Journal of Electronic Research and Application, 2024, Volume 8, Issue 6
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Deep Learning-Based Control System Design for Emergency Vehicles through Intersections

Dingru Li[1]*, Yinghui He[1], Yuanbo Yang[2], Jiale Xu[1]

[1]College of Civil Engineering and Transportation, Northeast Forestry University, Harbin 150040, China
[2]School of Software, College of Computer Information Science, Southwest University, Chongqing 400715, China

Corresponding author: Dingru Li, lytmjt2024@163.com

Abstract: This paper addresses the challenge of integrating priority passage for emergency vehicles with optimal intersection control in modern urban traffic. It proposes an innovative strategy based on deep learning to enable emergency vehicles to pass through intersections efficiently and safely. The research aims to develop a deep learning model that utilizes intersection violation monitoring cameras to identify emergency vehicles in real time. This system adjusts traffic signals to ensure the rapid passage of emergency vehicles while simultaneously optimizing the overall efficiency of the traffic system. In this study, OpenCV is used in combination with Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to jointly complete complex image processing and analysis tasks, to realize the purpose of fast travel of emergency vehicles. At the end of this study, the principle of the You Only Look Once (YOLO) algorithm can be used to design a website and a mobile phone application (app) to enable private vehicles with emergency needs to realize emergency passage through the application, which is also of great significance to improve the overall level of urban traffic management, reduce traffic congestion and promote the development of related technologies.

Keywords: Emergency vehicle priority; Deep learning; Signal light adjustment; Traffic congestion reduction; Trajectory optimization

Online publication: December 2, 2024

1. Introduction

1.1. Background of the study

With the rapid process of urbanization, urban traffic congestion has become a big problem faced by many cities around the world daily. Traffic congestion is not only a prolongation of travel time but is also associated with increased energy consumption, increased environmental pollution, and a decline in the quality of life of urban residents. In many large cities in developing and developed countries, especially during the morning and evening rush hours, major arterial roads are often heavily congested, with vehicle speeds plummeting and in some cases coming to a complete standstill.

Traffic congestion not only increases the travel and time costs for commuters but may also delay the arrival of emergency vehicles and threaten public safety. Therefore, solving the urban traffic congestion problem is urgent and requires the joint efforts of the government, urban planners, and traffic engineers to adopt diversified measures, including optimizing the existing road network, expanding the public transport system, and promoting intelligent traffic management technologies, to alleviate the increasingly severe urban traffic pressure and enhance the overall operational efficiency of the city.

Therefore, we propose innovative deep reinforcement learning-based strategies for emergency vehicles to pass through intersections, which are essential for improving the efficiency and safety of emergency services.

1.2. Traffic signal control status

1.2.1. Existing traffic signal control systems

With the rapid development of the economy, urban transportation infrastructure has also advanced in step with economic growth. However, there is still room for improvement, particularly in the application of science and technology, specifically in optimizing existing traffic signal control systems [1]. The basic principle of a traffic signal control system is to set up at intersections and through the periodic changes in traffic signals to direct traffic flow, to achieve smooth and safe traffic. The existing traffic signal control systems mainly include the following aspects: adaptive multi-objective signal optimization for intersections with drastic fluctuations in traffic flow, a reinforcement learning method for signal timing optimization based on traffic prediction, analysis of automatic control methods for signal timing in urban rail transit, a framework for collaborative optimization of traffic signals and hybrid vehicle trajectories in multilane intersections, and so on [2–5]. However, with the continuous development of urban traffic, the traffic signal control system still needs to continuously carry out technological innovation and optimization to adapt to the more complex and diverse traffic environment in the future.

1.2.2. Strategies for prioritizing emergency vehicles

This study proposes a strategy for prioritizing the passage of emergency vehicles at intersections by optimizing the detection algorithm of intersection monitoring cameras using deep learning methods. The system records the license plates of emergency vehicles in advance to an Internet cloud database, enabling real-time identification. Through cloud computation, the system can promptly adjust traffic lights before the emergency vehicle arrives at the intersection, ensuring its safe and smooth passage.

