# Construction and Optimization of a Precise Positioning Model for Logistics Vehicles Based on Sustainable Operation

Jiashu Li, Zhenghui Tian
College of Economics and Trade, Henan Polytechnic Institute, Nanyang 473000, China
E-mail: lijiashuljsh@163.com
*Corresponding author

**Keywords:** vehicle positioning, high real-time performance, environmental adaptability, gradient descent, deep learning

**Received:** June 14, 2024

*The booming development of e-commerce drives the demand of the logistics industry. An effective logistics vehicle positioning system is crucial for improving logistics operational efficiency. However, current positioning systems based on global positioning systems and global system mobile communication suffer from issues such as low positioning accuracy and poor real-time performance. A current research focus is to improve and optimize positioning systems. To address this issue, a logistics vehicle precise positioning model based on deep learning was constructed. High-definition images were captured using digital cameras. Data augmentation and preprocessing techniques were introduced to adapt to various environments and vehicle types. These experiments confirmed that through this model, the vehicle positioning accuracy reached up to 93.3%. The positioning accuracy under urban road conditions was 96%. The AP of different types of logistics vehicles ranged from 92.4% to 94.7%, far exceeding other positioning algorithms. For CPU usage, the optimization algorithm gradually increased to 77% within 120 minutes of experimental time. Overall, this research model provides strong technical support for the logistics industry and an effective way to improve logistics operational efficiency and service quality.*

*Povzetek: Uveden je nov model pozicioniranja logističnih vozil, ki temelji na globokem učenju in v urbanih okoljih pomembno izboljša učinkovitost logistike.*

## 1 Introduction

The booming development of e-commerce has made logistics transportation an indispensable part of modern society. Therefore, the development of precise positioning technology for logistics vehicles is particularly important. At present, traditional logistics vehicle positioning technologies include Global Positioning System (GPS) and Global System for Mobile Communication (GSM). These positioning technologies still face many challenges in terms of positioning accuracy and real-time performance [1-2]. The positioning accuracy and real-time performance of these technologies are easily limited by environmental interference and device performance. In addition, due to the complexity and diversity of logistics operation scenarios, precise positioning technology is more needed [3-4].

The promotion and application of deep learning have further increased the research on precise positioning of logistics vehicles. However, the large amount of training samples required for deep learning is a major challenge, especially in obtaining high-quality and effective image samples of logistics vehicles of various types and shapes [5-6]. On the other hand, deep learning generally faces problems such as high computational complexity and high computational resource consumption when processing large-scale high-dimensional data. Therefore, how to effectively reduce the computational complexity and resource consumption of algorithms becomes a common attention in academia and industry [7-9]. Meanwhile, it is necessary to maintain the accuracy and real-time positioning of logistics vehicles.

This study consists of four parts in total. Firstly, the related works are sorted out, summarizing and elaborating on logistics vehicle positioning and deep learning networks. Next is the implementation of the proposed method. Then comes the method validation. Finally, the research results and prospects are summarized.

The precise positioning of logistics vehicles has always been a focus of industry attention. Early research relied on traditional GPS for vehicle positioning. However, due to environmental interference and equipment performance constraints, it is difficult to obtain high-precision positioning of vehicles for GPS positioning. Subsequently, GSM-based positioning technology is introduced to improve positioning accuracy through multi-base station positioning methods. For example, An et al. used the method of fuzzy theory to integrate and calculate the importance of each positioning method. Then the weighted average method was used to calculate the composite positioning, which was used as the measurement value in the FKF section. A new system architecture was designed through research. This architecture was used to calculate composite localization using methods such as fuzzy theory, weighted average method, and measurement values from the FKF part [10]. Averbakh and Yu assumed that there were m mobile service units located in a transportation network base station with n nodes. Then a certain service positioning algorithm was constructed to find an ideal warehouse location for the vehicle and minimize the expected travel distance. In general networks, this algorithm effectively reduced duplicate path selection, which was important for optimizing the operation of transportation networks [11]. Min et al. proposed a license plate localization method based on YOLO-L and license plate pre-recognition algorithm to detect license plates in complex road environments. This proposed method not only achieved an accuracy of 98.86% and a recall rate of 98.86%, but also had high efficiency in real-time performance. This new method helped to improve the accuracy and efficiency of license plate detection in various complex environments [12]. However, these methods still have limitations in complex environments and fail to meet the requirements of fast and accurate localization.

In recent years, deep learning technologies show great potential in logistics vehicle positioning. There is often a significant imbalance between the foreground (i.e. object) and background points in outdoor LiDAR point clouds. P Wu et al. proposed a novel network for object detection through semantic point-voxel feature interaction. Voxel queries using Manhattan distance were used to quickly sample voxel features around key points to achieve the efficient interaction between points and voxels. PV-RCNN++ achieved 3D mAP of 81.60%, 40.18%, and 68.21% on vehicles, pedestrians, and cyclists, respectively. Compared to the most advanced technology currently available, the performance of this method was comparable or even better. This narrowed the gap between the foreground and background points, improving the accuracy of 3D object detection [13]. Fang et al. proposed a hybrid method for robust detection to address the low percentage of object regions in current universal object detectors in processing bounding boxes. This method achieved significant improvements in both detection and localization. The practical application proved that this method was effective and practical [14]. Luo et al. utilized Faster RCNN (F-RCNN) for vehicle detection and achieved effective detection of multi-scale vehicle targets in traffic scenes. After training and testing on multiple datasets, their method exhibited advanced detection performance. This method provided a new perspective for vehicle detection and an effective method for dealing with vehicle detection problems in various traffic environments and driving conditions [15]. The findings of the above literature and this study are drawn into a table, as shown in Table 1.

Table 1: Summary of literature review

| Reference | Accuracy (%) | Computational complexity | Robustness | Methods |
|---|---|---|---|---|
| [10] | Positioning accuracy is 91.43. | Medium | High | Fuzzy theory + weighted average method +FKF |
| [11] | 92.38 | Low | Medium | Service location algorithm |
| [12] | Accuracy is 98.86, and recall rate is 98.86. | Low | High | YOLO-L model + license plate pre-recognition |
| [13] | 81.60 for vehicles, 40.18 for pedestrians, and 68.21 for cyclists. | High | Medium | Semantic point-voxel feature interaction |
| [14] | It's a big improvement from before. | High | Medium | Hybrid method |
| [15] | 98.63 | Medium | High | F-RCNN |
| This research | The mAP is 93.3 | Low | High | Deep learning + optimized F-RCNN + Adam |

In summary, while improving the accuracy of logistics vehicle positioning, current positioning models still need to consider the complexity of the model and the consumption of computing resources. Therefore, a high-precision multi-environment logistics vehicle precise positioning model is constructed in this study.

