Heliyon 10 (2024) e26467

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Real-time video surveillance on highways using combination of extended Kalman Filter and deep reinforcement learning

Liangju Fu [a], Qiang Zhang [b,*], Shengli Tian [c]

[a] School of Civil Engineering, Chongqing Jiaotong University, Chongqing, 400074, China
[b] School of Computer Science and Technology, XiDian University, XiAn, 710071, China
[c] College of Transportation Engineering, Chang'An University, XiAn, 710064, China

ARTICLE INFO

Keywords:
Video surveillance
Highway monitoring
Deep learning
Reinforcement learning
Kalman filter

ABSTRACT

Highways, as one of the main arteries of transit and transportation in today's world, play a fundamental role in accelerating transportation, and for this reason, continuous monitoring of them is of great importance. Among these, monitoring compliance with transportation laws by vehicles is of utmost importance; for automation, efficient and vehicle-specific models can be used. In this article, a new method for video surveillance of highways is presented using an extended Kalman filter (EKF) and reinforcement learning models. There are three primary stages to the suggested approach. During the first stage, the extended Kalman filter (EKF) is used to identify and track multiple targets. Next, in the second stage, a convolutional neural network (CNN) processes each detected moving item to determine the kind of vehicle. During this stage, the CNN model's ideal configuration is ascertained using a new optimization approach that combines Particle Swarm Optimization (PSO) and reinforcement learning. After identifying the type of vehicle, in the third phase, the proposed method uses a separate CNN model for each target vehicle to assess its compliance with transportation safety principles. It should be mentioned that each vehicle's associated CNN model is configured during this phase using the suggested optimization methodology. Investigations have been conducted into the effectiveness of the suggested method in identifying violations of road safety laws as well as how well it performed in the two phases of vehicle type identification. According to the findings, the suggested approach can identify the kind of vehicle with 98.72% accuracy, which is at least 3.41% better than the approaches that were compared. On the other hand, this model can detect the violation of road safety laws for each vehicle with an average accuracy of 91.5%, which shows at least a 3.49% improvement compared to the other methods.

1. Introduction

Highways are considered one of the main components of transportation in major cities, and video surveillance of highways is of great importance for enhancing safety and reducing road accidents [1]. Using surveillance cameras, drivers can be assisted in adhering to traffic laws to keep themselves and others safe on the highways. Additionally, the data collected by surveillance cameras can contribute to improving traffic conditions and transportation in the city [2]. For example, by analyzing the collected data, dense and accident-prone routes can be identified and efforts can be made to improve them [3]. Identifying stolen vehicles [4], recognizing

* Corresponding author.
E-mail address: king_zq@outlook.com (Q. Zhang).

https://doi.org/10.1016/j.heliyon.2024.e26467
Received 13 October 2023; Received in revised form 12 February 2024; Accepted 14 February 2024
Available online 15 February 2024

criminals [5], and automatic detection of driving violations [6] are among the applications of video surveillance on road transportation arteries.

In this context, automatic detection of compliance with transportation safety principles is of paramount importance. Failure to adhere to these principles can lead to serious accidents and pose a threat to people's lives. This process is more complex compared to the process of detecting driving violations because in the process of detecting driving violations, laws are uniformly applied to all vehicles, and violations (such as running a red light, changing lanes, unauthorized overtaking, etc.) are identified in a similar manner for every vehicle. In contrast, compliance with transportation safety principles varies for each type of vehicle and is subject to specific and proportional conditions [7]. For example, not wearing a helmet for motorcyclists or carrying an excessive load on trucks are examples of safety principles violations specific to each type of vehicle. Therefore, the detection of these types of violations should be based on vehicle-specific features. Despite numerous research efforts in the field of video surveillance of road transportation, the issue of automatically detecting compliance with transportation safety principles has not been specifically addressed, and a comprehensive solution for this problem has not been proposed. This research gap has motivated the current research.

In this paper, a comprehensive strategy for automatically detecting compliance with transportation safety principles through surveillance videos of highways is presented. The proposed method breaks down the research problem into three sub-problems: 1- Identifying moving objects in the scene, 2- Detecting the type of vehicle, and 3- Evaluating compliance with safety principles corresponding to the vehicle type. An extended Kalman filter (EKF) model is used to identify moving objects in the scene in order to solve the first sub-problem. A combination of deep learning and reinforcement learning models is used to solve the second and third sub-problems. A shared convolutional neural network (CNN) is used in the proposed method to identify the types of vehicles. Additionally, dedicated CNN models are used to detect safety violations specific to each type of vehicle. Both models use a combination of particle swarm optimization (PSO) and learning automata (LA) to determine the optimal configuration for each CNN model. The proposed combined optimization model and deep learning model optimization framework are innovative aspects of the current study and different from previous studies. The contribution of this paper can be summarized in the following main aspects:

- In this paper, the use of the EKF for tracking multiple targets in surveillance videos of highways has been examined, and the performance of this model in achieving real-time monitoring goals has been studied.
- A two-level strategy based on deep learning models for analyzing surveillance videos of highways is presented in this paper. At the first level, the type of vehicle is identified, and at the second level, compliance with safety principles for each vehicle is analyzed by a dedicated deep learning model.
- This study proposes a new strategy to optimize the configuration of deep learning models using a combination of reinforcement learning techniques and optimization. In this strategy, the use of the LA model in the PSO algorithm significantly increases the algorithm's convergence speed when determining the optimal configuration of the deep learning model.

The rest of the paper is organized as follows. The second section provides an overview of previous research. The third section provides details of the proposed method for real-time monitoring of highway videos. Furthermore, the fourth section evaluates and discusses the performance of the proposed method. Finally, the fifth section provides a summary of the results and conclusions.

2. Previous work

Video surveillance of highways can serve various purposes such as vehicle counting, traffic prediction, detection of law violations, or accident identification. In this section, some of the studies in this field are reviewed.

In [8], a machine vision-based method for vehicle identification and counting on highways using deep learning techniques is proposed. The approach first separates the highway region and then employs the YOLOv3 model for vehicle detection. Subsequently, the ORB algorithm is used to extract vehicle features and track targets, leading to the classification and counting of vehicles based on these features.

Research in Ref. [9] addresses three key challenges in real-time vehicle classification and tracking using deep learning: a lack of large training datasets, domain shifts, and integrating deep learning with multi-vehicle tracking algorithms. The approach proposed in this research improves classification accuracy by employing transfer learning and fine-tuning on a large dataset.

Research in Ref. [10] proposes a novel vehicle identification system using the Learning Vector Quantization (LVQ) algorithm. The system combines morphological feature extraction with the LVQ algorithm to achieve accuracy of 83%. This approach is computationally efficient due to its single-layer network architecture and competitive layers.

Research in Ref. [11] proposes an innovative vehicle classification system using a combination of pre-trained CNN model and PSO optimization. This model utilizes PSO for feature selection. The system outperforms existing methods in terms of accuracy, training time, and speed prediction.

Research in Ref. [12] presents an improved YOLO-based model for detecting and counting various types of vehicles in surveillance videos. To achieve this, a dataset containing various objects present on highways is collected, and ten different learning models are trained based on this dataset. Then, the model with higher accuracy is combined with YOLO, and vehicle classification is performed using this composite model. This combination significantly enhances YOLO's accuracy in accurately identifying various vehicle types.

Research [13] compares the performance of the tiny-YOLO model and Support Vector Machine (SVM) in detecting the type and color of vehicles. The accuracy of these two models is evaluated on different datasets. According to the results of this research, SVM outperforms tiny-YOLO in conditions of class imbalance.

[14] presents a model that uses artificial neural networks (ANN) to track moving vehicles on highways. In this method, the image

background is first identified by examining variable pixels in the image sequence. The moving target is then detected using an ANN optimized by the gravity search optimization (GSO) algorithm. In this approach, the weight vector values of the ANN model are adjusted using his GSO algorithm. In Ref. [15], a deep learning-based method for detecting motorcyclists wearing helmets in surveillance videos is proposed. Initially, moving objects in the video frame are identified using a background subtraction strategy. Then, a parametric classification approach is used to detect motorcycles among the set of moving objects. If a motorcycle is identified, the region of the rider's head is extracted from the frame, and after classification by a CNN model, the presence of a safety helmet is determined.

The research conducted in Ref. [16] introduces a multi-scale model for vehicle detection in surveillance videos. This approach aims to improve the performance of the YOLOv3 model in vehicle detection by adding additional convolutional layers to the feature pyramid network. To achieve this purpose, five two-dimensional convolutional layers are integrated into this model's characteristic pyramid network, which helps parse objects from largest to smallest. The study in Ref. [17] evaluates the use of CNN models to detect emergency vehicles in surveillance video images. The performance of eight CNN models was compared on this task, where DenseNet121 outperformed the other models. The research carried out in Ref. [18] simultaneously addresses two tasks: vehicle re-identification and anomaly detection. For solving the first task, a combination of high-level vehicle features and local manual features is employed. Additionally, multiple components for vehicle detection and anomaly detection are utilized for addressing the second task. In Ref. [19], the performance of two models, YOLOv3 and YOLOv4, is compared for vehicle detection in highway surveillance videos. The results indicate the superiority of the YOLOv4 model in terms of accuracy and processing speed.