1.2.3. Deep learning in intelligent transportation systems

Deep learning is a cutting-edge technology in the field of artificial intelligence to realize machine learning, and in recent years, many new deep learning methods have appeared and applied in many places, gradually changing people's lives. Currently, deep learning has made great progress in the field of transportation, as monitored in **Figure 1**, which provides great help for the development of intelligent transportation systems. Deep learning techniques are used in areas such as autonomous driving technology, predicting traffic speed and bicycle traffic, recognizing letters and numbers on license plates, and detecting whether drivers are distracted. Deep learning has provided many new ideas and methods for areas like traffic management and prediction, driving intelligence, and efficiency in transportation [6].

Figure 1. Intersection monitoring system using deep learning

1.3. Deep learning-based priority control system for emergency vehicles

This study uses the technology of OpenCV to optimize the detection algorithm of the intersection surveillance camera so that it can quickly detect the needs of emergency vehicles when they are passing through the intersection roads and provide clearer input data for the Convolutional Neural Network (CNN), which can be combined with the Recurrent Neural Network (RNN) to analyze the temporal dynamics information when it needs to analyze the sequential data afterward. Simultaneously, application software can be built to enable ordinary people to apply within the application when they need to travel in an emergency in special circumstances (such as patients with sudden medical conditions) so that they can also pass through the intersection as quickly as possible during emergency travel.

In this control system, our goal is to improve the efficiency of emergency vehicles (such as ambulances and fire trucks) in the road network and to reduce the time it takes for emergency vehicles to arrive at the scene of an accident or emergency. To accomplish such a goal, we need to collect real-time traffic data using traffic cameras, Global Positioning System (GPS), and in-vehicle sensors, among others, and at the same time, we need to pre-process the collected data to extract useful information, such as vehicle location, speed, and road conditions. Then we have to design the deep learning model and divide it into the following three steps.

(1) Traffic camera images are analyzed using CNN to identify vehicle types and traffic conditions [7].

(2) Analyzing Time Series Data Using RNN to Predict Traffic Flow Changes [8].

(3) Combining the results of CNN and RNN, a decision network (like a reinforcement learning-based model) is used to determine the optimal vehicle scheduling strategy.

After identification, the deep learning model is integrated into a central control system which is capable of interacting with traffic signal control systems, vehicle communication systems, and many more. After completing the system integration, we will use Vissim to simulate an intersection environment to test the performance of the system to ensure that it can work properly under different traffic conditions, adjust the model parameters, and optimize the control strategy according to the actual operation, as shown in **Figure 2**.

Finally, we would like to develop a user interface for the operators of the traffic management center to monitor and operate, as well as for the people to apply within their mobile phone application when there are special circumstances (such as a patient with a sudden medical condition) that requires emergency travel so that they can also make emergency trips. Concurrently, a feedback mechanism is provided so that the system settings can be adjusted according to the actual results.