# 2 Construction of precise positioning model for logistics vehicles

An optimized F-RCNN combined with high-definition image acquisition model is utilized to improve the logistics vehicle positioning accuracy. The Adam gradient descent method is utilized to calculate the loss. The algorithm is optimized through Non-maximum Suppression (NMS). This system not only saves manpower and time costs, but also improves the logistics operation efficiency and service quality.

## 2.1 Construction of high-definition image acquisition model for logistics vehicles

The continuous improvement of operational efficiency and accuracy requirements in the logistics industry. Therefore, vehicle positioning accuracy is a research focus. The high-definition image acquisition model is responsible for providing high-quality raw data input to ensure the accuracy and reliability of subsequent localization algorithms. Considering the actual needs of logistics vehicles operating in different environments and lighting conditions, the construction of image acquisition models must take into account a wide range of adaptability. Therefore, selecting a suitable image acquisition device is a preliminary goal for model construction. Generally speaking, high-resolution digital cameras are selected as the main acquisition equipment. These cameras should have a resolution of at least 1080P and High Dynamic Range (HDR) function to adapt to high-light contrast and low-brightness environments. To ensure accurate positioning of logistics vehicles and achieve accurate recognition of vehicle features, it is first necessary to construct a high-definition image acquisition model for vehicles. Figure 1 is a common classification directory for logistics engineering vehicles.

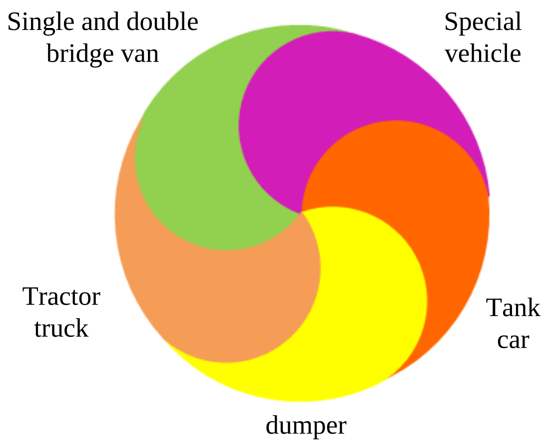



Figure 1: Logistics vehicle classification

In Figure 1, common logistics vehicles include five categories. Data augmentation technology is introduced to optimize the high-definition image acquisition system. Figure 2 shows the principle of multi-scale image scaling.

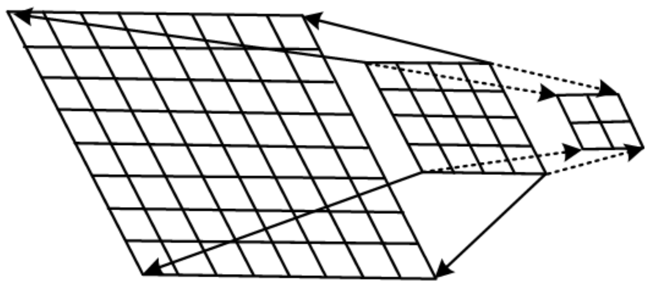


Figure 2: Picture scaling principle

In Figure 2, in vehicle image acquisition, multi-scale scaling can scale logistics vehicles while preserving the original scale features, avoiding the destruction of the original vehicle feature information. In a multi-scale scaling matrix, there is a following matrix.

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mu & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{1}$$

In formula (1), $(x_1, y_1)$ is a scaled position matrix. $(x_0, y_0)$ is a position coordinate of a pixel before scaling. $\mu$ is a scaling factor. However, in actual collection, vehicles will enter the collection area with different motion trajectories. Therefore, there is a need for oblique shooting. To adapt to various angles and shooting needs, the principle of image rotation is chosen. The image center is set as the rotation center. Therefore, the corresponding relationship before and after image rotation is represented by formula (2).

$$\begin{cases} x = r\sin\left(\dfrac{\pi}{2} - \theta - \alpha\right) = r\cos(\sigma + \alpha) \\ y = r\cos\left(\dfrac{\pi}{2} - \theta - \alpha\right) = r\sin(\theta + \alpha) \end{cases} \tag{2}$$

In formula (2), $r$ is the rotation point and the rotation center's distance. $\theta$ is a rotation angle. $(x, y)$ is a coordinate point after rotation. The frequency of image acquisition is represented by formula (3).

$$f = \frac{v}{d} \tag{3}$$

In formula (3), $f$ is the acquisition frequency. $v$ is the vehicle speed. $d$ is the required image spatial resolution. Image acquisition faces interference from environmental variables, including but not limited to weather conditions, lighting intensity, background complexity, etc. Figure 3 shows various environments.

(a) Mixing tank truck

(b) A flathead dump truck

(c) Oval tank car

(d) Square tank truck

Figure 3: Image collection in multiple environments

Figure 3 shows image collection in multiple environments. Therefore, adopting adaptive exposure control algorithm is an effective way to reduce environmental interference. This algorithm automatically adjusts exposure parameters based on the brightness level of real-time images to maintain image clarity and recognizability. Saturated lighting can adapt to logistics vehicle recognition in complex backgrounds with different lighting conditions. After calculating the maximum and minimum pixel values of logistics vehicles, saturation is calculated using formula (4).

$$\begin{cases} L = value / 2 \\ S = delta / value \;\; L < 0.5 \\ S = delta / (2 - value) \;\; L \geq 0.5 \end{cases} \tag{4}$$

In formula (4), $L$ represents brightness. $S$ refers to color saturation. $value$ is a measure of color brightness. $delta$ refers to the maximum and minimum values' difference of color. By setting saturation and adjusting lighting, adaptive exposure is represented by formula (5).

$$E = \phi * L + \upsilon \quad (5)$$

In formula (5), $E$ is the exposure value. $L$ refers to the measured value of image brightness. $\phi$ and $\upsilon$ are adjustment coefficients. During preprocessing, a Gaussian filter is selected for noise suppression, represented by formula (6).

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

In formula (6), $G(x, y)$ is a pixel value. $(x, y)$ refers to a pixel and the filtering center's distance. $\sigma$ is a standard deviation parameter.

## 2.2 Construction and optimization of logistics vehicle positioning model

The precise positioning of logistics vehicles is the key to improving operational efficiency. When constructing a precise positioning model for logistics vehicles, the study chooses F-RCNN for vehicle positioning recognition and optimization. F-RCNN combines object detection and classification functions to improve the efficiency of model training through end-to-end joint training. When locating logistics vehicles, F-RCNN can also perform regression training on different types, achieving accurate and fast positioning of logistics vehicles. Figure 4 shows the F-RCNN structure.