Research in Ref. [20], used Yolo v2 for detecting helmet violation. In a similar research [21], a real-time helmet violation detection system for motorcyclists using an R–CNN deep learning model was presented. The system is based on surveillance videos captured by roadside cameras and can detect and capture motorbikers without a helmet with 97.69% accuracy. Research in Ref. [22] proposes a deep-learning-based model that automatically adapts to camera rotation, enabling statewide surveillance and reducing wrong way driving fatalities.

Research in Ref. [23] proposed an automated system to identify traffic rule violators from images. The system uses YOLOv3 and other deep learning techniques to detect motorcycles, helmets, triple-seat riders, and number plates. In Ref. [24], A system that combines GPS tracking and road monitoring data has been developed to automatically detect non-compliant truck driver behavior in real time. This system allows for targeted enforcement action against non-compliant drivers without disrupting compliant drivers.

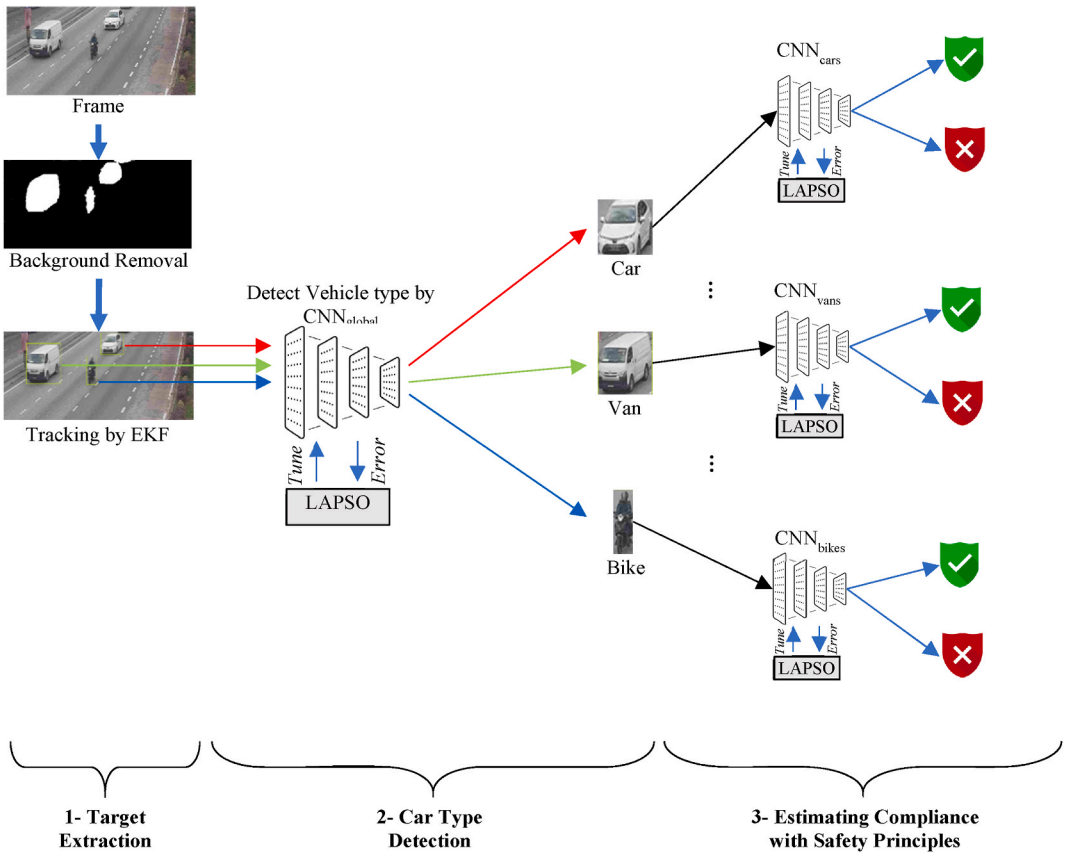

Fig. 1. Diagram of the proposed method stages.

3. Research methodology

In this section, after describing the technical specifications of the data used in the study, we present details of the proposed method to improve roadside video surveillance using a combination of EKF and deep reinforcement learning.

3.1. Data

The required data for implementing the proposed method in this article has been collected through highway surveillance cameras available online through https://141.ir/website. This database includes 60 surveillance videos with a minimum duration of 1 min, showing the movement of vehicles at different times of the day and night. Each video belongs to a different location. In order to address privacy concerns, none of the videos contain faces or license plate numbers. The database videos include traffic of interstate and state highways which have been recorded by infrared and regular CCTV dome cameras. The total number of vehicles in this dataset is 5246, which are labeled into 7 different categories: 1- Cars (sedan/SUV/coupe) (2572 samples), 2- Trucks (918 samples), 3- Motorcycles (105 samples), 4- Pickups (645 samples), 5- Vans (367 samples), 6- Buses (262 samples), and 7- Others (377 samples). The path of each vehicle's movement in each video is marked, and the type of vehicle is specified as a label. Furthermore, the compliance with transportation safety principles for trucks and motorcycles is categorized into two groups: 1- Proper and 2- Improper labeling. It should be noted that due to the imbalance in the target classes regarding the adherence to safety principles for other categories, labeling this condition for other types of vehicles has been disregarded.

3.2. Proposed method

To achieve an efficient and automated video surveillance system, it is necessary to perform processes of identifying moving targets and determining their types by utilizing suitable and optimized techniques. The proposed method in this study combines EKF and Reinforcement Deep Learning to meet these requirements. This strategy can be broken down into the following detailed steps:

1. Background subtraction and detection of multiple targets based on EKF.
2. Vehicle type detection based on the combination of CNN and Reinforcement Learning.
3. Evaluation of safety compliance in each vehicle category based on the combination of CNN and Reinforcement Learning.

The architecture of the proposed model is depicted in Fig. 1.

According to the diagram presented in Fig. 1, the proposed method begins by identifying the background region in the surveillance video scene, and moving regions are identified by subtracting pixels from the background. The foreground regions in the frame are processed using the EKF model to detect multiple targets. Then, the EKF model is used to track the target in the video frame sequence. After identifying moving targets in the frame, the region of each frame is processed separately to determine the type of the vehicle. This action is performed in the second step of the proposed method using the CNN model. After labeling each moving object in the frame, the third step of the proposed method is to evaluate the safety compliance. Following the structure shown in Fig. 1, a separate CNN model is used for each vehicle type in this step. Each CNN model in this step has a similar architecture in its layers but differs in the configuration of its parameters. The configuration of each layer in the CNN model used for vehicle type detection, as well as the CNN models corresponding to each type of vehicle, is adjusted using a new strategy based on a combination of Reinforcement Learning and the PSO algorithm. This optimization strategy, called LAPSO, assigns an LA model to each optimization variable, each of which attempts to discover the optimal strategy for configuring one of the CNN model parameters. The LAPSO algorithm, by combining the strategies determined by these LA models, can discover the optimal configuration for the CNN model in fewer iterations.

3.2.1. Background subtraction and multi-target identification based on EKF

The proposed method identifies the background region in surveillance camera footage. Since the road area remains relatively static, it can be extracted and subtracted from subsequent frames to estimate moving objects. To eliminate the impact of stationary objects, the result of background subtraction is converted into a binary image, with moving regions set to 1 and background regions set to 0. The binary matrix is multiplied with the input frame to create an image with a black background and a foreground containing moving objects.

After extracting the foreground region of the frame, the EKF model is used to track the foreground regions of these frames. Suppose there are M targets (continuous regions in the foreground sequence) under tracking. In this case, the maneuverability of the target must be considered. In the proposed method, the EKF model has been used to effectively track the foreground regions of these frames. Let's assume that there are J models in the Extended Interacting Multiple Model (IMM) Kalman Filter, and the motion of the target in model j is as equation (1) [25]:

$$x_i^j(t_k) = F_i^j(T_{k-1})x_i^j(t_{k-1}) + \Gamma_i^j(T_{k-1})w_i^j(t_{k-1}), j = 1, 2, \ldots, J \tag{1}$$

In this context, the vector $x_i^j(t_k) = [x_i^j(t_k), \dot{x}_i^j(t_k), \ddot{x}_i^j(t_k), y_i^j(t_k), \dot{y}_i^j(t_k), \ddot{y}_i^j(t_k)]^T$ represents the position, velocity, and acceleration of target i at time t_k. Additionally, $F_i^j(T_{k-1})$ denotes the state transition matrix of model j at time T_{k-1} where T_{k-1} refers to the time index between t_{k-1} and t_k. $w_i^j(t_{k-1})$ specifies the noise in model j and is described as a variable with a Gaussian distribution, mean zero, and covariance $Q_i^j(t_{k-1})$. Finally, $\Gamma_i^j(T_{k-1})$ represents the process noise input matrix in model j, and J denotes the number of models in the Extended

Interacting Multiple Model (IMM) Kalman Filter.

The IMM-EKF is an iterative method that minimizes the mean square error [26]. It is particularly well-suited for target tracking because it considers both observation noise and process noise [27]. This makes it more accurate than other tracking methods, especially when dealing with nonlinear observation equations. The observation equations for IMM-EKF typically involve estimating the range and tracker angle for each target.

