o denotes the possible threshold range. $P_{(T-o)}$ denotes the pixel probability at gray level $T-o$. The threshold that best distinguishes the foreground from the background is chosen by adding up all of the potential gray values from the range of gray values $T-o$. Detection of target vehicles is performed by the widely used Faster R-CNN.

3.3 Target Tracking Algorithm Based on SORT Algorithm

The TD of vehicles can be realized by the above method, in an effort to realize the traffic flow statistics in the daytime environment, target vehicle tracking is also needed on this basis. The study realizes the tracking of vehicles in consecutive multi-frame images through the target tracking algorithm. If the same vehicle is tracked in consecutive multi-frame images, the vehicle count is increased correspondingly. The SORT algorithm is a simple, online, real-time multi-target tracking algorithm that employs Kalman filter (KF) and Hungarian algorithms to deal with the two components of motion prediction and data association, respectively [20, 21]. Fig. 5 displays the SORT algorithm's flowchart.

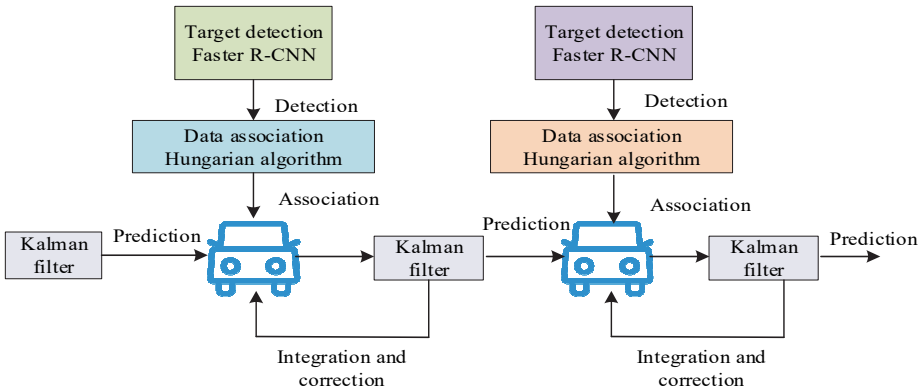


Figure 5 Flowchart of SORT algorithm

In Fig. 5, the SORT method initially uses the Faster R-CNN to jointly detect the target object's position and bounding box in each frame. The KF then models and forecasts the target's motion. The KF estimates the target's position and velocity in the subsequent frame by using the target's motion model and current state [22, 23]. For determining the association of targets between the CF and earlier frames, the SORT algorithm presents a data association technique based on the Hungarian algorithm. By minimizing the association cost, the approach determines the identification and trajectory of the target by matching the tracked target in the previous frame with the

target in the CF [24, 25]. Based on the above theory, the study constructs an automatic daytime TFD model. Fig. 6 displays the model's structure.