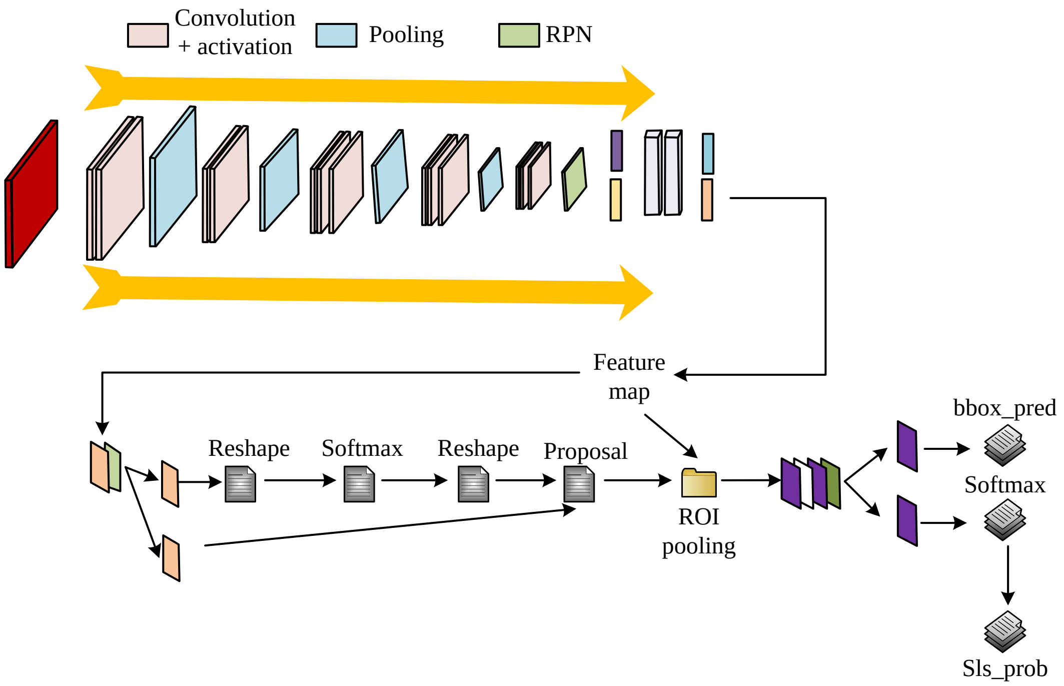



Figure 4: Network structure diagram

From Figure 4, F-RCNN combines with VGG16 extracting model and utilizes RNP for localization. However, in practical applications, F-RCNN often generates many redundant candidate boxes, which not only increases computational complexity but may also interfere with the localization accuracy. Therefore, integrating NMS into F-RCNN and suppressing local maximum search and non-maximum elements can reduce redundant candidate boxes. In the calculation of the intersection area of redundant boxes, the candidate box overlap's area is represented by formula (7).

$$S_{overlap} = W_{overlap} * H_{overlap} \quad (7)$$

In formula (7), $S_{overlap}$ is the area of the overlapping area. $W_{overlap}$ refers to the width. $H_{overlap}$ is the height. $W_{overlap}$ is represented by formula (8).

$$W_{overlap} = Min(x_2, x_4) - Max(x_1, x_3) + 1 \quad (8)$$

$H_{overlap}$ is represented by formula (9).

$$H_{overlap} = Min(y_2, y_4) - Max(y_1, y_3) + 1 \quad (9)$$

In formulas (8) and (9), $x_2$ and $x_3$ are the abscissa of the rightmost and leftmost intersection areas. $x_1$ and $x_4$ are the rightmost horizontal coordinates. $y$ is the same. Therefore, the intersection and union ratio is represented by formula (10).

$$IoU = \frac{S_{overlap}}{S_A \cup S_B} \tag{10}$$

In formula (10), $S_A$ and $S_B$ are the suggested box areas with the highest confidence scores. The Adam gradient descent method is chosen for calculating the loss to improve the recognition effect. The gradient is represented by formula (11).

$$g_t = \nabla \theta f_{t(\theta)} \tag{11}$$

In formula (11), $\theta$ refers to the model's all parameters using this update rule. The unbiased estimation of the square gradient at each moment is represented by formula (12).

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) g_t \tag{12}$$

The unbiased estimation of gradients at each time step is represented by formula (13).

$$v_t = \beta_2 \cdot v_l + (1 - \beta_2) \cdot g_{t^2} \tag{13}$$

In formulas (12) and (13), $\beta_1$ and $\beta_2$ are hyperparameters, usually set to 0.9 and 0.999, respectively. $g_t$ is the corresponding gradient value.

Therefore, the parameter update is represented by formula (14).

$$\theta_j = \theta_j - \alpha \cdot m \cdot \nabla_{\theta_j} J(\theta) \tag{14}$$

In formula (14), $\alpha$ is a learning rate. $m$ is the batch size of the corresponding batch of samples. $n$ is a parameter dimension. To adjust $\alpha$, for a given dataset, the average loss minimum strategy is used for optimization, represented by formula (15).

$$L(W) = \frac{1}{|D|} \sum_t^{|D|} f_w(X^{(i)}) + \lambda \cdot R(W) \tag{15}$$

In formula (15), $F_w(X^{(i)})$ is a loss function of the data. $R(W)$ is a regular term. $\lambda$ is a weight. To increase $\alpha$, the decay of the learning rate is ensured through the deployment form, represented by formula (16).

$$\alpha = \alpha \cdot \left(1 - \frac{iter}{\max\_iter}\right)^{power} \tag{16}$$

In formula (16), $power$ is a power function parameter. $iter$ refers to the current iteration. $\max\_iter$ refers to the total iteration. Figure 5 shows the optimization algorithm.