In the first step of the Extended Kalman Filter (EKF) multi-model interactive process, the initial initialization is for the target's motion vector is considered as $x_j(0) = [x_j(0), \dot{x}_j(0), \ddot{x}_j(0), y_i(0), \dot{y}_i(0), \ddot{y}_i(0)]^T$, and also $R(0) = 10\,I$. Where, the vector $[x_j(0), \dot{x}_j(0), \ddot{x}_j(0)]$ represents the initial position, velocity, and acceleration of the target in the horizontal dimension, and $[y_i(0), \dot{y}_i(0), \ddot{y}_i(0)]$ specifies the same values in the vertical dimension. Additionally, I is a diagonal square matrix and is considered as the initial value of the error covariance matrix R in this context. The next step in the EKF multi-model interactive process is an iterative process that repeats N_{sample} times. In each iteration of this process, the target's new position vector and error covariance matrix are updated using the following equations:

$$\hat{x}_i(t_k|t_{k-1}) = J.\hat{x}_i\left(t_{(k-1)}\middle|t_{(k-1)}\right) \tag{2}$$

$$R(t_k|t_{k-1}) = J.R\left(t_{(k-1)}\middle|t_{(k-1)}\right).J^T + Q \tag{3}$$

In the equations above, $\hat{x}_i(t_k|t_{k-1})$ represents the estimate of the target's position, velocity, and acceleration at the new sampling time t_k based on its previous state at time $t_{(k-1)}$. J is a Jacobian matrix for the coordinate transformation from polar to Cartesian coordinates. Additionally, $R(t_k|t_{k-1})$ is the error covariance matrix at time t_k based on the previous period's error covariance matrix, $R(t_{(k-1)}|t_{(k-1)})$. Q is the covariance matrix of the noise with a normal distribution and zero mean.

$$K(t_k) = R(t_k|t_{k-1}).H^T(t_k)\left(C + H(t_k).R(t_k|t_{k-1}).H^T(t_k)\right)^{-1} \tag{4}$$

In the equation above, the error covariance matrix $R(t_k|t_{k-1})$ is calculated using equation (3), and C represents the diagonal matrix of the target's position estimation error. Additionally, $H(t_k)$ represents the measurement matrix for the range measurement of the target's position at the sampling time t_k, which will be discussed further. At the end of each sampling period, the target's position and the corresponding adaptation matrix for it are calculated for use in the next sampling period.

$$\hat{x}_i(t_k|t_k) = \hat{x}_i(t_k|t_{k-1}) + K(t_k)(p(t_k) - h(\hat{x}_i(t_k|t_{k-1}))) \tag{5}$$

$$R(t_k|t_k) = (1 - K(t_k)H(t_k))R(t_k|t_{k-1}) \tag{6}$$

In the above relationship, $\hat{x}_i(t_k|t_{k-1})$ represents the predicted position, velocity, and acceleration of the target at the new sampling time t_k based on its previous state at time $t_{(k-1)}$, which is calculated using equation (2). Additionally, the adjustment matrix $R(t_k|t_{k-1})$ and the error correction matrix $K(t_k)$ are calculated through equations (3) and (4) respectively. Furthermore, $p(t_k)$ represents the observed state at the current sampling time. In the developed multi-object interaction modeling process, the calculations from equations (2)–(6) will be repeated N_{sample} times, where N_{sample} is the number of samples. During these computations, the measurement matrix H can be calculated using equation (7):

$$H(t_k) = \frac{\partial h}{\partial x(t_k)}\Big| x(t_k) = \hat{x}(t_k|t_{k-1}) = \begin{bmatrix} \frac{x_i(t_k) - x_1^r}{\sqrt{(x_i(t_k) - x_1^r)^2 + (y_i(t_k) - y_1^r)^2}} & \frac{y_i(t_k) - y_1^r}{\sqrt{(x_i(t_k) - x_1^r)^2 + (y_i(t_k) - y_1^r)^2}} & 0 & 0 \\ \frac{-(y_i(t_k) - y_1^r)}{\sqrt{(x_i(t_k) - x_1^r)^2 + (y_i(t_k) - y_1^r)^2}} & \frac{x_i(t_k) - x_1^r}{\sqrt{(x_i(t_k) - x_1^r)^2 + (y_i(t_k) - y_1^r)^2}} & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \frac{x_i(t_k) - x_N^r}{\sqrt{(x_i(t_k) - x_N^r)^2 + (y_i(t_k) - y_N^r)^2}} & \frac{y_i(t_k) - y_N^r}{\sqrt{(x_i(t_k) - x_N^r)^2 + (y_i(t_k) - y_N^r)^2}} & 0 & 0 \\ \frac{-(y_i(t_k) - y_N^r)}{\sqrt{(x_i(t_k) - x_N^r)^2 + (y_i(t_k) - y_N^r)^2}} & \frac{x_i(t_k) - x_N^r}{\sqrt{(x_i(t_k) - x_N^r)^2 + (y_i(t_k) - y_N^r)^2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

The region of each tracked target is extracted from the sequence of input frames, and each of these regions is used as input in the second step of the proposed method to identify the type of vehicle corresponding to each target area using a CNN model configured by reinforcement learning strategy.

3.2.2. Detection of vehicle types based on a combination of CNN and reinforcement learning

The result of the first step is to extract a set of regions corresponding to the moving target in the input frame. The second step of the proposed method is to determine the car model corresponding to each of these regions. This involves resizing each region to 80x80 pixels and using a CNN model to classify each region. The proposed basic CNN model structure for vehicle type recognition is shown in Fig. 2.

According to the structure shown in Fig. 2, the proposed CNN model consists of five two-dimensional convolutional layers, each layer followed by a batch normalization layer (BN) and ReLU as the activation function, and a pooling layer. The first two-dimensional convolutional layer is responsible for learning specific patterns in the input frame, and subsequent convolutional layers focus on learning more general patterns in the image. After these five convolution layers, two fully connected layers (FC1 and FC2) are considered in the proposed CNN model. The first fully connected layer (FC1) provides feature maps describing the input in vector form, and the second fully connected layer (FC2) is used to transfer information to the classification layer and its size is equivalent to the number of target classes (vehicle types). Finally, SoftMax layers and classification are employed for vehicle type recognition. In the proposed CNN model, the stride size is set to 1 in all dimensions for convolution and pooling layers. Additionally, the dimensions of the pooling layers are adjusted based on the dimensions of the convolution layers preceding them. It's worth noting that this model includes an extra category for detecting non-vehicle objects (such as signs, pedestrians, etc.) to identify various moving objects present in the input frames.

The precise configuration of hyperparameters for a CNN model is of great importance. On the other hand, a deep CNN model has more parameters and complexities compared to initial artificial neural network models, making the configuration of these hyperparameters a complex and time-consuming process. Therefore, this research uses a novel algorithm based on a combination of Reinforcement Learning and PSO to tune the hyperparameters of the CNN model. The following section describes the process of tuning the hyperparameters of the CNN model using the proposed LAPSO algorithm. In the proposed CNN model, there are three sets of configurable hyperparameters:

1. *Hyperparameters of convolution layers C1 to C5*: This set of hyperparameters includes the length and width of the convolution filter, as well as the number of filters. To limit the search space, the hyperparameters for the length and width of the filter are considered equal. This is because, based on experimental results, using convolution filters with the same length and width can yield satisfactory results. Depending on the dimensions of the input samples, the hyperparameters for the length and width of the filter can be assigned natural numbers within the range [3,18]. Additionally, the hyperparameter for the number of filters is adjustable and can take a natural number value within the range [8, 128], with a step size of 8.
2. *Type of pooling function P1 to P5*: Each of the pooling layers P1 to P5 can choose one of the functions, either max or average.
3. *Size of the fully connected layer FC1*: The dimensions of this layer can be described as a natural number within the range [30, 100].

The number of different combinations to initialize the above hyperparameters is very large, so determining the optimal configuration between these states will take time. The proposed method uses a combination of reinforcement learning model and PSO algorithm to tune the hyperparameters of the CNN model. In this strategy, the PSO algorithm uses reinforcement learning models to store suitable configurations for each layer of the CNN model. In other words, the control over how to configure each of the CNN hyperparameters is delegated to a reinforcement learning model to identify configurations that lead to a reduction in the model's error through the examination of the impact of parameter changes on training error. The LAPSO algorithm, by combining the optimization strategies of each reinforcement learning model, attempts to create a suitable configuration for the CNN. Below, we first describe the particle structure in LAPSO, and after explaining the fit evaluation method, we present the steps for discovering the optimal configuration using this algorithm.