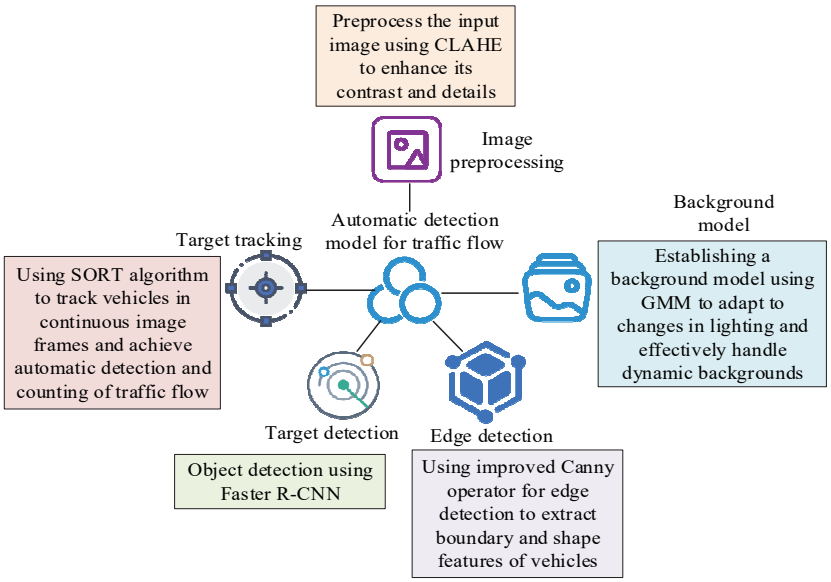


Figure 6 Structure of automatic daytime TFD model

In Fig. 6, the structure of the automatic TFD model includes five main modules. In an effort to prevent blurring or a lack of details in the image caused by variations in daylight illumination, weather, and other factors, CLAHE preprocesses the input image to improve contrast and details. The background is modeled using GMM to adapt to lighting changes and handle dynamic background effectively. To identify the moving vehicles, the background difference approach extracts the foreground mask. Edge detection using ICO to extract boundary and shape features of vehicles. TD is performed by Faster R-CNN and the SORT algorithm is used to realize the tracking of vehicles in consecutive image frames, so as to complete the automatic detection and counting of traffic flow. The proposed process for automatic detection of daytime traffic flow is shown in Fig. 7.

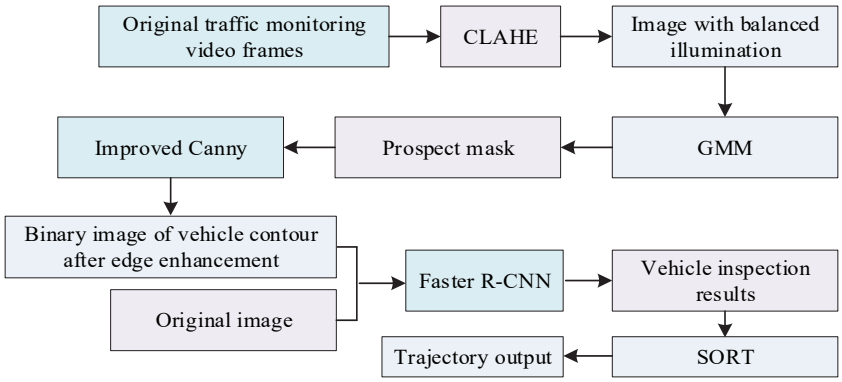


Figure 7 Flow chart of automatic detection of daytime traffic flow

In Fig. 7, the automatic detection process of daytime traffic flow is as follows: Firstly, adaptive lighting enhancement is applied to the original video frames through CLAHE to solve the problem of insufficient contrast caused by backlighting/shadows. Subsequently, using GMM background modeling to separate moving vehicles, significantly reducing ghosting interference by dynamically updating Gaussian distribution parameters. By improving the Canny operator for fine edge extraction of foreground regions, combined with directional gradient calculation and adaptive thresholding, the detection rate of small targets is improved. Faster R-CNN achieves precise vehicle positioning based on enhanced multi-channel input, and finally completes cross frame trajectory association through SORT algorithm.

4 RESULTS AND DISCUSSION

The foreground mask is extracted using the background difference technique in an attempt to identify the moving cars. The study is compared with other advanced models on Cityscapes dataset. The experimental environment is set up with AMD Ryzen 7 5800X CPU, NVIDIA GeForce RTX 3060 GPU, 16 GB RAM, Ubuntu 20.04 LTS operating system, PyTorch 1.x DL framework, and Python 3.7 programming language. The gathered dataset is split 8:2 between training and validation sets. The proposed model is subjected to ablation experiments to verify the importance of each module, which is evaluated using multiple OT accuracy (MOTA), identity F1 score (IDF1 score), frames per second (FPS) and other metrics.

Tab. 1 displays the ablation experiment's findings.

Table 1 Ablation experiments

/	Precision	Recall	F1 score	MOTA	IDF1 score	FPS
Benchmark model	0.85	0.78	0.81	0.74	0.75	10
Benchmark model + CLAHE	0.89	0.86	0.87	0.78	0.82	9.8
Benchmark model + CLAHE + GMM	0.94	0.92	0.93	0.8	0.86	9.5
Benchmark model + CLAHE + GMM + ICanny	0.99	0.97	0.98	0.85	0.91	9.2

In Tab. 1, the benchmark model is a detection model that only uses OD algorithms and OT algorithms. With the enhancement of the model, accuracy continues to improve, indicating that most of the detected vehicles are correct, with a maximum accuracy of 0.99. The model's improvement raises the recall from 0.78 to 0.97. The F1 score (F1) also raises from 0.81 to 0.98 as the model strengthened. Both MOTA and IDF1 scores show a gradually increasing trend. However, in the FPS evaluation results, as the model complexity increases, the FPS of the model gradually decreases, but still remains at a relatively high level. The outcomes demonstrate that the model's improvement greatly boosts OD and OT performance, particularly in important metrics like accuracy, recall, and F1. The proposed model can effectively achieve the task of detecting daytime traffic flow. To verify the impact of bilateral filtering and the improved Sobel operator on performance, experiments were conducted on the improved Canny operator, and the results are shown in Tab. 2.