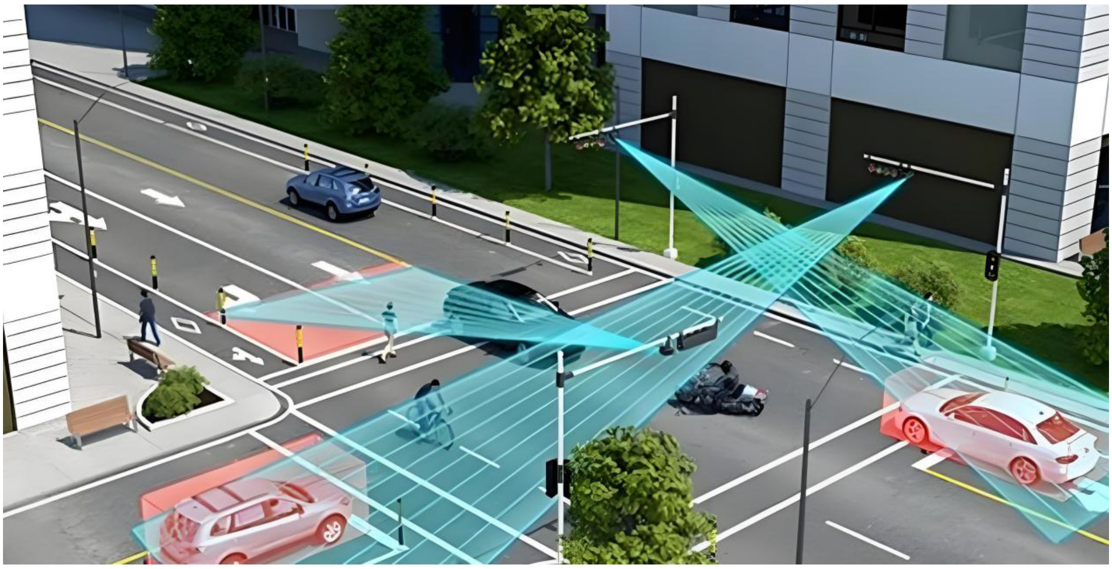


Figure 2. Emergency vehicle priority control system incorporating deep learning

2. Deep learning methods

2.1. Emergency vehicle detection and classification—image recognition using CNN

Image recognition using Convolutional Neural Networks (CNN) involves multiple steps, as outlined below [7].

Firstly, the objective of the system is clarified: to ensure that emergency vehicles, such as ambulances and fire trucks, can pass quickly at the intersection. Immediately after that, we use the pre-processing capability of OpenCV to optimize the detection algorithm of the intersection monitoring camera and collect the traffic image data of the intersection, including pictures of emergency vehicles and ordinary vehicles. The collected images are then processed, such as cropping, scaling, normalization, and so on, to fit the input requirements of the CNN model. After this, the images are labeled to differentiate between emergency vehicles and ordinary vehicles to train the supervised learning model.

After all the previous preparations, a CNN model is constructed which usually contains multiple convolutional layers, activation functions, pooling layers, and fully connected layers. The convolutional and pooling layers of the CNN are then used to extract image features that help to distinguish between different types of vehicles. Classifiers (such as fully connected layers) are also designed at the end of the CNN for recognizing and classifying the types of vehicles in the input image. Then, proceed to train the CNN model using the labeled dataset and optimize the model parameters to improve the recognition accuracy. After training, the performance of the model is evaluated on a test set to ensure that the model can accurately recognize emergency vehicles. The trained CNN model is also integrated into the signal control system at the intersection to analyze the surveillance images in real-time and respond quickly to the passage needs of emergency vehicles.

Following this, according to the recognition results of the CNN model, the signal control strategy is adjusted to provide a green light channel for emergency vehicles. During the operation of the system, the performance of the CNN model is continuously monitored and the model parameters and signal control strategy are adjusted according to the actual traffic situation.

Finally, a user interface needs to be developed to facilitate traffic managers to monitor the system status and emergency response. It also ensures the safety and reliability of the system to prevent traffic problems caused by misidentification and system failures.

Through the above steps, a CNN-based image recognition system can be constructed to realize fast recognition and priority access control for emergency vehicles at intersections. This system can significantly improve emergency response efficiency, reduce traffic congestion, and enhance public safety, as shown in **Figure 3**.

Figure 3. CNN-based image recognition for intersection monitoring system

2.2. Traffic flow forecasting—time series analysis using RNN

Time series analysis using recurrent neural networks (RNNs) is a multi-step process involving data pre-processing, model design, training, and deployment of applications [8].

The process of data collection and pre-processing involves the collection of traffic flow data at the intersection, including key information such as the number of vehicles, speed, and time of day. These data will be input into the model as time series data. The pre-processing includes data cleaning, normalization, and the like to ensure data quality.

After collecting the data and pre-processing, it is time to choose the model, in which the Long Short-Term Memory Network (LSTM) can be chosen as a variant of RNN to deal with time-series data. The LSTM can learn long-term dependencies, which makes it suitable for dealing with time-correlated data such as traffic flow. Then when constructing the RNN model, the input layer, hidden layer (including the LSTM layer) and output layer need to be considered. The input layer receives the time series data, the LSTM layer is responsible for learning and remembering the features of the time series, and the output layer predicts future traffic conditions or optimizes the signal light control strategy.

Immediately after that, training and validation will be carried out to train the RNN model using historical traffic data and adjust the model parameters by validating the dataset to improve the prediction accuracy of the model. Then testing of the model is carried out**.** In the testing phase, real-time traffic data can be input into the trained RNN model, and the model will output the prediction results, which will be used to guide the priority access strategy for emergency vehicles.

Further optimization of the model is needed to improve its efficiency and accuracy. One can consider combining the self-attention mechanism to model the hidden time series features or using more advanced network structures such as Dynamic Graph Convolutional Recurrent Network (DGCRN) to improve the model performance. Afterward, the trained model is deployed to the actual traffic signal control system to achieve real-time signal optimization and ensure that emergency vehicles can pass quickly at the intersection.