The number of tunable parameters (number of optimization variables in LAPSO) is 16. Therefore, the length of each particle in the LAPSO CNN construction problem is written as a numerical vector of length 16. For each particle vector, 10 variables are used to tune

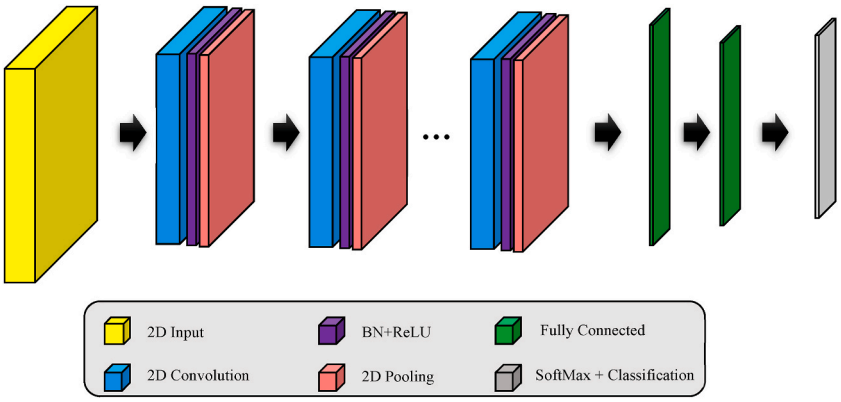

Fig. 2. Proposed CNN model structure for recognizing vehicle type.

the hyperparameters of the 5 convolutional layers. In this set, five variables refer to the number of filters in each convolutional layer, and five variables determine the filter dimensions. Additionally, five variables determine the type of pooling function and can take either the values 0 = maximum or 1 = average. The last variable for each particle indicates the size of the fully connected layer FC1. To evaluate the suitability of each particle, the configuration determined by the particle is first applied to the CNN model. Then, the training process of the constructed CNN model is performed based on 25% of the training samples, and finally the fitness of the particles is evaluated using the validation error metric as equation (8).

$$Fitness = \frac{V}{N} \tag{8}$$

In the above equation, N shows the total number of validation samples and V represents the number of samples for which the predicted label deviates from its ground truth label. The goal of the LAPSO algorithm in the proposed method is to achieve a CNN configuration that minimizes the fitness function. To expedite the convergence of search algorithms, you can leverage the search history to construct new solutions. In LAPSO, reinforcement learning models are used to generate random responses that are close to the global optimum. The combination of the PSO algorithm with reinforcement learning aims to discover the global optimum solution more quickly, which is an innovation introduced in the proposed method. The proposed method uses reinforcement learning units to improve the convergence speed of optimization. The reinforcement learning model used in the proposed method uses reward and penalty mechanisms to explore new optimal configuration strategies. The structure of the LA model is shown in Fig. 3.

Each LA model is defined through a set of selectable actions. The set of actions is denoted as $A = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. Each action in A has a probability value, and the selection of each action is based on these probabilities. The LA model initiates its activity by selecting an action from the set of actions A and applying it to the environment. The action taken is evaluated by the environment, and the LA model selects its next action based on the response received from the environment. During each action selection, if the response from the environment is desirable, the LA model increases the probability associated with the selected action. Conversely, if an undesirable response is received from the environment, the probability of that action decreases.

During this process, the LA learn which actions are optimal by adjusting the probabilities of actions based on reward (increase in the probability of the selected action) and penalty (decrease in the probability of the selected action) operators. They learn which actions should be chosen more frequently in subsequent cycles. In the proposed method, 16 reinforcement learning models are defined to describe configuration strategies for the adjustable hyperparameters in CNN. For each adjustable hyperparameter, a LA is defined, where each one has M selectable actions (M represents the number of possible choices for that hyperparameter). The goal of each LA is to provide an approximation of the optimal value for the corresponding hyperparameter.

In the first iteration of the LAPSO algorithm, all actions have equal probabilities ($\frac{1}{M}$). With this reinforcement learning-based structure in mind, the optimization algorithm generates half of the responses based on the probabilities determined by the LA. This approach allows the algorithm to maintain the probability of fully exploring the problem space and, thus, prevent the algorithm from getting stuck in local minima.

At the end of each LAPSO search cycle, the best and worst solutions from the population are used to update the structure of the LAs. First, the best and worst solutions in the current population are identified. Then, the probabilities of all actions corresponding to the best solution in the population are increased in the LA models using the reward operator. This operation is performed for each reinforcement learning model using equation (9) [28]:

$$p_j(k+1) = \begin{cases} p_j(k) + a\left[1 - p_j(k)\right] & j = i, \\ (1-a)p_j(k) & \forall j \neq i. \end{cases} \tag{9}$$

Where, a represents the penalty coefficient and is set to 0.5. $p_j(k)$ denotes the probability of selecting action j in the k_{th} iteration for the LA. This operation is performed for all the LAs corresponding to the CNN hyperparameters. Then, the probabilities of all actions corresponding to the worst solution in the current population are decreased in the LA models using the penalty operator. This operation is performed for each LA model using equation (10) [28]:

$$p_j(k+1) = \begin{cases} (1-b)p_j(k) & j = i, \\ \left(\frac{b}{M-1}\right) + (1-b)p_j(k) & \forall j \neq i. \end{cases} \tag{10}$$

In the equation mentioned, b represents the penalty coefficient and is set to 0.5. Also, M denotes the total number of actions of the LA. Given the information provided, the steps of optimizing configurations by LAPSO will be as follows:

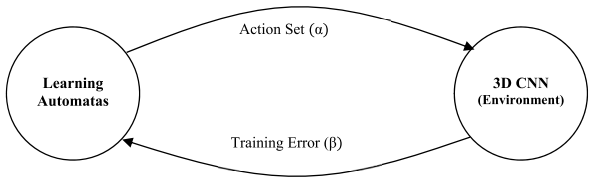

Fig. 3. Functioning of LA models for tuning proposed CNN hyperparameters.

Step 1. Determine the search parameters in LAPSO, including the population size *P* and the number of iterations *G*.

Step 2. Initialize the initial population of particles and their velocities randomly.

Step 3. Calculate the fitness for each particle.

Step 4. If a particle's fitness does not improve after *T* iterations, then with a probability of 0.5, update the particle's configuration based on the optimal actions defined in the set of LAs and reevaluate the fitness of the new particle.

Step 5. Determine the values of the best individual particle (pbest) and the best global particle (gbest) in the population.

Step 6. Update the velocity of each particle based on the following equation [29]:

$$v_i^{t+1} = v_i^t + c_1 r_1 \left(pbest_i^t - p_i^t\right) + c_2 r_2 \left(gbest_i^t - p_i^t\right) \tag{11}$$

In equation (11), v_i^t represents the current velocity of particle i in the current iteration, and $pbest_i^t$ and $gbest_i^t$ represent the best individual and global positions for particle i in the current iteration, respectively. The coefficients c_1 and c_2 are the individual and social acceleration factors, respectively. Finally, r_1 and r_2 are random values that simulate stochastic effects and the social behavior of particles.

Step 7. With the updated velocity for each particle, the position of each particle is updated using equation (12) [29]:

$$p_i^{t+1} = p_i^t + v_i^{t+1} \tag{12}$$

Step 8. If the fitness reaches zero or the number of algorithm iterations equals G, step 9 will be executed; otherwise, the algorithm will repeat from step 3.

Step 9. The gbest (global best) particle is returned as the optimal solution.

After completing the above steps, the radius values found for the optimal particles are applied to the CNN model, and this constructed model is trained on all samples. The resulting trained model is used to recognize vehicle types.

3.2.3. Evaluation of safety compliance in each vehicle category

After determining the vehicle type using the CNN model described in the previous section, we use specific CNN models for different vehicle categories to assess compliance with safety principles. To do this, if a moving object in the input frame belongs to a vehicle category such as *A*, the object's region of interest is extracted from the initial frame and is used as input to the CNN_A model. Each CNN model corresponds to a specific vehicle category and is trained solely based on samples related to vehicles in that category. This CNN model, through processing the vehicle's image, categorizes it as safe or unsafe. As mentioned earlier, the reason for using dedicated CNN models for each vehicle category is the difference in safety regulations definitions for each vehicle category. This mechanism allows for more precise detection of safety regulation violations. It's worth noting that the structure of each CNN model used for different vehicle categories is similar to the one described in step 2 (Fig. 2). Each of these CNN models is configured separately using the proposed LAPSO algorithm. The optimization procedure employed in this step is similar to the one discussed in the previous step, and repetition is omitted.

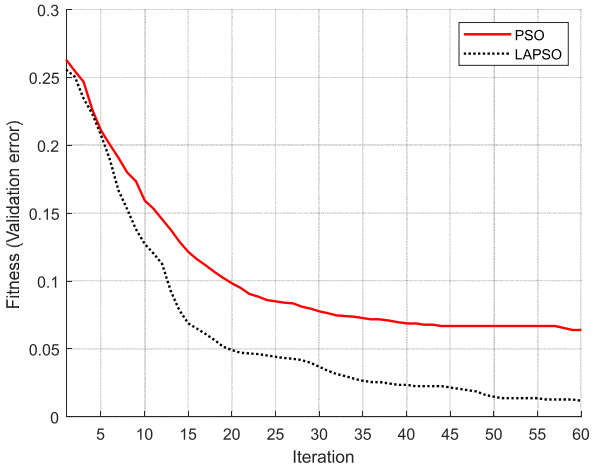

Fig. 4. Changes in the best LAPSO fitness compared to PSO for CNN model Configuration.