Table 2 Optimization strategy and performance impact of improving Canny operator

Improving dimensions	Optimization plan	Comparison of Performance Metrics (Traditional Canny Operator/Improved Solution)	Lifting effect
Filtering method	Bilateral filter	Edge continuity: 72.3/91.7%	19.4%
Gradient calculation	Gradient direction + Gradient interpolation	Edge recall rate: 68.4/89.2%	20.8%
Dynamic threshold	Variance weighted OTSU adaptive adjustment	F1 under strong light: 0.71/0.93	0.22

According to Tab. 2, in terms of filtering methods, bilateral filtering replaces traditional Gaussian filtering and improves edge continuity by 19.4%. Gradient calculation improves edge recall by 20.8% by increasing detection direction and optimizing interpolation. The dynamic threshold adopts an adaptive algorithm, and the F1 score under strong light is increased by 0.22. The table quantifies data to demonstrate how the improved Canny operator can effectively address issues such as edge blurring, missed detections, and light sensitivity in traditional methods. It also validates the practical value of the improved Canny operator in real-time traffic detection systems. To verify the performance of the models for TFD, advanced models of the same type are used for comparison. The comparison models are TFD model based on YOLOv5 and KF (YOLOv5-KF), universal OT based on YOLOv8 and for regression networks (YOLOv8-GOTURN). Fig. 8 displays the comparison test results.

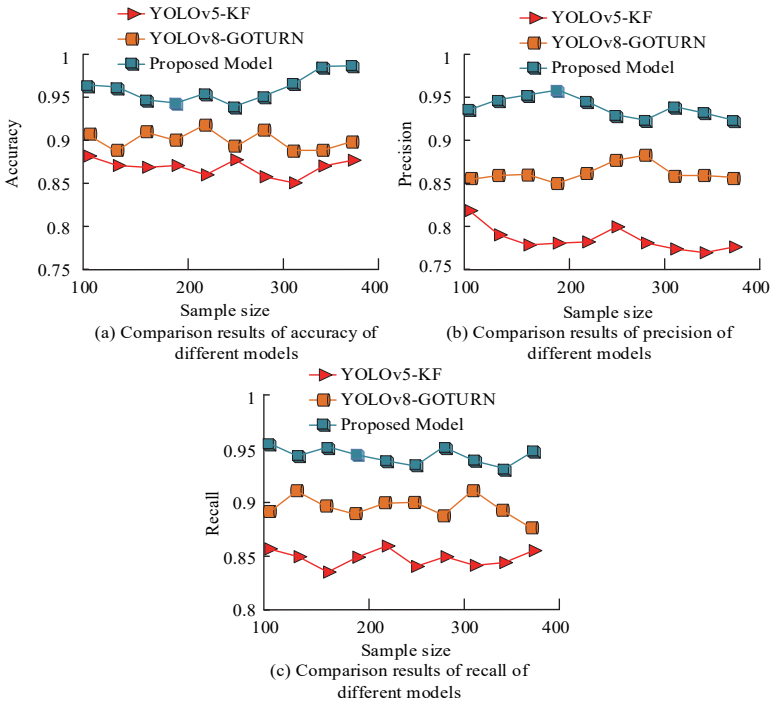
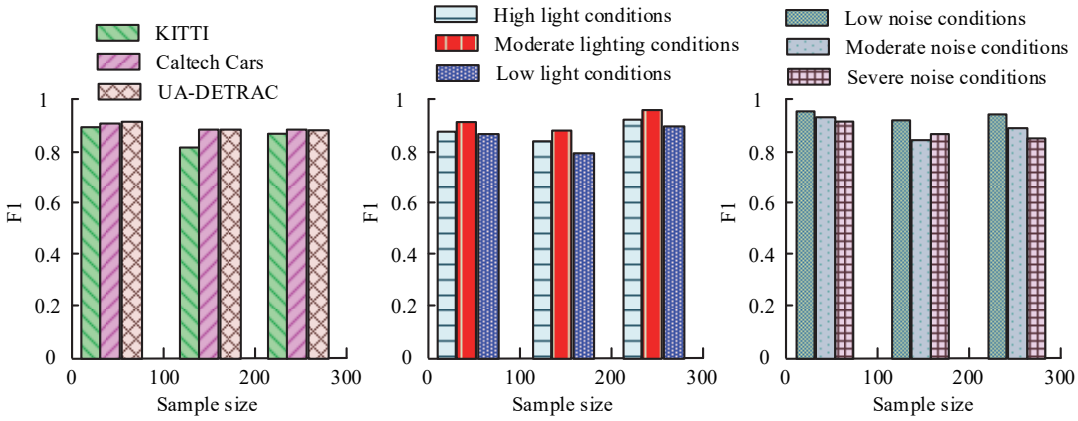