In summary, a time series analysis model based on RNN can be constructed to optimize traffic signal control at intersections and guarantee the rapid passage of emergency vehicles. In practical application, the real-time accuracy of the model and the cooperative work with other traffic management systems also need to be considered, as shown in **Figure 4**.

Figure 4. Intersection monitoring system constructed by RNN-based time series analysis model

2.3. Reinforcement learning in signal control

Combining the results of CNN and RNN, a decision network is used to decide the optimal vehicle traveling scheduling strategy.

2.3.1. Define the reward function and state space

2.3.1.1. Define the reward function

Reward functions are used to evaluate the behavior of intelligence. In an intersection signal control problem, the reward function can be designed based on the passage time of the emergency vehicle. For example, a positive reward is given if the emergency vehicle can pass through the intersection quickly, and a negative reward is given if it leads to congestion or other undesirable consequences.

2.3.1.2. Define the state space

It is necessary to define the environment of the intersection first, including information such as the status of traffic signals, position, and speed of vehicles. This information will be passed as input to the Deep Q-Network (DQN) model. In DQN, the state is an abstract representation of the environment. For the intersection signal control problem, the state can include the color of the current signal, the number of vehicles in each lane, and so on. Actions are behaviors that can be taken by an intelligent body, and in the case of intersection signal control problems, an action can be changing the color of a signal light, such as changing from red to green or yellow.

2.3.2. Q-learning or Deep Q-Network (DQN) algorithm implementation

(1) Constructing a DQN model: A DQN model usually consists of a Convolutional Neural Network (CNN) and a fully connected layer. The CNN is used to extract the features of the state and the fully connected layer is used to output the Q-value of each action.

(2) Training the DQN model: A DQN model is trained using a reinforcement learning algorithm (such as Q-learning). At each time step, the intelligent body selects an action based on the current state and then observes the next state and reward. By constantly updating the Q-value, the model will learn to select the optimal action in different states.

(3) Testing and optimization: After training is complete, the performance of the DQN model is evaluated using a test dataset. If the performance is poor, try to adjust the model structure, parameters, reward function, and the like, to optimize the model.
(4) Deployment: The trained DQN model is deployed to the actual intersection signal control system to realize real-time signal control.

3. System design and implementation

3.1. Web design

3.1.1. YOLO algorithm principle

The loss function of the YOLO family of algorithms mainly consists of coordinate point bias loss, size bias loss, in-box object confidence loss, category detection confidence loss, and node object confidence loss [9,10]. The object type detection confidence loss is only verified under the condition that the box is non-background. The loss function formula is shown below. Where indicates whether the object is present in cell i or not. This is a binary variable that indicates whether the current grid cell is responsible for predicting an object. If $x = 1$, then an object exists; if $x = 0$, then no object exists. indicates whether the j^{th} prior frame in cell i is responsible for predicting the object. There may be multiple a priori boxes contained in a grid, each of which may match the bounding box of a real object. y is used to determine which a priori box is most responsible for predicting that object.

$$
\begin{aligned}
Loss &= \lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_j^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2] \\
&+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2] \\
&+\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_j^{obj}(C_i-\hat{C}_i)^2+\lambda_{noobj}\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_j^{noobj}(C_i-\hat{C}_i)^2+ \\
&\sum_{i=0}^{S^2}\sum_{c\in classes}1_j^{noobj}\left(p_i(c)-\hat{p}_i(c)\right)^2 \quad (1)
\end{aligned}
$$

The above equation demonstrates that the total loss of the YOLO algorithm is accumulated from the losses of its components, while the details of each component are shown in **Table 1** below.