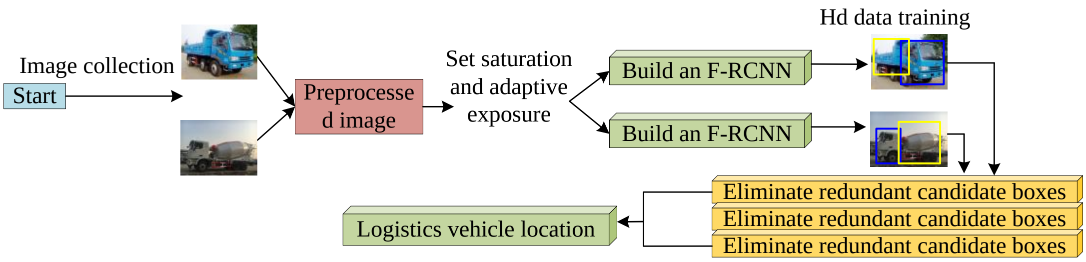



Figure 5: Simple algorithm flow

From Figure 5, based on F-RCNN, image acquisition and data processing are combined for vehicle recognition and localization. In the high-definition image acquisition model of logistics vehicles, a high-resolution digital camera is selected to cope with different lighting conditions with 1080P resolution and HDR function. Meanwhile, high-definition images are obtained. Data augmentation technology is adopted to adapt to the actual situation of logistics vehicles entering the collection area from multiple angles. During preprocessing, Gaussian filters are selected for noise suppression. After obtaining high-definition images of logistics vehicles, F-RCNN is constructed for logistics vehicle localization and recognition. F-RCNN generates many redundant candidate boxes in applications. This may increase computational complexity and accuracy in interference localization. Therefore, NMS is integrated into F-RCNN to screen redundant candidate boxes. Finally, the Adam gradient descent method is used to calculate the loss and improve the recognition performance. In the gradient calculation, the unbiased estimation of square gradients and unbiased estimation of gradients are used to adjust model parameters. The adjustment of $\alpha$ is mainly achieved through minimizing average loss and reducing $\alpha$ through deployment. Specifically, data enhancement simulates the vehicle entering the acquisition area from different angles by applying a rotation transform to the captured image. The imaging effect of vehicles at different distances is simulated by multi-scale images. Color conversion includes brightness, contrast, and saturation adjustment to simulate vehicle images under different lighting conditions. Random noise is added to the image to simulate the possible image noise in the actual acquisition process. Then the model robustness to noise can be improved. The image is flipped horizontally to increase data diversity while simulating a vehicle entering the acquisition area from the opposite direction. A part of the original image is cropped randomly to simulate the local view when the image is collected. The model's recognition ability to the local features of the vehicle is enhanced. In the pre-processing step, Gaussian filter is used to de-noise the image. The Gaussian filter can reduce the noise that may be introduced in the image acquisition and improve the image quality. The histogram of the image is equalized to enhance the contrast of the image. Then the model can recognize the detailed features of the vehicle better. The image pixel values are normalized to the range of 0 to 1 to accelerate the convergence of model training. Meanwhile, the model adaptability to different lighting conditions can be improved. Each channel of the image is normalized to have a mean of 0 and a standard deviation of 1. Through data enhancement techniques, the sample diversity in the data set is significantly increased, which helps the model learn a more comprehensive feature representation. The enhanced data set enables the model to better generalize to previously unseen data, reducing the risk of overfitting. The pre-processing steps ensure the stability and robustness of the model when dealing with noise and illumination changes. Adam gradient descent optimization sets an initial learning rate at the beginning of training. As the training progresses, the learning rate can be attenuated according to a predetermined plan to ensure the stability of convergence. A regular term is added to the loss function to avoid overfitting. This is equivalent to applying an attenuation to the parameter value each time the parameter is updated. L1 regularization is similar to L2 regularization, but uses the sum of the absolute values of the parameters. L1 regularization helps to produce sparse weight matrices and sometimes improves the interpretability of the model. Adam algorithm includes momentum, which calculates an exponentially weighted average of a gradient to produce an effect similar to momentum. RMSprop calculates the square root of the exponentially weighted average of the square gradient, which is used to adjust the learning rate for each parameter. Based on the estimates of the first and second moments, an adaptive learning rate is calculated for each parameter. In the beginning, the estimates of first and second moments may be biased. Adam algorithm solves this problem by correcting for the bias. Finally, the calculated adaptive learning rate is used to update the model parameters.

# 3 Analysis of precise positioning model for logistics vehicles

Firstly, the configuration of the experimental environment was determined to analyze the optimized logistics vehicle positioning model. The algorithm testing was conducted in different environments. Afterwards, this model was applied in practice and evaluated based on the satisfaction ratings of practitioners towards its use in actual operating environments.

## 3.1 Model experiment analysis

The experimental data set comes from the pictures collected on the site of the logistics park. After one year's collection, 10,000 pictures with good features are selected. The dataset contains 10,000 images for training, validation, and testing. Data enhancement methods such as multi-scale equal scaling, random image flipping, and saturated illumination are carried out to increase the diversity of samples and solve the problem of sample uniformity. The data sets are divided into training, validation, and test sets, with the ratio of 10:1:1 to ensure independent co-distribution between samples and balance between categories. The batch size is 2 pictures. The total training iteration is 90,000 times. Since each image goes through 20 iterations, for a training set of 9,000 images, this is equivalent to training 20 epochs. The basic learning rate is set to 0.001. The multi-stage learning rate adjustment strategy is adopted. The learning rate is adjusted to the original 0.1 when the iterations are [22500, 45,000, 90,000]. Hyperparameter uses Adam optimization algorithm, which combines the advantages of AdaGrad and RMSProp algorithms. The Adam optimization algorithm has the ability to adjust the learning rate of each parameter adaptively. Batch Normalization is also used to speed up the training and reduce the reliance on other regularization methods. Table 2 shows the experimental configuration.

Table 2: Experimental environment configuration

| Form | Recommended products | Note |
|---|---|---|
| Digital camera | Canon 5D | With high resolution and HDR function, it adapts to different lighting environments. |
| Computing hardware | NVIDIA GeForce RTX 3090 | Powerful GPU performance for running computationally demanding models. |
| Storage hardware | Seagate Exos X16 16TB Enterprise | Huge storage space, stable, and |

| | | |
|---|---|---|
| | Hard Drive | reliable performance. |
| Operating system | Ubuntu 20.04 LTS | Combining stability and new features, it is widely used in scientific research. |
| Development environment (computer) | Python 3.7 | Tested for best compatibility with most Python libraries. |
| Deep learning framework | TensorFlow 2.x | Provide powerful model building and training capabilities. |
| Image processing library | OpenCV 4.5.2 | Provide rich image processing functions, including image enhancement, noise reduction, and so on. |
| Data visualization library | Matplotlib 3.3.4 | Used for data visualization and drawing complex statistical charts. |

In the precise positioning system for logistics vehicles, the Canon 5D digital camera was chosen as the camera, which had high resolution and HDR function to adapt to image acquisition under different lighting conditions. For hardware, NVIDIA GeForce RTX 3090 was selected for GPU. Seagate Exos X16 16TB enterprise hard drive was used for storage hardware. For software, the Ubuntu 20.04 LTS operating system was chosen, with Python 3.7 as the development environment. TensorFlow 2.x and OpenCV 4.5.2 served as deep learning frameworks and image processing libraries, respectively. Matplotlib 3.3.4, as a data visualization library, can provide intuitive feedback information during training. To ensure the detection and positioning of logistics vehicles in different environments, multiple environments were selected for accuracy testing in Figure 6.