Figure 8 Evaluation results of proposed model accuracy, recall, and F1

In Fig. 8a, in the accuracy evaluation, the accuracy of each model is 0.88, 0.92, and 0.96. In Fig. 8b, in the recall evaluation, the recall values for each model are 0.82, 0.86, and 0.94. In Fig. 8c, in the F1 evaluation, the F1s of each model are 0.85, 0.89, and 0.95. The outcome displays that the proposed model excels in misdetection of vehicles. Meanwhile, the statistics reveal that vehicles of various sorts, speeds, and angles can be captured more effectively by the suggested model. The high F1 of the proposed model indicates that it is able to effectively recognize vehicles in vehicle flow detection and has more reliable data accuracy. The model is evaluated on various datasets, replicating various daytime lighting situations and data noise environments, in order to confirm its resilience. The results are shown in Fig. 9.



(a) F1 values of models on different datasets

(b) F1 values of the model under different lighting conditions

(c) F1 values of the model under different noise conditions

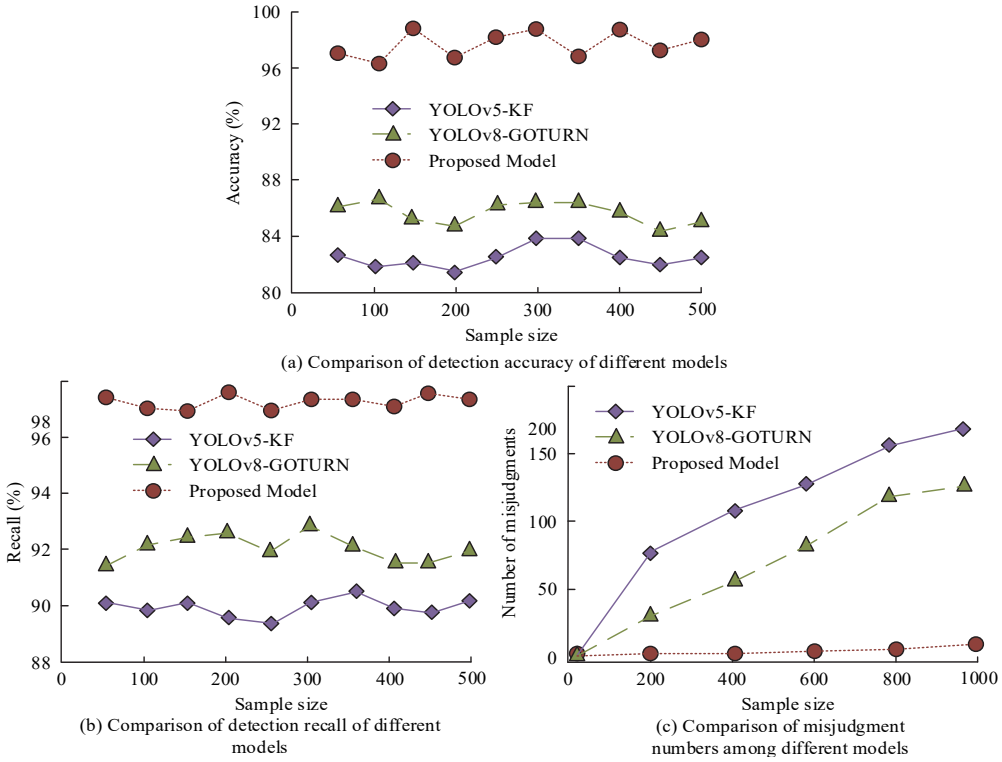
Figure 9 F1 values of the model

In Fig. 9a, the proposed model has an F1 of 0.92 in the KITTI dataset, 0.94 in the Caltech Cars dataset, and 0.94 in the UA-DETRAC dataset. In Fig. 9b, the F1 of the proposed model is 0.91 under high light conditions, 0.88 under moderate light conditions, and 0.87 under low light conditions. In Fig. 9c, under low noise conditions, the F1 of the model is 0.94. In a moderately noisy environment, the F1 of the model is 0.89. In a heavily noisy environment, the F1 of the model is 0.85. The outcome displays that the proposed model exhibits excellent robustness under different data sets and environmental conditions, has good adaptability and generalization. It is able to maintain a high level of detection performance in diverse data environments. The model is tested at a traffic intersection in an attempt to confirm the detection effect of the model in real-world scenarios. The traffic intersection is located in the center of the city, with dense surrounding commercial facilities, including shopping