Table 1. Meaning of YOLO algorithm loss function

Hidden meaning	Term (in a mathematical formula)
Loss of target frame coordinate point deviation	$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_j^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2]$
Loss of target frame size deviation	$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2]$
Confidence loss of the object in the frame	$\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_j^{obj}[(C_i-\hat{C}_i)^2$
Nodal object confidence loss	$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_j^{noobj}(C_i-\hat{C}_i)^2$
Loss of object classification	$\sum_{i=0}^{S^2}\sum_{c\in classes}1_j^{noobj}\left(p_i(c)-\hat{p}_i(c)\right)^2$

Assuming that the generating frame *A(x1, y1, w1, h1)* where the emergency vehicle is located is calculated by the YOLO algorithm and the real frame *B(x2, y2, w2, h2)* where the emergency vehicle exists in the picture, the Intersection Over Union (IOU) calculation for the generating frame *A* is shown in **Equation (2)**.

$$IOU = \frac{A \cap B}{A \cup B} \tag{2}$$

The initial YOLOv5 uses the Generalized Intersection Over Union (GIOU) loss as shown in **Equation (3)**.

$$GIOU = \left(IOU - \frac{|C - (A \cup B)|}{|C|} \right) \tag{3}$$

GIOU cannot determine the relative position between two frames, while Complete Intersection Over Union (CIOU) is more advantageous for the target detection task in some cases, especially when dealing with overlapping or irregular shapes between targets therefore this paper adopts CIOU as shown in **Equation (4)**.

$$CIOU = IOU - \frac{\text{Distance } 1^2}{\text{Distance } 2^2} - \frac{v^2}{(1 - IOU) + v} \tag{4}$$

Where *Distance1* is the Euclidean distance between the generated frame and the center point of the real frame, *Distance2* is the diagonal distance between the smallest wrapped frames of the two frames, v is the diagonal distance between the smallest outer rectangle of the two frames, and v is:

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^g}{h^g} - \arctan \frac{w^p}{h^p} \right)^2 \tag{5}$$

Where w^{gt} and h^{gt} correspond to the width and height of the real frame, and w^p and h^p correspond to the width and height of the generated frame. Factors such as overlapping area, distance from the center point, and aspect ratio are included in the calculation of the difference between the generated frame and the real frame for target detection.

For the task of emergency vehicle detection, various data enhancement methods are adopted to improve the accuracy of YOLOv5 algorithm in recognizing emergency vehicles. Mosaic data enhancement is built into YOLOv5, which is a kind of data enhancement technology, and to help improve the performance and effect of detection. Additionally, mirror transformation, random rotation, random scaling and zooming are applied to expand the samples in the training set.

3.1.2. Design of the website

The development of the website's frontend uses the SSH framework, which is composed of Struts, Spring, and Hibernate [11]. SSH is a widely used open-source web application framework in current Java Enterprise Edition (J2EE) development [12]. The three frameworks in the Model-View-Controller (MVC) architecture each play distinct roles, with a clear separation of concerns while remaining interconnected [13]. The entire application is divided into three layers: the outermost layer which consists of the presentation layer and controller, the middle layer which handles the service layer and domain logic, and the innermost layer which focuses on data persistence and storage.

The Struts framework is mainly used in the representation layer, the Spring framework in the business logic layer, and the Hibernate framework in the data persistence layer. The representation layer mainly completes the MVC view and controller part of the content, according to the user's identity to determine the data flow. The persistence layer is responsible for the persistence of data objects and the business logic layer is mainly to complete the processing of software business logic. In the SSH framework system, the Spring framework serves as the core, integrating Struts for MVC-based control in the presentation layer and Hibernate for persistence layer access. Spring organizes objects in the presentation, business, and persistence layers in a loosely coupled manner. This design significantly simplifies system maintenance and upgrades, as the interface-oriented approach to object interaction greatly reduces the workload associated with implementing changes [14]. By analyzing the overall logic of the system, the SSH framework is derived to realize the implementation of the entire web application. The SSH framework is shown in **Figure 5**.