4. Implementation and evaluation

This section is devoted to discussing the results obtained by implementing the proposed method and comparing its performance with previous studies. The proposed method was implemented using his MATLAB 2021a software, and the database example described in 3–1 was used as evaluation data. During the experiments, metrics such as accuracy, precision, recall, and F-measure were used to evaluate the performance of the proposed method in vehicle type detection and safety violation detection. It should be noted that in the second case, the focus was solely on detecting the lack of compliance with safety principles in the truck category. This was because the samples related to compliance with safety principles in other vehicle categories had a high class imbalance issue, and training dedicated learning models for these vehicle categories led to overfitting issues.

To configure the proposed CNN models in the phases of vehicle type detection and safety regulation violation detection, the LAPSO algorithm was used. In each of these cases, parameters including the number of particles and the number of LAPSO iterations were set to 100 and 300, respectively. Additionally, the threshold parameter for the number of iterations without change for each particle was set to 5. Fig. 4 presents a graph showing the changes in the best fitness in different iterations of the LAPSO algorithm.

To evaluate the effectiveness of the reinforcement learning strategy in detecting more efficient topologies, the results obtained were compared with the simple PSO algorithm. Both of these algorithms were evaluated based on the same parameters and data.

As Fig. 4 illustrates, the proposed LAPSO algorithm can perform more successfully in discovering an optimal configuration for the CNN model. These results indicate that in the first 5 iterations, the PSO and LAPSO strategies have nearly similar performance. However, as the search progresses, the LAPSO strategy can generate more suitable combinations for the CNN model by using actions stored in LA models. Furthermore, as the search iterations advance, the difference between PSO and LAPSO becomes more significant, and the reason behind this is that the LA models achieve a more precise understanding of the environment (the CNN model's response to various configuration changes). These results demonstrate the LAPSO algorithm's notable efficiency in quickly discovering optimal solutions. It can be predicted that this superiority is not limited to configuring CNN models and can yield similar results in solving other problems.

In Table 1, an example of the optimal configuration discovered by the LAPSO algorithm for each of the CNN models used in the proposed method is given. As the information in Table 1 shows, the two CNN models used in the proposed method are configured based on the same principles, but have differences in terms of configuration details. In both CNN models, as the convolution layers advance, the dimensions of the filters decrease and their number increases. Because in the deeper layers of each CNN model, the complexity of the patterns in the feature maps increases, and in order to more fully extract the patterns related to vehicle type/safety violation, it is necessary to increase the diversity of feature map combinations. On the other hand, in both CNN models, the final pooling layers are determined as average and the initial layers are of max pooling type. On the other hand, each convolution layer in the CNN model used for detecting compliance with safety principles, generally includes more convolution filters with smaller dimensions compared to the other CNN. This feature can indicate the higher complexity of the patterns related to this sub problem, compared to the complexity of the features related to vehicle type.

When implementing the proposed method at each stage of vehicle type detection and safety violation detection, 10-fold cross validation was used for evaluation. In each iteration, the labels predicted by the proposed algorithm were compared with the actual labels of the test samples, and based on this, precision, precision, recall, and F-measure metrics were calculated. Since precision and recall metrics are used to evaluate binary classification tasks, for vehicle type recognition these metrics are computed separately for each class and the average value obtained across all classes is considered. In this case, to calculate these metrics for each class, the current class was considered as positive, and other classes were considered as negative. The precision metric represents the algorithm's accuracy in classifying samples of each class separately. On the other hand, the recall metric indicates what proportion of true positives there are. The F-Measure metric represents the harmonic mean of the accuracy and recall metrics. These metrics can be calculated based on the following equations [30]:

$$Precision = 100 \times \frac{TP}{TP + FP} \tag{13}$$

Table 1
An example of the optimal configuration discovered by the LAPSO for the proposed CNNs.

Layer	CNN (vehicle type)	CNN (safety principles)
$Convolution_1$ (W × L,N)	18 × 18,16	16 × 16,16
$Pooling_1$	Max Pooling	Max Pooling
$Convolution_2$ (W × L,N)	13 × 13,24	13 × 13,40
$Pooling_2$	Max Pooling	Max Pooling
$Convolution_3$ (W × L,N)	11 × 11,24	11 × 11,48
$Pooling_3$	Max Pooling	Max Pooling
$Convolution_4$ (W × L,N)	9 × 9,32	7 × 7,64
$Pooling_4$	Average Pooling	Average Pooling
$Convolution_5$ (W × L,N)	6 × 6,48	4 × 4,88
$Pooling_5$	Average Pooling	Average Pooling
FC_1	68	83

$$Recall = 100 \times \frac{TP}{TP + FN} \tag{14}$$

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precison + Recall} \tag{15}$$

In equation (13), TP represents the number of correctly identified true positive samples. Also, in equation (14) FN indicates the number of positive samples incorrectly classified into other (negative) categories. Additionally, FP represents the number of negative samples incorrectly assigned to the positive category. Finally, F-measure criterion in equation (15) is calculated using precision and recall criteria. For the implementation of the proposed method, MATLAB 2021a software has been used. To evaluate the effectiveness of each technique used in the proposed method in the vehicle type detection phase, its performance has been compared with different scenarios. The first compared scenario, referred to as "KF + CNN," involves replacing the EKF model used in the proposed method with a basic KF model. The purpose of comparing the proposed method with this scenario is to investigate the impact of the EKF model on correctly detecting moving targets. The second compared scenario, referred to as "static CNN," indicates conditions in which the CNN model configuration step by the proposed LAPSO algorithm is ignored. By comparing the performance of the proposed method with this scenario, the effect of LAPSO on improving the accuracy of the CNN model can be assessed. Additionally, results obtained using the methods presented in Refs. [9,11], and [12] were also compared. Furthermore, in the safety compliance detection phase, the performance of the proposed method was compared with the "static CNN" scenario and the condition of predicting the target variable using his ResNet and VGG-16 models. It is worth noting that all these models were trained and tested on the same data. In Fig. 5, the average accuracy of the proposed method in two scenarios, vehicle type detection, and safety regulation compliance detection, is presented. It's worth noting that the results presented in this figure and other graphs in this section are the result of aggregating the results of 10 repetitions of experiments.

According to Fig. 5a, the proposed method can detect various types of vehicles with an accuracy of 98.72%, and it outperforms the compared methods in this aspect. On the other hand, if the EKF model used in the proposed method for target identification is replaced with a basic KF model, the detection accuracy will decrease to 93.5%. Furthermore, if the configuration strategy of the CNN model based on LAPSO is omitted, vehicle type detection will be performed with an accuracy of 94.57%.

These comparisons indicate the following:

- Firstly, the use of the proposed EKF model can lead to more successful performance compared to the basic KF model and increase the detection accuracy by at least 5.22%.
- Secondly, comparing the performance of the LAPSO-configured model with the static CNN model shows that this approach can accurately detect vehicle types and increase the accuracy of vehicle type detection by at least 4.15%. These results demonstrate that the method used in the proposed method is effective in improving the accuracy of vehicle type recognition. Based on this experiment, the method presented in Ref. [12] has the closest performance to the proposed method, with the techniques used in the proposed method leading to a 3.41% increase in accuracy compared to this model.

On the other hand, Fig. 5b illustrates that the proposed LAPSO strategy for configuring each CNN model dedicated to vehicle types can increase the accuracy of safety principles detection by 5.12%. Additionally, the CNN model configurations specific to each vehicle type outperform well-known models such as ResNet and VGG-16. This is because these models are often suitable for problems with a large number of classes, and for binary classification tasks like safety principles detection, simpler models are more appropriate. In this scenario, the ResNet model, despite having a 3.48% higher error, shows the closest performance to the proposed method.

Confusion matrices can provide more detailed insights into how classification methods perform in identifying vehicle types. In

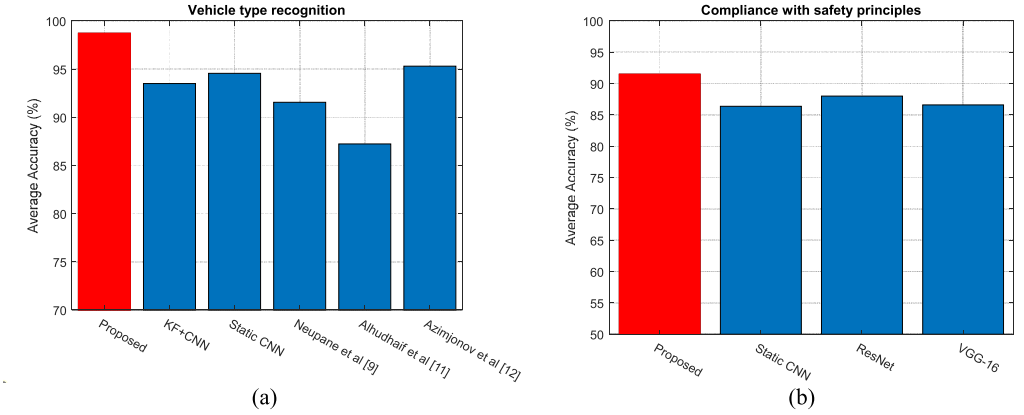

Fig. 5. Average accuracy of the proposed method and other approaches in two scenarios: (a) Vehicle type recognition and (b) compliance with safety principles.

Fig. 6, the confusion matrix for the proposed method and other approaches in the vehicle type identification phase is presented. It's worth noting that in this experiment, slices corresponding to each vehicle image in the video samples of the database were used as input for each method. Furthermore, in this case, only one frame of each unique vehicle in the database was utilized.