centers, restaurants, office buildings, and high traffic flow. The structure of the intersection is a four-fork intersection with traffic light control, and vehicles and pedestrians are staggered. This study selected the peak hours of weekdays (8:00-10:00, 17:00-19:00) as the experimental observation time. The lighting conditions in the testing environment are complex and varied (mostly cloudy, occasionally sunny, fluctuating between 5000-80000 lux). During the observation process, it was found that under strong backlighting with a solar altitude angle below 15 °, there would be a misjudgment of the reflection of metal vehicles. In addition, when the distance between the motorcycle and the car traveling in parallel is less than 0.3 meters, the system will experience tracking loss. Tab. 3 displays the findings.

Table 3 Actual application test results of proposed model

Number of experiments	Actual traffic flow	Detecting traffic flow	Vehicle inspection accuracy / %
1	124	122	98.4
2	128	128	100
3	106	105	99.1
4	94	94	100
5	108	108	100
Average value	/	/	99.5

In Tab. 3, the detection accuracy in the five experiments is 98.39%, 100%, 99.06%, 100%, and 100%, respectively. The model's excellent performance and resilience are demonstrated by the high accuracy of several tests. In the 2nd, 4th, and 5th experiments, the accuracy reaches 100%, and the accuracy in the 1st and 3rd experiments is over 98%. This indicates that the model can effectively identify actual traffic flow in multiple tests. The final average vehicle inspection accuracy is 99.49%, demonstrating the reliability and effectiveness of the model in TFD. The experimental results provide strong data support for intelligent traffic monitoring and show the potential of the proposed model in practical applications, which can effectively promote the optimization and intelligence of traffic management. To prove that the model is capable of effective detection and high stability in intelligent traffic scenarios, several different models are used to test 1000 vehicles. The results are shown in Fig. 10.



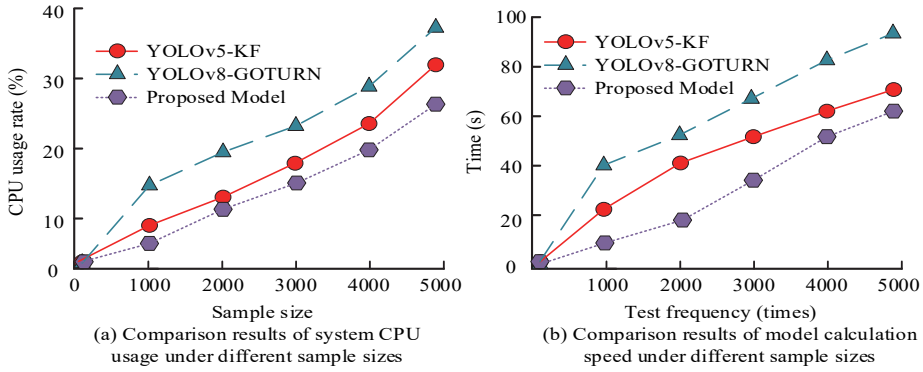
(a) Comparison of detection accuracy of different models