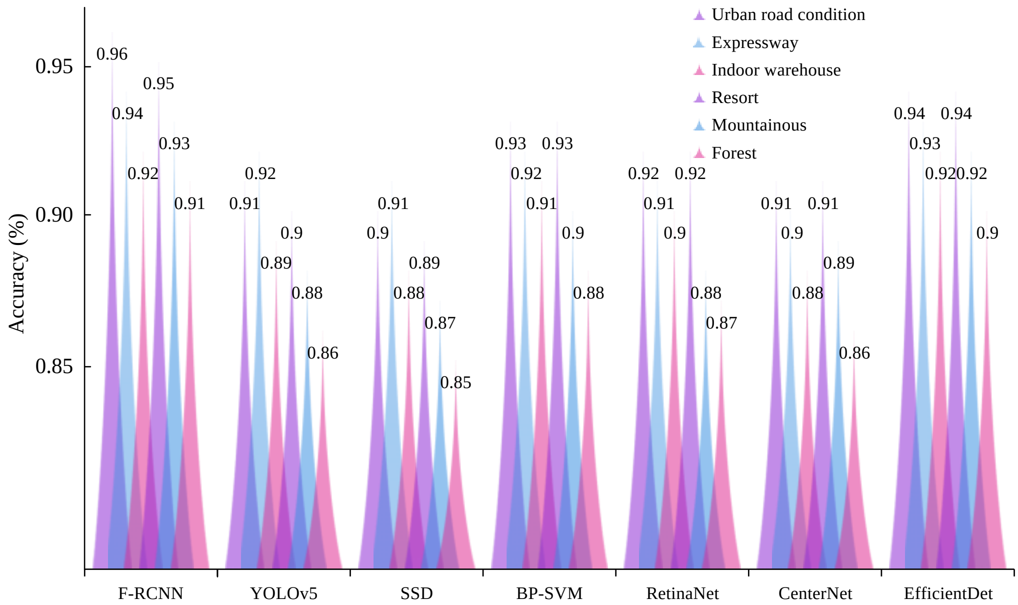


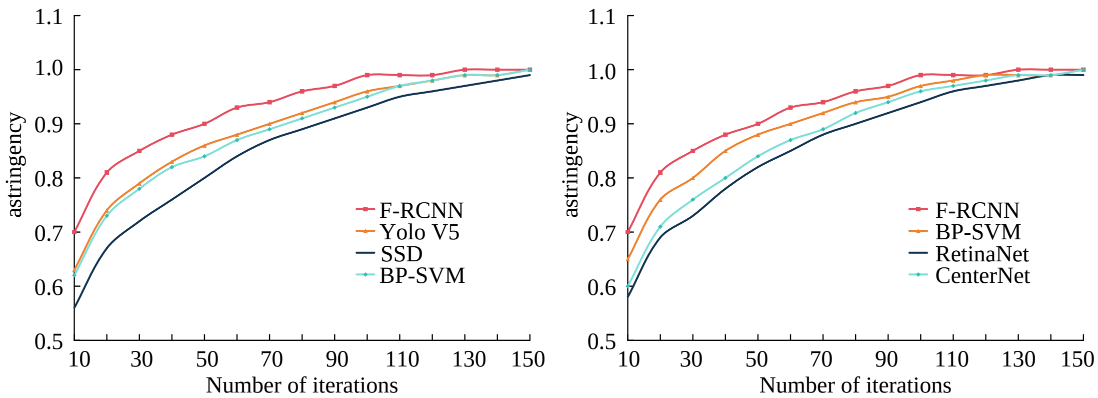

Figure 6: Comparison of accuracy

Figure 6 shows the comparison of logistics vehicle positioning accuracy under different algorithms. The testing scenarios include urban road conditions, expressways, indoor warehouses, villas, mountains, and forests. In these six testing environments, F-RCNN achieved an accuracy of 96% in urban road conditions, 90% in SSD, and 93% in BP-SVM. In indoor warehouses, the accuracy of this optimization algorithm exceeded 90%, higher than other centralized algorithms, with only ED remaining the same. In logistics vehicle positioning in expressways and complex natural environments (mountains and forests), this optimization algorithm performed better than other algorithms and had better positioning effects. In stability analysis, the reliability comparison under different testing environments was compared in Table 3.

Table 3: Stability comparison

| Algorithm | Failure rate | Precision fluctuation | | | | | |
|---|---|---|---|---|---|---|---|
| | | Urban roads | Express ways | Indoor warehous es | Villas | Mountains | Forests |
| F-RCNN | 0.001 | 0.01 | 0.02 | 0.02 | 0.01 | 0.02 | 0.02 |
| YOLOv5 | 0.002 | 0.03 | 0.04 | 0.04 | 0.05 | 0.03 | 0.04 |
| SSD | 0.005 | 0.05 | 0.04 | 0.04 | 0.05 | 0.05 | 0.06 |
| BP-SVM | 0.006 | 0.03 | 0.04 | 0.04 | 0.04 | 0.03 | 0.03 |
| RetinaNet | 0.004 | 0.04 | 0.05 | 0.04 | 0.05 | 0.05 | 0.04 |
| CenterNet | 0.003 | 0.05 | 0.05 | 0.05 | 0.06 | 0.06 | 0.05 |
| EfficientDet | 0.001 | 0.05 | 0.06 | 0.02 | 0.03 | 0.05 | 0.04 |

From Table 3, the failure rates of F-RCNN, SSD, YOLOv5, BP-SVM, RetinaNet, CenterNet, and EfficientDet were all within 0.007, indicating relatively high stability. F-RCNN had the lowest failure rate, only 0.001. For accuracy, F-RCNN had a fluctuation of only 0.01 in urban and mountainous environments, which was the smallest among all algorithms. The fluctuations of F-RCNN in other environments did not exceed 0.02. This indicated that F-RCNN was ahead of other algorithms in terms of stability and accuracy. For accuracy fluctuations, EfficientDet achieved fluctuations of 0.06 and 0.02 in indoor and villa environments, respectively. This was the highest among all algorithms, indicating that EfficientDet might have unstable outputs in these environments. CenterNet's accuracy fluctuated above 0.05 in all environments, and even reached 0.06 in complex natural environments. This indicated that the performance of CenterNet was not ideal in complex environments. In summary, in achieving precise positioning of logistics vehicles, considering both stability and accuracy fluctuations, this optimization algorithm performed well overall and had high robustness. Figure 7 shows a comparison of algorithmic convergence speed.



(a) Convergence rate comparison
(b) Convergence rate comparison

Figure 7: Algorithmic convergence speed comparison

Figure 7 shows the comparison of convergence rates between different algorithms. F-RCNN had the best performance among all compared algorithms, reaching a convergence value of 1.00 at iteration 150. At 130 iterations, the convergence value of F-RCNN was close to 1.00, indicating a faster convergence speed. On the contrary, the convergence values of other recognition algorithms did not achieve the effect of F-RCNN under the same iteration. Although EfficientDet converged to 1.00 at iteration 150, its convergence value was only 0.99 at iteration 130, still lower than F-RCNN. The loss functions of F-RCNN, YOLOv5, SSD, and BP-SVM algorithms are shown in Figure 8.