Each column of the confusion matrix represents the actual labels of the test samples (the actual vehicle types), and the rows of the matrix represent the labeling of the samples by each classification method (the predicted type by the model). For example, in Fig. 6a, out of 2572 test samples of sedan/suv/coupe cars (the sum of values in the first column of the matrix), 2538 samples are correctly classified by the proposed method, and only 34 samples are misclassified into other categories. Consequently, the proposed model identifies sedan/suv/coupe cars with an accuracy of 98.68%, which demonstrates a significant superiority compared to the other methods depicted in Fig. 6b to d. On the other hand, the sum of the values in the first row of Fig. 6a shows that the proposed method classifies 2546 samples into Sedan/SUV/Coupe category and the model correctly classifies 99 samples considering 2538 correct outputs. This indicates that it was possible to call. 69% of the sample falls into this category. The interpretation of classification results for other categories is possible in a similar manner. By examining these results, it can be understood that the superiority in precision and recall of the proposed method is not limited to the sedan/suv/coupe car class alone, and this model can also classify vehicles belonging to other categories with higher precision and recall. Overall, comparison of these confusion matrices shows that the proposed method has advantages in classifying samples of most diseases compared to other methods and increases the accuracy in detecting diseases up to at least 3.41%.

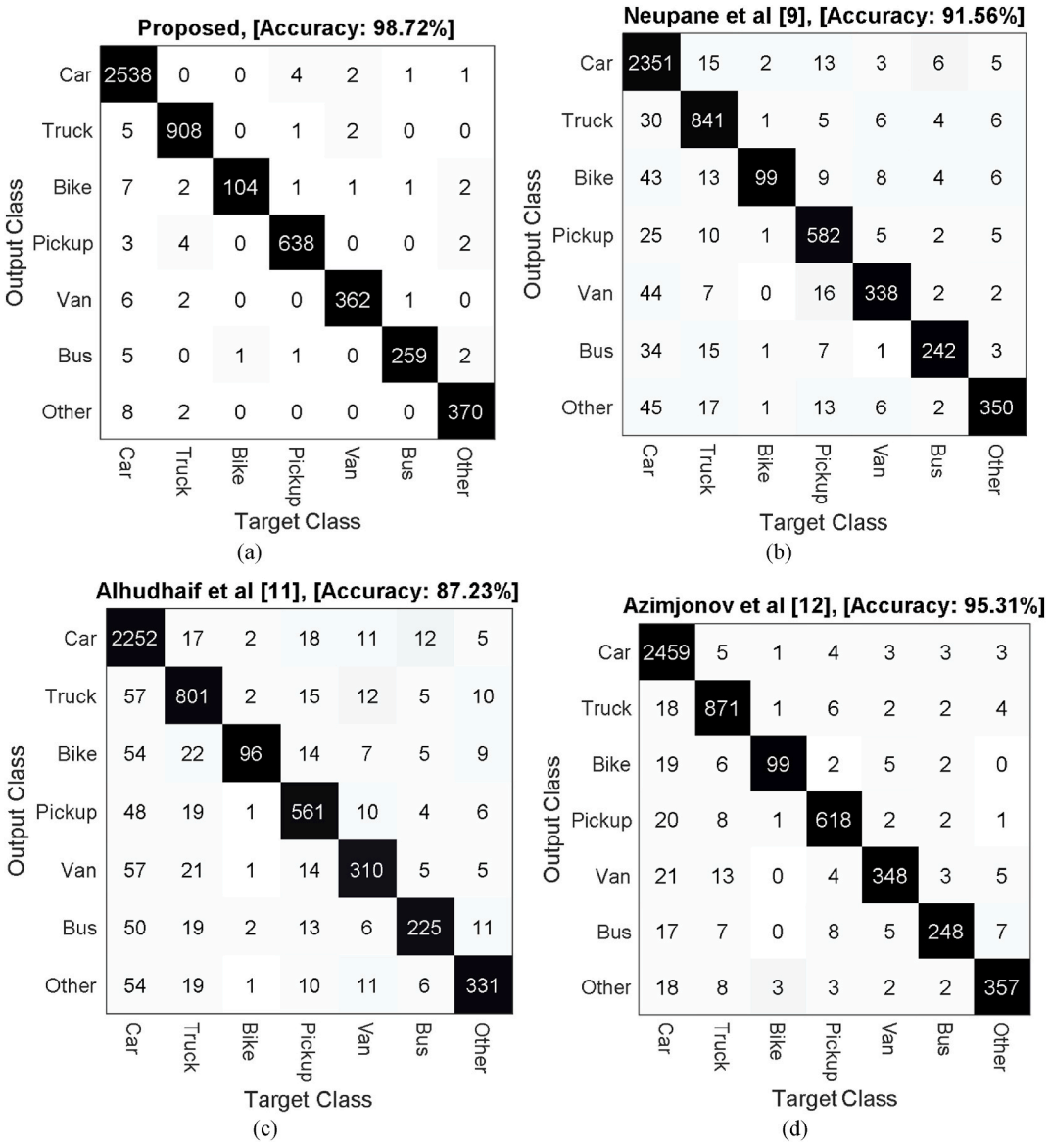

Fig. 6. Confusion matrix of different methods in the vehicle type identification phase.

In Fig. 7, the confusion matrix of the proposed method and other methods in the safety principles detection phase has been compared. In the matrices plotted in Fig. 7, the first row/column represents the safe category, and the second row/column represents the unsafe category, indicating compliance with safety principles in vehicle transport. It's worth noting that the results presented in Fig. 7 only pertain to the truck category. Comparing Fig. 7a with Fig. 7b to d shows that in the safety principles detection phase, the proposed model also exhibits superior performance compared to the other methods. This superiority is observable in the classification of samples in both the safe and unsafe categories.

Fig. 8 compares the performance of different vehicle type detection methods based on precision, recall, and F-measure metrics, broken down by category in Fig. 8a to c, and the average of the numbers. This data across all category targets is presented in Fig. 8d. In each graph shown in Fig. 8a to c, the first dimension represents the categories associated with the media type and the second dimension corresponds to the methods being compared. By examining these plots, it can be observed that the proposed method, using a combination of EKF and LAPSO, can classify different categories with higher efficiency compared to other methods. Fig. 8d shows the average of the precision, recall, and F-Measure metrics calculated for all classes. This chart represents the overall performance of different methods concerning classification quality.

Fig. 9 shows the precision, recall, and F-measure metrics for different methods during the security compliance detection phase. Figs. 8d and 9 show that the proposed CNN model, which uses a reinforcement learning-based configuration strategy, can perform better in accurate target detection for both evaluation phases compared to other methods. Additionally, the numerical results obtained from the experiments in this section are provided in Table 2.

Comparing the accuracy, precision, recall and F-measure metrics in Table 2, Figs. 8 and 9 confirms that the proposed method can achieve higher quality in both method type recognition Convenience and security compliance detection compared to other methods. These results show that the proposed method can lead to improvements in accuracy, precision, recall, and F-Measure metrics. The higher accuracy of the proposed method suggests that the results generated for each vehicle type/safety compliance are more likely to be accurate than other methods. Furthermore, the higher recall demonstrates that the proposed method can correctly identify a higher proportion of samples belonging to the target category. Fig. 10a and b, show the ROC curves of different methods for vehicle type detection phase and safety compliance detection phase, respectively. According to these graphs, the proposed method has higher true positive rate and lower false positive rate in both scenarios, and the area under his ROC curve of the proposed method is lower than that of the comparative method in both security compliance detection and vehicle type detection. It can be concluded that the method