(b) Comparison of detection recall of different models

(c) Comparison of misjudgment numbers among different models

Figure 10 Results of misjudgement and omission test for different models

In Fig. 10a, the YOLOv5-KF model has a detection accuracy of 82.3%, YOLOv8-GOTURN has a detection accuracy of 86.4%, and the proposed model has a detection accuracy of 98.6%. In Fig. 10b, the YOLOv5-KF model detects a recall rate of 89.6%, YOLOv8 GOTURN detects a recall rate of 92.4%, and the proposed model detects a recall rate of 97.8%. Compared to YOLOv5-KF, it has increased by 9.1%. Compared to YOLOv8GOTURN, it has increased by 5.8%. In Fig. 10c, the YOLOv5 KF model has 177 false positives, YOLOv8 GOTURN detects 136 false positives, and the proposed model detects 4 false positives. Experimental data shows that the proposed model significantly outperforms YOLOv5-KF and YOLOv8-GOTURN in terms of detection accuracy and recall. In addition, the false positive rate of the proposed model is only 4, far lower than YOLOv5-KF's 177 and YOLOv8-GOTURN's 136. This shows its high advantage in recognizing the target vehicle. The data illustrates that the suggested model may offer dependable data assistance in intricate traffic conditions and has improved accuracy and stability in TFD. The computational overhead of the suggested model is evaluated in an attempt to show its potential for real-time detection. The results are shown in Fig. 11.



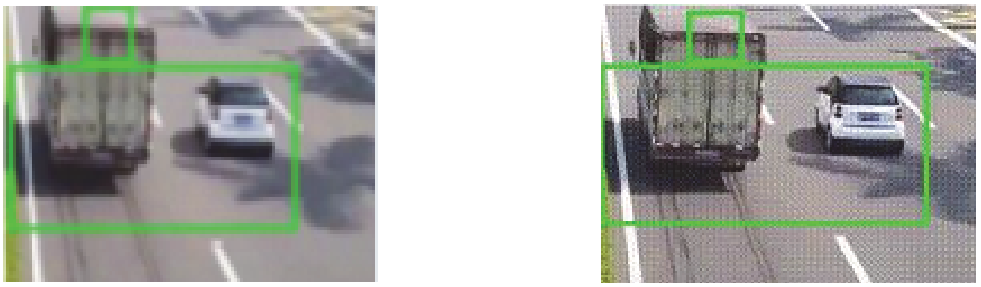
(a) Comparison results of system CPU usage under different sample sizes

(b) Comparison results of model calculation speed under different sample sizes

Figure 11 Comparison of computational resources and computational speed of different models

In Fig. 11a, YOLOv5-KF has a GPU occupancy rate of 32.6% when processing 5000 images, indicating a moderate load state. YOLOv8-GOTURN processes 5000 images with a GPU usage rate of 38.9%, also in a medium load state. The GPU usage rate for processing 5000 images using the proposed model is 24.4%, which is in a low compliance state. In Fig. 11b, YOLOv5-KF detects 5000 images in approximately 75 seconds. YOLOv8-GOTURN takes about 92 seconds to process 5000 images, while the proposed model takes about 58 seconds. The outcomes demonstrate that the suggested model works very well in terms of efficiency and resource usage, greatly lessening the strain on the GPU and cutting down on processing time. Compared to YOLOv5-KF and YOLOv8-GOTURN, it is 17 seconds and 34 seconds faster, respectively. The proposed model has been demonstrated to enhance

accuracy and demonstrate superior computational efficiency, rendering it particularly well-suited for real-time applications necessitating rapid response. The information demonstrates the suggested model's potential for use in fields like intelligent transportation monitoring and supports its competitiveness in contemporary computing settings. An example of using baseline methods and the methods proposed in the study for vehicle detection is shown in Fig. 12.



(a) Baseline method

(b) The method proposed by the research institute

Fig. 12 Example diagram of applying different methods for vehicle detection

According to Fig. 12 a, it can be seen that the baseline method is used for detection, and there are obvious burrs on the vehicle contour. As shown in Fig. 12b, the proposed method for detection results in clear and continuous metal edges of the vehicle, with significantly reduced motion blur interference. The example results indicate that the proposed method can effectively complete the vehicle target detection task, with a significant improvement in detection performance compared to baseline methods.