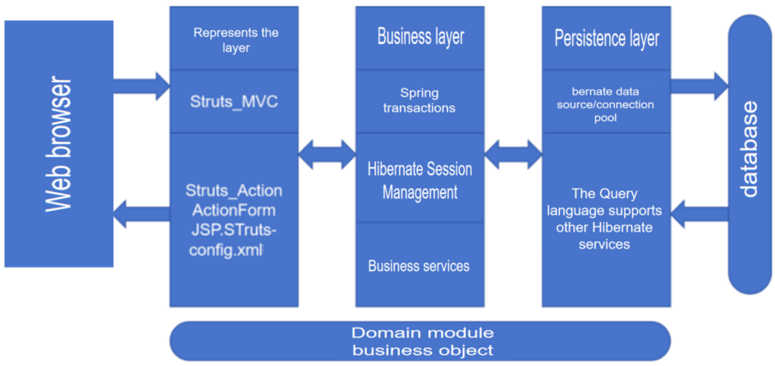


Figure 5. SSH framework

The location data for this system is intended to be uploaded using a BeiDou positioning device. To support this, a server environment with network capabilities is required. It is planned to use an Aliyun server to host the website and to purchase a domain name to enable the upload of location data from the BeiDou positioning device. The website system is developed in a Java (JDK 17) environment using MyEclipse software. The main functionalities of the site are implemented through the construction of Java classes. A brief introduction to the key Java classes used in this system is provided in **Table 2**.

Table 2. Functional description of system Java classes

Serial number	Tools	Functionality
1	com.missmore.amber.Bulletin	Realization of bulletin additions, deletions, and checks
2	com.missmore.amber.Map	Realize map data delivery to the front-end
3	com.missmore.amber.	Realization of additions, deletions, and changes in the field
4	com.missmore.amber.Position	Realization of emergency vehicle coordinate additions, deletions, and changes
5	com.missmore.amber.Users	Realization of user additions, deletions, and changes
6	com.missmore.amber.web	Implementing the detection of interfaces
7	com.missmore.kernel.sysmodule.account	Realization of user account additions, deletions, and changes
8	com.missmore.kernel.sysmodule.application	Implementing external privilege authentication
9	com.missmore.kernel.sysmodule.common	Generic feature jumps, such as the welcome page after login
10	com.missmore.kernel.sysmodule.group	Implement user rights grouping

Gaode Maps is a leading free electronic map and navigation application in China. It offers positioning and navigation functions, real-time traffic updates, congestion information, and the ability to generate routes that avoid traffic jams. These features make it one of the most popular map applications in China [15]. Gaode Maps provides developers with a comprehensive set of application interfaces based on its mapping services. These include the JavaScript Application Programming Interface (API), Web Services API, iOS Software Development Kit (SDK), Android SDK, Positioning SDK, Location-Based Services (LBS) Cloud, Telematics API, and other development tools and services. These tools enable features such as basic map display, positioning, search functionality, route planning, reverse geocoding, data retrieval, and LBS cloud storage [16–18]. LBS cloud storage and other features are suitable for developing map applications across various devices and operating systems, such as Personal Computers (PCs), mobile devices, and servers [19]. In this project, we utilize the methods provided by AutoNavi Maps to support the implementation of the system. The project primarily employs the following types of methods from AutoNavi Maps.

(1) Create a map object: First, create a map container. Then use the zoom property to set the map's display level and the center property to define the coordinates of the map's center point.

(2) Add an overlay: Create an overlay marker, then use the icon attribute to set the icon of the overlay, the position attribute to set the position of the overlay, and the map attribute to set the map container that the overlay belongs to.

(3) Add path: A path is created using the Gaode Map's polyline generation method. The path property defines the polyline's overlay path, while lineArr is an array of point coordinates. The strokeColor property sets the line's color, and strokeOpacity, strokeWeight, and strokeStyle represent the line's transparency, width, and style, respectively.

3.1.3. Website functions

The website system primarily implements an intersection monitoring function. The traffic management department begins by constructing the road layout of the intersection. The system then automatically generates the road range using a predefined algorithm. When an emergency vehicle travels on the constructed road, the system applies a road-fitting algorithm to adjust the vehicle's trajectory deviation points, ensuring the trajectory is smooth and easy to interpret. The website system includes the following pages: the homepage, login page, user page, and backend management page. The user page allows users to check for emergency vehicle trajectories on the current road section and apply for emergency travel permits for private cars. The backend management page, designed for the traffic management department, integrates the system's main functions, including location query, trajectory query, site view, message management, bulletin management, user management, and more.

3.2. Mobile app function realization

3.2.1. Mobile app overview

The mobile app of this design is developed based on the Android system, and the development tool uses Android Studio, which is an Android development environment launched by Google in 2013, with the advantages of a stable system environment, intelligent code prompts, and a powerful search function [20].