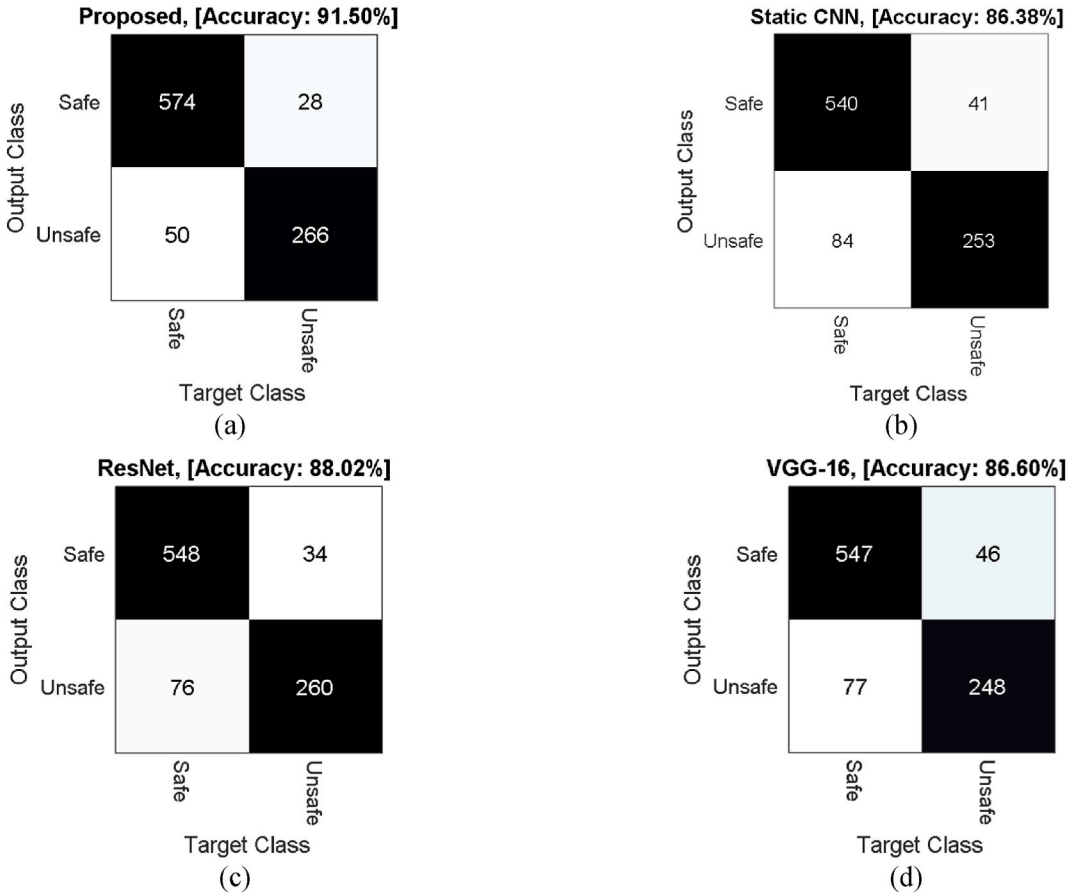

Fig. 7. Confusion matrix of different methods in the safety principles detection phase.

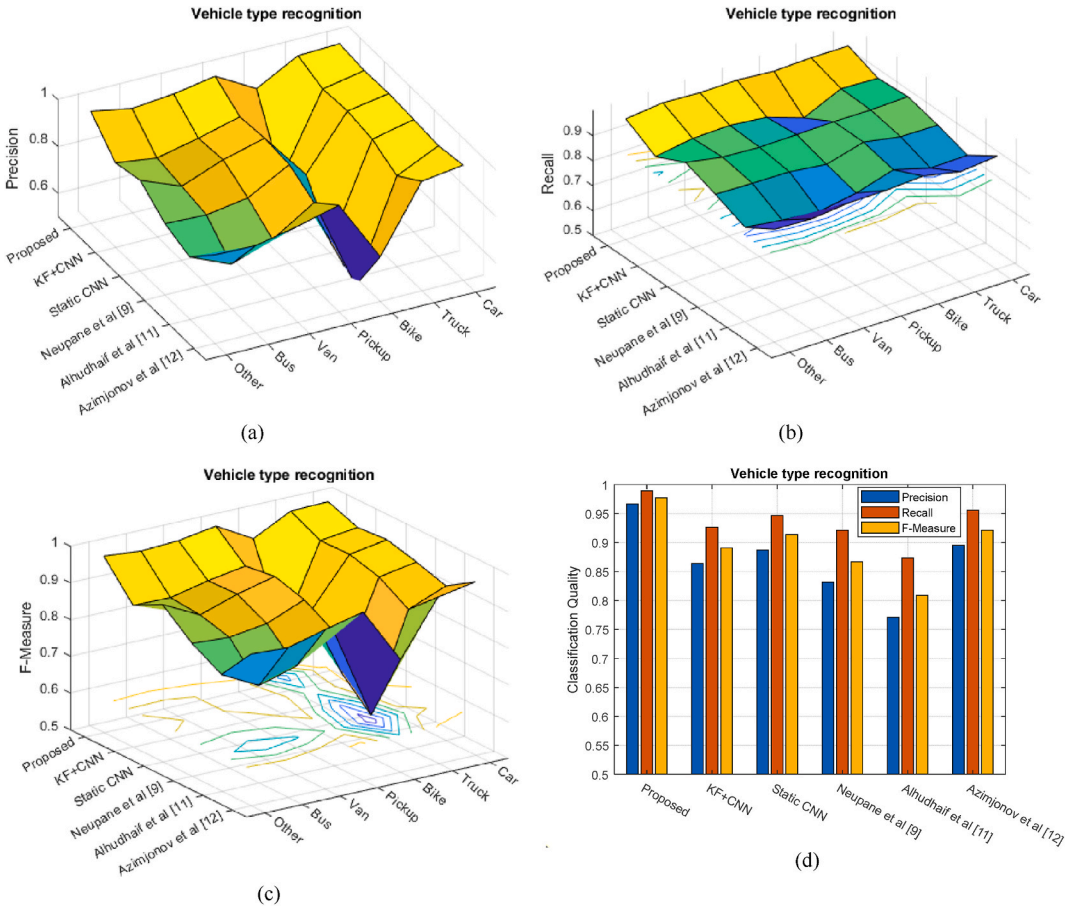

Fig. 8. Comparison of the performance of different methods in the vehicle type detection phase based on (a) precision, (b) recall, (c) F-Measure, categorized by class, and (d) the average of these metrics.

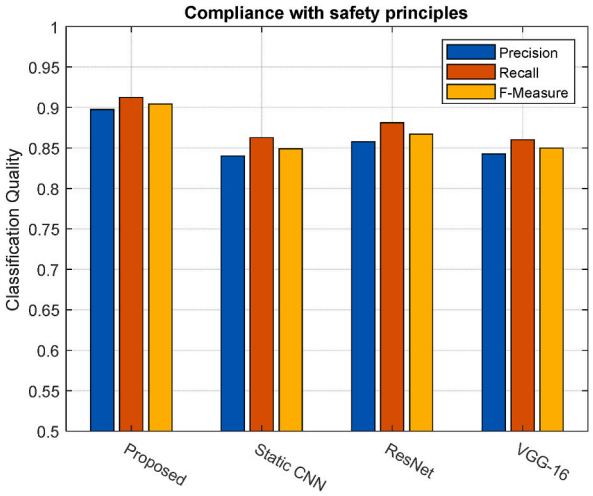

Fig. 9. Average precision, recall, and F-Measure metrics in the safety compliance detection phase.

Table 2
Comparison of the performance of the proposed method with other methods.

Phase	Method	Accuracy	F-measure	Recall	precision
Vehicle type recognition	Proposed	98.7228	0.9772	0.9888	0.9666
	KF + CNN	93.4998	0.8908	0.9260	0.8637
	Static CNN	94.5673	0.9140	0.9467	0.8872
	Neupane et al. [9]	91.5555	0.8670	0.9212	0.8318
	Alhudhaif et al. [11]	87.2284	0.8089	0.8734	0.7707
	Azimjonov et al. [12]	95.3107	0.9212	0.9556	0.8949
Compliance with safety principles	Proposed	91.5033	0.9043	0.9123	0.8976
	Static CNN	86.3834	0.8491	0.8630	0.8401
	ResNet	88.0174	0.8671	0.8813	0.8577
	VGG-16	86.6013	0.8501	0.8601	0.8428

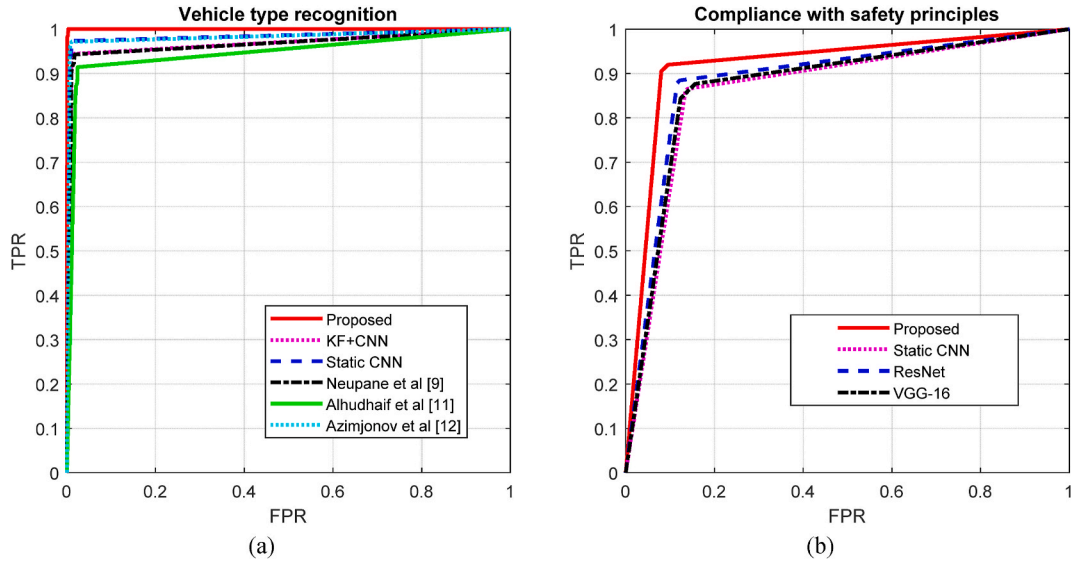

Fig. 10. ROC curves of different methods in both (a) vehicle type detection and (b) safety compliance detection phases.

proposed in this article has a high average accuracy in correctly classifying samples for both tasks.