5 CONCLUSION

This study developed and validated a highly effective automated traffic flow detection system, integrating CLAHE, GMM, improved Canny edge detection, Faster R-CNN, and SORT algorithms. Comparative tests confirmed that this model significantly outperformed baseline methods, achieving high accuracy (0.99), recall (0.97), and an F1 score (0.98). Real-world testing further demonstrated an exceptional average vehicle detection accuracy of 99.49%, underscoring the model's practical viability and effectiveness in complex traffic environments. Robustness testing across diverse scenarios and datasets indicated strong generalization capabilities. Despite these strengths, further improvements are required to enhance detection performance under nighttime conditions and extreme weather scenarios. Future research will focus on incorporating additional sensor data and optimizing the system's adaptability to dynamic environmental changes, aiming for broader applicability and improved real-time performance. The preliminary hypothesis is as follows: provide distance and velocity information through a 77 GHz millimeter wave radar for verifying visual detection targets, and use the point cloud reflectivity characteristics of 16 line LiDAR to distinguish between real vehicles and raindrop interference. Finally, the thermal imaging data of the long wave infrared camera is integrated to achieve reliable identification in unlit environments by detecting typical thermal radiation characteristics of the vehicle.

6 REFERENCES

1. Kalsotra, R. & Arora, S. (2022). Background subtraction for moving object detection: Explorations of recent developments and challenges. *The Visual Computer*, *38*(12), 4151-4178. https://doi.org/10.1007/s00371-021-02286-0
2. Mihalca, V. O., Moldovan, O., Țarcă, I., Anton, D., & Noje, D. (2024). Integrating deep learning in target tracking applications, as enabler of control systems. International *Journal of Computers Communications & Control*, *19*(6), 6854. https://doi.org/10.15837/ijccc.2024.6.6854
3. Djerida, A., Zhao, Z., & Zhao, J. (2020). Background subtraction in dynamic scenes using the dynamic principal component analysis. *IET Image Processing*, *14*(2), 245-255. https://doi.org/10.1049/iet-ipr.2018.6095
4. Stephe, S. & Kumar, K. V. (2022). Motor imagery EEG recognition using deep generative adversarial network with EMD for BCI applications. *Tehnički vjesnik*, *29*(1), 92-100. https://doi.org/10.17559/TV-20210121112228
5. Budiraharjo, R., Sarno, R., Wijaya, D. R., Prasetyo, H. N., & Waspada, I. (2024). Simulations to predict process model alignment with standard operating procedure. *International Journal of Simulation Modelling*, *23*(1). https://doi.org/10.2507/IJSIMM23-1-657
6. Petrescu, M., Ștefan, M. C., Panagoreț, A. A., Croitoru, I. O., Chiriță, S., & Steriopol, B. C. (2024). Route planning and machine learning algorithms for sensor-equipped autonomous vehicles in agriculture. *Studies in Informatics and Control*, *33*(4). https://doi.org/10.24846/v33i4y202410
7. Hassaballah, M., Kenk, M. A., Muhammad, K., & Minaee, S. (2020). Vehicle detection and tracking in adverse weather using a deep learning framework. *IEEE Transactions on Intelligent Transportation Systems*, *22*(7), 4230-4242. https://doi.org/10.1109/TITS.2020.3014013
8. Renhao, C., Ning, C., Zhiyong, P., Shize, D., Jianmin, A., & Gang, J. (2022). Lightweight infrared dim vehicle target detection algorithm based on deep learning. *Infrared and Laser Engineering*, *51*(12), 20220253. https://doi.org/10.3788/IRLA20220253
9. Ahmed, Z. J. & Mustfai, M. A. (2023). Exploration of vehicle target detection and classification method based on sea lion optimization with deep convolutional neural network. *Journal of Smart Internet of Things*, *2022*(1), 65-80. https://doi.org/10.2478/jsiot-2022-0005