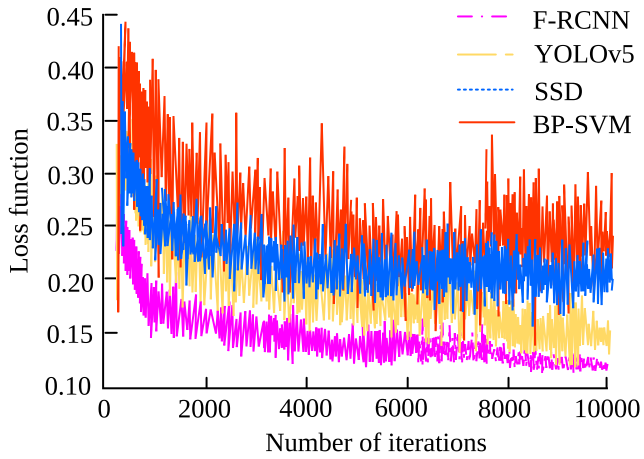



Figure 8: Loss function diagram of F-RCNN, YOLOv5, SSD, and BP-SVM algorithms

In Figure 8, F-RCNN showed a lower loss function value, which had less deviation from the actual value during the prediction. After 1000 iterations, F-RCNN successfully reduced the loss value to below 0.20, which highlighted the superior performance and fast convergence rate. In contrast, the BP-SVM algorithm had a higher loss function value with a peak value of 0.45, indicating a large bias in the prediction. Meanwhile, more time or adjustment is needed to optimize the parameters to achieve a similar performance as F-RCNN.

## 3.2 Practical application analysis of the model

In practical application analysis, the evaluation indicators of various types of logistics engineering vehicle models were first compared in Table 4.

Table 4: Resolution and high dynamic range

| Category | Simple bridge | Double bridges | Semi-trailer | Full hanging | Flat | Pointed | Ellipse | Square | Stir | mAP |
|---|---|---|---|---|---|---|---|---|---|---|
| Original FasterR-CNN | 82 | 83.3 | 86.8 | 81.2 | 80.7 | 85.5 | 84.6 | 80.9 | 83.2 | 84.2 |
| Sample_mining_fc | 83.8 | 84.6 | 82 | 85.9 | 85.3 | 82.8 | 85.7 | 84.8 | 82.6 | 84.7 |
| Sample mining_1:1 | 84 | 81.8 | 83.1 | 85.6 | 85.3 | 84.7 | 85 | 85.4 | 85 | 85.1 |
| Sample Dig_1:10 | 85.4 | 84 | 86.2 | 86.3 | 87.4 | 86.8 | 86.2 | 86.5 | 87.5 | 86.4 |
| Sample Dig_1:3 | 88.5 | 87.6 | 88.7 | 87.4 | 89.6 | 88.9 | 89.5 | 88.8 | 89.9 | 88.7 |
| Data enhancement | 88.6 | 88.8 | 90.5 | 89 | 90.6 | 89.3 | 90.6 | 89.7 | 90 | 89.8 |
| BP-SVM | 89.8 | 90.8 | 91.8 | 90.9 | 92.7 | 91.5 | 90.5 | 90.9 | 91.7 | 91.3 |
| Optimization algorithm | 92.8 | 93.3 | 93.8 | 92.9 | 94.7 | 93 | 93.5 | 92.4 | 93.2 | 93.3 |

From Table 4, for single-bridge, double-bridge, and semi-trailer vehicles, the AP of this optimization algorithm reached 92.8%, 93.3%, and 93.8%, all higher than other algorithms. In full hanging, flat, and pointed logistics vehicles, this optimization algorithm also achieved high-precision positioning rates of 92.9%, 94.7%, and 93%. In detecting ellipse, square, and stir logistics vehicles, this optimization algorithm achieved accuracy of 93.5%, 92.4%, and 93.2%, respectively. When calculating the Mean Average Precision (mAP) of all categories, this optimization algorithm achieved a high accuracy of 93.3%, far exceeding other localization algorithms. Figure 9 shows the CPU usage without the algorithm.

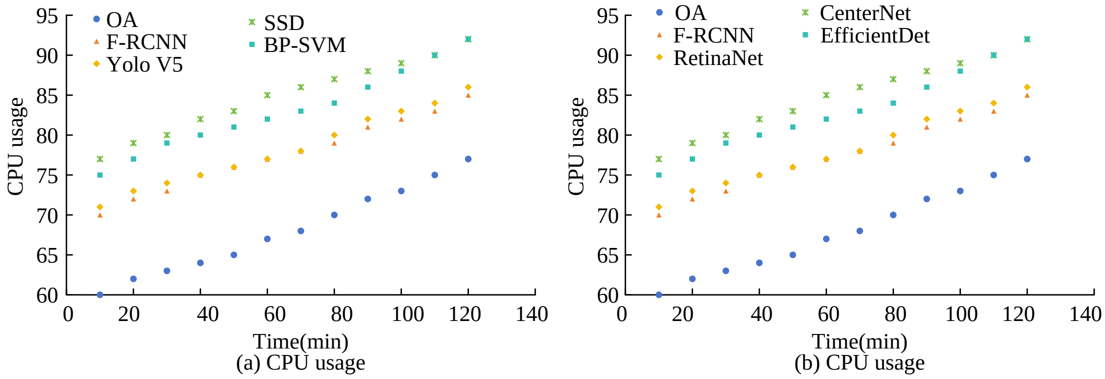



Figure 9: CPU usage

Figure 9 shows the CPU usage of optimization algorithms and other mainstream logistics vehicle positioning methods at different time nodes. The CPU usage of this optimization algorithm gradually increased from 60% in 10 minutes to 77% in 120 minutes. As time went on, this optimization algorithm gradually increased its CPU usage in handling logistics vehicle positioning tasks. However, compared to other algorithms, the CPU usage of this optimization algorithm always remained at a relatively low level. At 120 minutes, CenterNet had the highest usage rate, reaching 92%. Compared to 77% of this optimization algorithm, CenterNet posed greater pressure on the CPU. The CPU utilization of F-RCNN was 70% at 10 minutes and increased to 85% at 120 minutes. The usage rate of CenterNet increased from 77% to 92%.

Overall, this optimization algorithm had lower CPU usage compared to other mainstream positioning algorithms when handling logistics vehicle positioning tasks, meaning less pressure on the CPU. This indicated that this optimization algorithm was superior to other localization algorithms in resource management. After practical application, the system was reasonably rated by relevant practitioners. The rating is based on a score of 1-10, with 1 indicating very dissatisfied and 10 indicating very satisfied. Table 5 shows the scoring results.