5. Conclusion

In this article, we explored the process of analyzing highway surveillance videos for two purposes: vehicle type detection and road safety compliance assessment. To achieve this, we used a combination of EKF and deep reinforcement learning. In the proposed method, the process of target identification and tracking is performed using his EKF model. After identifying the moving target, a deep learning model and reinforcement learning-based optimization strategy are used for two-level binary classification. At the first level, a shared CNN model is used to identify the type of the moving object. Additionally, at the second level, specialized CNN models corresponding to the type of vehicle are employed to detect non-compliance with road safety regulations. It should be noted that in both of these levels, a combination of LAs and the PSO algorithm is used to determine the optimal configuration for each CNN model. Investigations have shown that the use of this optimization strategy can improve the accuracy of vehicle type detection by at least 4.98%. On the other hand, the proposed approach is capable of increasing the accuracy of safety compliance detection by 5.12%. Performance evaluation of the proposed method based on real-world surveillance videos has demonstrated that the proposed method can achieve 98.72% accuracy in vehicle type detection and 91.5% accuracy in detecting non-compliance with road safety laws, distinguishing each type of vehicle. This represents an improvement of at least 3.41% and 3.49% compared to the other methods, respectively.

One of the limitations of present study is the lack of sufficient data to assess safety compliance for all types of vehicles. Therefore, this study focused only on evaluating the safety compliance of heavy vehicles. Future work could address this limitation by developing databases tailored to this goal. We can follow two avenues for addressing these limitations in future research: The first option is planning to develop a dedicated dataset specifically designed for safety compliance assessment in various vehicle categories. The second option is investigating and implementing effective techniques such as [31,32] for handling imbalanced target classes in deep learning models. Additionally, in future research, the output of the proposed model could be used in traffic volume prediction systems. In this case, a regression model can be employed to analyze current traffic changes and predict future traffic volume.

Data availability

All data generated or analyzed during this study are included in this published article.

CRediT authorship contribution statement

Liangju Fu: Investigation. **Qiang Zhang:** Project administration, Investigation. **Shengli Tian:** Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the Gansu Provincial Department of Transportation Science and Technology Project Research on the Technical Standards for Traffic Video Cloud Networking in Gansu Province (2022–45).

References

[1] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, A review of video surveillance systems, J. Vis. Commun. Image Represent. 77 (2021) 103116.
[2] H. Cui, G. Yuan, N. Liu, M. Xu, H. Song, Convolutional neural network for recognizing highway traffic congestion, J. Intell. Transp. Syst. 24 (3) (2020) 279–289.
[3] H. Ghahremannezhad, C. Liu, H. Shi, Traffic surveillance video analytics: a concise survey, in: Proc. 18th Int. Conf. Mach. Learn. Data Mining, New York, NY, USA, 2022, pp. 263–291.
[4] S. Sri Jamiya, P. Esther Rani, A survey on vehicle detection and tracking algorithms in real time video surveillance, Int. J. Sci. Technol. Res. (2019).
[5] S. Sri Jamiya, P. Esther Rani, An efficient method for moving vehicle detection in real-time video surveillance, in: Advances in Smart System Technologies: Select Proceedings of ICFSST 2019, Springer Singapore, 2021, pp. 577–585.
[6] Q. Zhang, H. Sun, X. Wu, H. Zhong, Edge video analytics for public safety: a review, Proc. IEEE 107 (8) (2019) 1675–1696.
[7] A. Pramanik, S. Sarkar, J. Maiti, A real-time video surveillance system for traffic pre-events detection, Accid. Anal. Prev. 154 (2021) 106019.
[8] H. Song, H. Liang, H. Li, Z. Dai, X. Yun, Vision-based vehicle detection and counting system using deep learning in highway scenes, Eur. Transp. Res. Rev. 11 (1) (2019) 1–16.
[9] B. Neupane, T. Horanont, J. Aryal, Real-time vehicle classification and tracking using a transfer learning-improved deep learning network, Sensors 22 (10) (2022) 3813.
[10] R.I. Borman, Y. Fernando, Y.E.P. Yudoutomo, Identification of vehicle types using learning vector quantization algorithm with morphological features, J. RESTI (Rekayasa Sistem dan Teknologi Informasi) 6 (2) (2022) 339–345.
[11] A. Alhudhaif, A. Saeed, T. Imran, M. Kamran, A.S. Alghamdi, A.O. Aseeri, S. Alsubai, A particle swarm optimization based deep learning model for vehicle classification. Comput, Syst. Sci. Eng. 40 (1) (2022) 223–235.
[12] J. Azimjonov, A. Özmen, A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways, Adv. Eng. Inf. 50 (2021) 101393.
[13] A. Şentaş, İ. Tashiev, F. Küçükayvaz, S. Kul, S. Eken, A. Sayar, Y. Becerikli, Performance evaluation of support vector machine and convolutional neural network algorithms in real-time vehicle type and color classification, Evol. Intell. 13 (2020) 83–91.
[14] A. Appathurai, R. Sundarasekar, C. Raja, E.J. Alex, C.A. Palagan, A. Nithya, An efficient optimal neural network-based moving vehicle detection in traffic video surveillance system, Circ. Syst. Signal Process. 39 (2020) 734–756.
[15] L. Shine, J. Cv, Automated detection of helmet on motorcyclists from traffic surveillance videos: a comparative analysis using hand-crafted features and CNN, Multimed. Tool. Appl. 79 (19–20) (2020) 14179–14199.
[16] K.J. Kim, P.K. Kim, Y.S. Chung, D.H. Choi, Multi-scale detector for accurate vehicle detection in traffic surveillance data, IEEE Access 7 (2019) 78311–78319.
[17] A. Kherraki, R. El Ouazzani, Deep convolutional neural networks architecture for an efficient emergency vehicle classification in real-time traffic monitoring, IAES Int. J. Artif. Intell. 11 (1) (2022) 110.
[18] K.T. Nguyen, T.H. Hoang, M.T. Tran, T.N. Le, N.M. Bui, T.L. Do, M.N. Do, Vehicle re-identification with learned representation and spatial verification and abnormality detection with multi-adaptive vehicle detectors for traffic video analysis, in: CVPR Workshops, 2019, June, pp. 363–372.
[19] C. Kumar, R. Punitha, Yolov3 and yolov4: multiple object detection for surveillance applications, in: 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE, 2020, August, pp. 1316–1321.
[20] P. Sridhar, M. Jagadeeswari, S.H. Sri, N. Akshaya, J. Haritha, Helmet violation detection using YOLO v2 deep learning framework, in: 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), IEEE, 2022, April, pp. 1207–1212.
[21] T. Waris, M. Asif, M.B. Ahmad, T. Mahmood, S. Zafar, M. Shah, A. Ayaz, CNN-based automatic helmet violation detection of motorcyclists for an intelligent transportation system, Math. Probl Eng. 2022 (2022).
[22] A. Haghighat, A. Sharma, A computer vision-based deep learning model to detect wrong-way driving using pan–tilt–zoom traffic cameras, Comput. Aided Civ. Infrastruct. Eng. 38 (1) (2023) 119–132.
[23] M. Kathane, S. Abhang, A. Jadhavar, A.D. Joshi, S.T. Sawant, Traffic rule violation detection system: deep learning approach, in: Advanced Machine Intelligence and Signal Processing, Springer Nature Singapore, Singapore, 2022, pp. 191–201.
[24] A.J. Hoffman, P. Schutte, S.J. Rabè, Automated real-time detection of truck driver non-compliance, in: Lindholmen Conference Centre & Online October 19–20, 2022, 2022, p. 8.
[25] N. Bogdanović, H. Driessen, A.G. Yarovoy, Target selection for tracking in multifunction radar networks: nash and correlated equilibria, IEEE Trans. Aero. Electron. Syst. 54 (5) (2018) 2448–2462.
[26] B. Xu, X. Wang, J. Zhang, Y. Guo, A.A. Razzaqi, A novel adaptive filtering for cooperative localization under compass failure and non-Gaussian noise, IEEE Trans. Veh. Technol. 71 (4) (2022) 3737–3749.
[27] B. Xu, Y. Guo, A novel DVL calibration method based on robust invariant extended Kalman filter, IEEE Trans. Veh. Technol. 71 (9) (2022) 9422–9434.
[28] K.S. Narendra, M.A. Thathachar, Learning Automata: an Introduction, Courier corporation, 2012.
[29] T.M. Shami, A.A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M.A. Summakieh, S. Mirjalili, Particle swarm optimization: a comprehensive survey, IEEE Access 10 (2022) 10031–10061.

[30] D.M. Powers, Evaluation: from Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation, 2020 *arXiv preprint arXiv:2010.16061*.
[31] R.R. Boukhriss, I. Chaabane, R. Guermazi, E. Fendri, M. Hammami, Imbalanced learning for robust moving object classification in video surveillance applications, in: International Conference on Intelligent Systems Design and Applications, Springer International Publishing, Cham, 2021, December, pp. 199–209.
[32] G. Rekha, A.K. Tyagi, N. Sreenath, S. Mishra, Class imbalanced data: open issues and future research directions, in: 2021 International Conference on Computer Communication and Informatics (ICCCI), IEEE, 2021, January, pp. 1–6.