3.2.2. SOA architecture applications

Some functions of the mobile application are consistent with those of the website, such as viewing vehicle trajectories and applying for emergency travel by private car. However, due to positioning deviations, the website system's positioning and trajectory display are not achieved by directly calling methods from the Gaode API. Instead, a custom road-fitting algorithm is designed to eliminate deviations. By repeatedly refining the positioning points, the system achieves improved display accuracy. The interfaces provided by Gaode Maps differ between the Web and Android platforms. As a result, the road-fitting algorithm developed for the Web cannot be directly applied to Android. To avoid rewriting the algorithm and to ensure data consistency, the system adopts a Service-Oriented Architecture (SOA) for developing the mobile application [21]. The system encapsulates the existing methods and generates Web Service interfaces for the mobile application to call. The application sends an interface access request with parameters to the server, which processes the request, executes the corresponding functionality, and returns the result to the application [22].

3.2.3. Mobile application functions

The system's mobile application is not limited to use by the traffic management department; it is also available for general users. The main functions of the application include user login, real-time positioning, emergency vehicle tracking, and private car emergency travel applications.

(1) User login: Mobile app users are categorized into two types, ordinary users and official users from the traffic management department. Users are identified based on whether they possess official certification from the traffic management department. Official users have access to all functions of the app, while ordinary users can only access real-time self-localization and apply for emergency travel by private car after logging in. The user login interface is shown in **Figure 6(a)**.

(2) Main interface: The interface contains three buttons at the bottom: real-time positioning of emergency vehicles, track query of emergency vehicles, and acceptance of applications for emergency travel by private vehicles. These buttons are visible to official users from the traffic management department. Clicking these buttons activates their respective functions. For ordinary users, only the buttons for real-time self-positioning and application for emergency travel by private vehicles are displayed. The main interface for ordinary users is shown in **Figure 6(b)**.

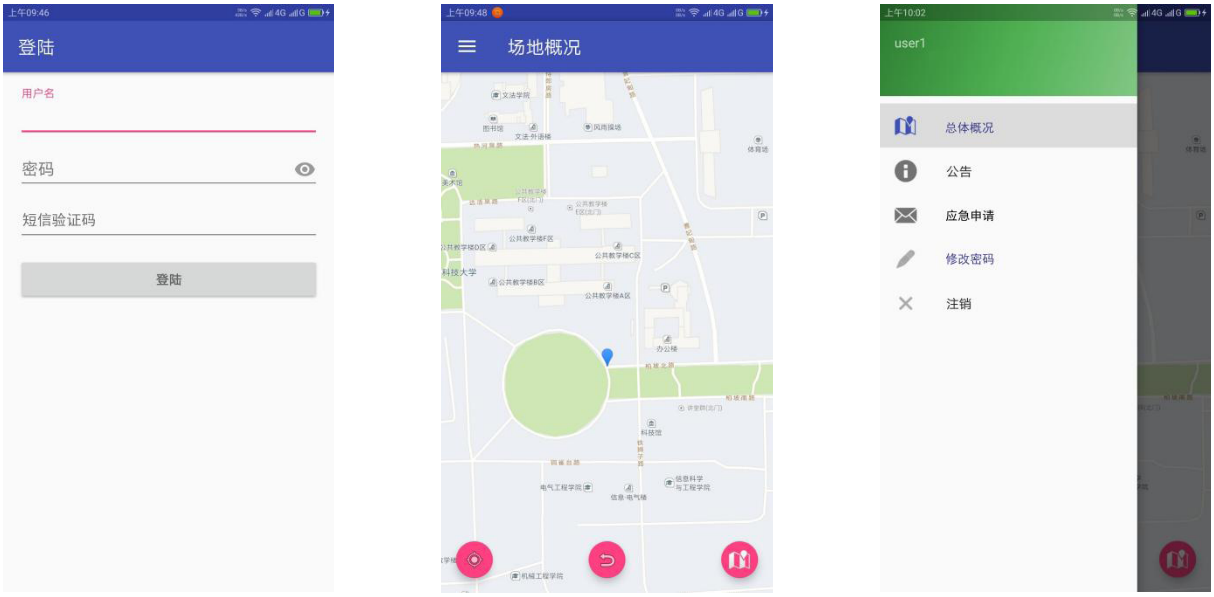
(3) Real-time positioning: Ordinary users can view the location of their vehicle for real-time positioning after logging in, while official users of the traffic management department can view the location of all local emergency vehicles after logging in.