Table 5: Comparison of satisfaction scores

| Sports event | Optimization algorithm | BP-SVM | SSD | Yolov5 | RetinaNet | CenterNet | EfficientDet |
|---|---|---|---|---|---|---|---|
| Accuracy | High (9.5) | Excellent (8.4) | Good (7.5) | Good (7.3) | Good (7.6) | Good (7.4) | Excellent (8.1) |
| Topicality | Extremely fast (9.8) | Faster (8.0) | Faster (8.1) | Medium (6.5) | Medium (6.9) | Slower (5.7) | Medium (6.6) |
| Stability | Very stable (9.6) | Stable (8.2) | More stable (7.2) | More stable (7.1) | Stable (8.3) | Stable (8.0) | More stable (7.4) |
| Anti-noise | High (9.3) | High (8.6) | Medium (7.1) | Medium (7.0) | High (8.7) | High (8.5) | High (7.8) |
| Usability | Intuitive (9.5) | More intuitive (8.1) | More intuitive (7.9) | General (6.8) | General (6.6) | Difficult (5.9) | More intuitive (7.8) |
| Scope of application | Wide (9.4) | Broader (8.3) | Broader (7.8) | General (6.5) | Hiro (8.5) | Hiro (8.9) | Broader (7.6) |
| Total satisfaction | Very happy | Dissatisfied | More satisfied | General | Dissatisfied | Dissatisfied | More satisfied |

From Table 5, this optimization algorithm had the highest accuracy with a score of 9.5. This algorithm performed excellently in the prediction task of logistics vehicle positioning. For real-time performance, this optimization algorithm scored over 9.5, making it suitable for real-time and efficient positioning tasks. This optimization algorithm had the highest stability with a score of 9.6, indicating that it maintained high accuracy in various environments and data. For noise resistance, this optimization algorithm (9.3 points) also demonstrated excellent adaptability, ensuring good stability in the face of noise and flawed data. For ease of use, this optimization algorithm was intuitive and easy to learn, had a wide range of applications, and met the practical needs of various logistics vehicle positioning. Compared with other algorithms, this optimization algorithm showed significant advantages in both single evaluation metrics and overall satisfaction. Based on a comprehensive analysis of satisfaction, this optimization algorithm has superiority in the precise positioning task of logistics vehicles. Figure 10 shows the comparison of localization and classification performance between F-RCNN and SSD algorithms. F-RCNN performed better in localization and classification. SSD was more dispersed and prone to mislocalization.

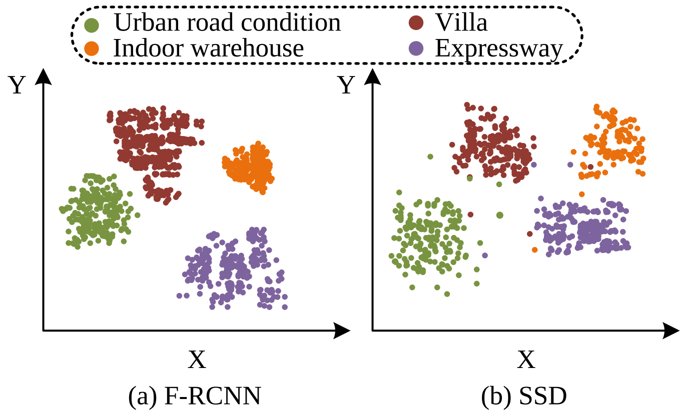



Figure 10: Positioning classification effect

Finally, the performance of the proposed logistics vehicle positioning model in different environments was visualized, as shown in Figure 11.

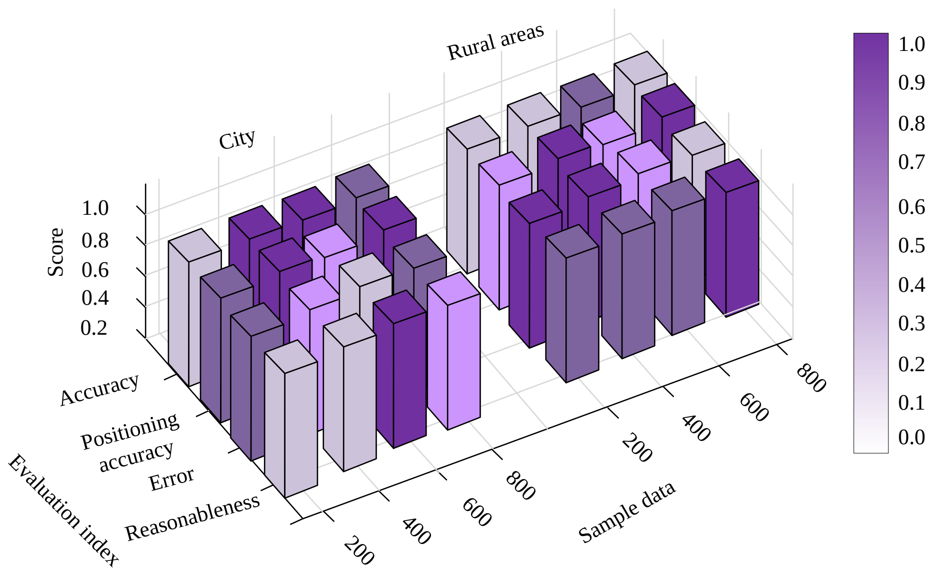



Figure 11: Visual presentation of model performance in different environments

In Figure 11, the accuracy, positioning accuracy, error, and reasonableness scores of the proposed logistics vehicle positioning model were all above 0.9 in both city and rural environments. The proposed logistics vehicle location model could provide high quality location service in various environments, which had good generalization ability and practicability. This has important practical significance for the operation management and vehicle scheduling of the logistics industry. The method can significantly improve logistics efficiency and reduce unnecessary costs.

# 4 Discussion

A precise positioning model of logistics vehicles based on deep learning was constructed and optimized to improve the accuracy and real-time location of logistics vehicles. The proposed model achieved an average accuracy of 93.3%, which was much higher than other methods. For example, the positioning accuracy of reference [10] was 91.43%, and that of reference [11] was 92.38%. Although reference [12] performed well in complex road environment, its accuracy and recall were both 98.86%, which still did not exceed the model in this study. Especially for single bridge, double bridge, and semi-mounted vehicles, the AP values of the optimization algorithm reached 92.8%, 93.3%, and 93.8%, respectively, showing extremely high positioning accuracy. The proposed model exhibited lower resource consumption in terms of computational complexity. On the basis of the optimization of F-RCNN, Adam gradient descent method and NMS were optimized to achieve low computational complexity. In contrast, the method in reference [13] is excellent in 3D object detection. However, this method had high computational complexity and might not be suitable for real-time applications. In this study, multi-scale scaling, image rotation, color transformation, and other data enhancement techniques were used to improve the model's adaptability to different environmental and lighting conditions. Meanwhile, Gaussian filter and other pre-processing steps were used. F-RCNN was selected and optimized by introducing Adam gradient descent method and NMS. The improved model reduced computational complexity and improved real-time performance while maintaining high accuracy. The adaptive exposure control algorithm and Gaussian filter were used to reduce the environmental interference and improve the model robustness in complex environment.