10. Wang, W., He, H., & Ma, C. (2023). An improved Deeplabv3+ model for semantic segmentation of urban environments targeting autonomous driving. *International Journal of Computers Communications & Control*, *18*(6), 5879. https://doi.org/10.15837/ijccc.2023.6.5879
11. Zeng, F., Yang, B., Zhao, M., Xing, Y., & Ma, Y. (2022). Masanet: Multi-angle self-attention network for semantic segmentation of remote sensing images. *Tehnički vjesnik*, *29*(5), 1567-1575. https://doi.org/10.17559/TV-20220421142959
12. Chang, F. R., Huang, H. L., Schwebel, D. C., Chan, A. H., & Hu, G. Q. (2020). Global road traffic injury statistics: Challenges, mechanisms and solutions. *Chinese Journal of Traumatology*, *23*(4), 216-218. https://doi.org/10.1016/j.cjtee.2020.06.001
13. Liu, W. & Zhang, L. (2022). Aerial traffic statistics based on YOLOv5+ DeepSORT. *Academic Journal of Science and Technology*, *3*(3), 198-201. https://doi.org/10.54097/ajst.v3i3.2981
14. Youssef, Y. & Elshenawy, M. (2021). Automatic vehicle counting and tracking in aerial video feeds using cascade region-based convolutional neural networks and feature pyramid networks. *Transportation Research Record*, *2675*(8), 304-317. https://doi.org/10.1177/0361198121997833
15. Valencia, D., España, E. M., & Añasco, M. M. (2024). Impact of the preprocessing stage on the performance of offline automatic vehicle counting using YOLO. *IEEE Latin America Transactions*, *22*(9), 723-732. https://doi.org/10.1109/TLA.2024.10669248
16. Agudelo, O. E. A., Marín, C. E. M., & Crespo, R. G. (2021). Sound measurement and automatic vehicle classification and counting applied to road traffic noise characterization. *Soft Computing*, *25*(18), 12075-12087. https://doi.org/10.1007/s00500-021-05766-6
17. Mandal, V. & Adu-Gyamfi, Y. (2020). Object detection and tracking algorithms for vehicle counting: A comparative analysis. *Journal of Big Data Analytics in Transportation*, *2*(3), 251-261. https://doi.org/10.1007/s42421-020-00025-w
18. Liu, C., Huynh, D. Q., Sun, Y., Reynolds, M., & Atkinson, S. (2020). A vision-based pipeline for vehicle counting, speed estimation, and classification. *IEEE Transactions on Intelligent Transportation Systems*, *22*(12), 7547-7560. https://doi.org/10.1109/TITS.2020.3004066
19. Banta, V. C., Sacala, I. S., Tutui, D., Cretu, R. F., & Serban, E. C. (2024). Manufacturing processes in the era of Industry 4.0. Case study: Analysis of a system architecture in automotive industry. *Studies in Informatics and Control*, *33*(3). https://doi.org/10.24846/v33i3y202409
20. Zhang, Y., Song, P., Jing, Q., Cretu, R. F., & Serban, E. C. (2023). A novel point cloud compression algorithm for vehicle recognition using boundary extraction. *Tehnički vjesnik*, *30*(6), 1899-1910. https://doi.org/10.17559/TV-20230507000612
21. Ma, D., Fang, H., Wang, N., Zhang, C., Dong, J., & Hu, H. (2022). Automatic detection and counting system for pavement cracks based on PCGAN and YOLO-MF. *IEEE Transactions on Intelligent Transportation Systems*, *23*(11), 22166-22178. https://doi.org/10.1109/TITS.2022.3161960
22. Zhu, G., Liu, Y., & Wang, J. (2024). Multi-frame network feature fusion model and self-attention mechanism for vehicle lane line detection. *Computer Science and Information Systems*, *54*. https://doi.org/10.2298/CSIS240314054Z
23. Wan, J., Wang, Q., & Chan, A. B. (2020). Kernel-based density map generation for dense object counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(3), 1357-1370. https://doi.org/10.1109/TPAMI.2020.3022878
24. Chen, J. I. Z. & Zong, J. I. (2021). Automatic vehicle license plate detection using K-means clustering algorithm and CNN. *Journal of Electrical Engineering and Automation*, *3*(1), 15-23. https://doi.org/10.36548/jeea.2021.1.002
25. Fernandez-Gallego, J. A., Lootens, P., Borra-Serrano, I., Derycke, V., Haesaert, G., Roldán-Ruiz, I., & Kefauver, S. C. (2020). Automatic wheat ear counting using machine learning based on RGB UAV imagery. *The Plant Journal*, *103*(4), 1603-1613. https://doi.org/10.1111/tpj.14799

Contact information:

Ke DAI
ChangChun University,
ChangChun 130022, China
E-mail: onizuka717@126.com