# 5 Conclusion

Logistics transportation is an important component of modern social life. Therefore, the positioning accuracy and real-time performance can be improved by studying a logistics vehicle precise positioning model based on deep learning. These experiments confirmed that this model exhibited high-precision and high real-time positioning advantages in various environments and vehicle types. Compared to other algorithms, this optimization algorithm had higher robustness in achieving precise positioning of logistics vehicles. Its CPU usage was low, demonstrating excellent resource management capabilities, thereby improving operational efficiency and service quality. In the precise positioning task of logistics vehicles, this model exhibited high positioning accuracy, with an mAP of 93.3%. Its positioning accuracy in specific types of logistics vehicles such as single-bridge, double-bridge, semi-trailer, etc. was as high as 92.4% to 94.7%. Secondly, for operational efficiency, this model exhibited superior performance, with a usage rate of only 77% at 120 minutes. The usage rate was significantly lower than other logistics vehicle positioning algorithms. Finally, the logistics vehicle positioning system using this model achieved a satisfaction score of 9.5 among practitioners, demonstrating a wide range of application prospects and user acceptance. However, the lack of research is due to the limited range of high-quality image acquisition. In the future, multi-source data can be used to further improve the positioning accuracy of logistics vehicles and meet the needs of more complex and diverse scenarios. Overall, this constructed logistics vehicle precise positioning model based on deep learning effectively improves the positioning accuracy and real-time performance of logistics vehicles.

# Declarations

Availability of data and material. All data generated or analyzed during this study are included in this article.

# Competing interests

The authors declare that they have no competing interests.

# Funding

There is no funding in this article.

# Authors' contributions

Jiashu Li wrote the draft manuscript and Zhenghui Tian revised the manuscript critically. Both authors reviewed and approved it for publication.

# Acknowledgements

Not applicable.

# References

[1] Chen Mingzhou, Zhang Shuai, Zhang Wenyu, and Lin Jian. Collaborative vehicle routing problem with rough location using extended ant colony optimization algorithm. Journal of intelligent & fuzzy systems, 37(2):2385-2402, 2019. https://doi.org/10.3233/JIFS-182715

[2] Zühal Kartal. Integrated hub location and capacitated vehicle routing problem over incomplete hub networks. International journal of industrial engineering, 30(1):256-272, 2023. https://doi.org/10.23055/ijietap.2023.30.1.8025

[3] A K G Revathi, Jeba Belsam, Ananth, Saravanan Lakshmanan, and Ranjith Kumar Anandan. Gps enabled vehicle location identification using gsm and fare collection using smart card. Turkish journal of computer and mathematics education, 12(10):2657-2668, 2021. https://creativecommons.org/licenses/by/4.0 (the

“License”)
[4] Sandipan Choudhuri, Suli Adeniye, and Arunabha Sen. Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation. Artificial intelligence and applications, 1(1):43-51, 2023. https://doi.org/10.47852/bonviewAIA2202524
[5] Tengxian Xu, Xianpeng Wang, Ting Su, Liangtian Wan, and Lu Sun. Vehicle location in edge computing enabling IoTs based on bistatic FDA-MIMO radar. IEEE access, 2(1):1-2, 2021. https://doi.org/10.1109/ACCESS.2021.3064849
[6] Donghun Lee, Kyungin Min, and Jungha Kim. Wireless LAN-based vehicle location estimation in GPS shading environment. The journal of the Korea institute of intelligent transport systems, 19(1):94-106, 2020. https://doi.org/10.12815/kits.2020.19.1.94
[7] Mohammad S. Alzyout, and Mohammad A. Alsmirat. Performance of design options of automated ARIMA model construction for dynamic vehicle GPS location prediction. Simulation modelling practice and theory, 104(1):2-26, 2020. https://doi.org/10.1016/j.simpat.2020.102148
[8] Karthik V, Manoj Balaji K, Manoj Kumar M, and Manoj Kumar N. Traffic controller for vip/ambulance vehicle & location tracking. International research journal of multidisciplinary technovation, 3(1):6-9, 2021. https://doi.org/10.34256/irjmt2112
[9] Sadia Nur Amin, Palaiahnakote Shivakumara, Tang Xue Jun, Kai Yang Chong, Dillon Leong Lon Zan, and Ramachandra Rahavendra. An augmented reality-based approach for designing interactive food menu of restaurant using android. In artificial intelligence and applications, 1(1):26-34, 2023. https://doi.org/10.47852/bonviewAIA2202354
[10] Jiyao An, Yingjun Yu, Jie Tang, and Jiawei Zhan. Fuzzy-based hybrid location algorithm for vehicle position in VANETs via fuzzy kalman filtering approach. Advances in fuzzy systems, 2019(1):1-11, 2019. https://doi.org/10.1155/2019/5142937
[11] Igor Averbakh, and Wei Yu. The probabilistic uncapacitated open vehicle routing location problem. Networks, 82(1):68-83, 2023. https://doi.org/10.1002/net.22147
[12] Weidong Min, Xiangpeng Li, Qi Wang, Qingpeng Zeng, and Yanqiu Liao. New approach to vehicle license plate location based on new model YOLO-L and plate pre-identification. IET image processing, 13(7):1041-1049, 2019. https://doi.org/10.1049/iet-ipr.2018.6449
[13] Peng Wu, Lipeng Gu, Xuefeng Yan, Haoran Xie, Fu Lee Wang, Gary Cheng, and Mingqiang Wei. PV-RCNN++: semantical point-voxel feature interaction for 3D object detection. The visual computer, 39(6):2425-2440, 2023. https://doi.org/10.1007/s00371-022-02672-2
[14] Fen Fang, Liyuan Li, Hongyuan Zhu, and Joo-Hwee Lim. Combining faster R-CNN and model-driven clustering for elongated object detection. IEEE transactions on image processing, 29(1):2052-2065, 2019. https://doi.org/10.1109/TIP.2019.2947792
[15] Ji-qing Luo, Hu-sheng Fang, Fa-ming Shao, Yue Zhong, and Xia Hua. Multi-scale traffic vehicle detection based on faster R-CNN with NAS optimization and feature enrichment. Defence technology, 17(4):1542-1554, 2021. https://doi.org/10.1016/j.dt.2020.10.006