(4) Emergency vehicle track query: By clicking on the emergency vehicle track query, the system displays the current location of the emergency vehicle and plays the track animation. If the intersection camera fails to recognize the emergency vehicle in time, the system will trigger a signal for manual operation to change the traffic light.

(5) Private car application for emergency travel: Ordinary users click on the private car emergency travel application (as shown in **Figure 6(c)**). They are required to enter their name, ID card number, and the current license plate number of their private car. They must also briefly describe the situation and the reason for the emergency travel such as transporting an emergency patient to the hospital. Finally, users must enter their phone number and the verification code before applying. The application will be processed after official acceptance.

(6) Private car application for emergency travel acceptance: When ordinary users apply, the official user from the traffic management department receives the acceptance message through the app. After

confirming the situation, the official can approve the application, adding the car's license plate number to the cloud. The system will then temporarily treat the vehicle as an emergency vehicle, allowing the intersection camera to identify it and trigger traffic signal changes.



(a) User login (b) Main interface (c) Menu screen

Figure 6. App interface display

4. Conclusion

Aiming to address the challenge of integrating priority access for emergency vehicles with optimal control of intersections in modern urban traffic, this paper proposes an innovative strategy based on deep learning for enabling emergency vehicles to pass through intersections efficiently. This approach is crucial for improving the efficiency and safety of emergency services.

In this paper, OpenCV is used to establish a deep learning model. The intersection violation monitoring camera identifies emergency vehicles in urgent need of passage and adjusts the traffic signals to ensure that these vehicles pass quickly, thereby improving the efficiency of the entire traffic system.

Next, a combination of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) is employed to handle complex image processing and analysis tasks, enabling fast travel for emergency vehicles.

Finally, the YOLO algorithm is utilized to design a website and mobile app, allowing private vehicles with emergency needs to apply for priority access. This development is significant for improving urban traffic management, reducing congestion, and advancing related technologies.

Disclosure statement

The authors declare no conflict of interest.

References

[1] Tan H, 2019, Status Quo and Development Countermeasures of Intelligent Transportation System in China. Small and Medium-Sized Enterprise Management and Science and Technology (Upper Ten Journal), 2019(01): 43–44.

[2] Mengjia Z, Siyu C, Jianwu L, et al., 2024, Adaptive Multi-Objective Signal Optimization for Intersections with Dramatic Traffic Fluctuations. Chinese and Foreign Highways, 2024: 1–13.

[3] Xu M, Li J, Zuo D, et al., 2024, Reinforcement Learning Method for Signal Timing Optimization Based on Traffic Prediction. Journal of System Simulation, 2023: 1–12. https://doi.org/10.16182/j.issn1004731x.joss.23-1416

[4] Sun J, 2024, Analysis of Automatic Control Method for Signal Timing in Urban Rail Transit. Integrated Circuit Applications, 41(02): 178–179. https://doi.org/10.19339/j.issn.1674-2583.2024.02.078

[5] Wang LF, Liu YF, Kong Z, et al., 2024, A Cooperative Optimization Framework for Traffic Signals and Hybrid Vehicle Trajectories at Multi-Lane Intersections. Control and Decision Making, 2024: 1–9. https://doi.org/10.13195/j.kzyjc.2023.1616

[6] Wu W, Gao Y, 2024, Research and Application of Artificial Intelligence Technology in Intelligent Transportation. Industry and Technology Forum, 23(03): 49–53.

[7] Dong L, Xia S, Chen K, et al., 2018, A Convolutional Neural Network-Based Method for License Plate Image Clarification, patent, cn108549892a.

[8] Cui J, Yao A, Zhao P, 2024, A Review of Research Methods for Short-Term Traffic Flow Prediction Based on Deep Learning. Journal of Transportation Engineering, 2024: 1–20.

[9] Zhao H, Feng YB, 2023, A Research on Traffic Sign Detection Based on CGS-Ghost YOLO. Computer Engineering, 2023: 1–13.

[10] Li T, Zhang YC, Zhang TC, et al., 2023, Train Driver Gesture Recognition Based on Improved YOLOv5s Algorithm. Journal of Railways, 2023(1): 75–83.

[11] Shi X, Liao S, 2009, Research on Lightweight Web Application Framework Based on Struts+Spring+Hibernate. Computer Knowledge and Technology, 5(1): 69–70.

[12] Johnson R, 2005, J2EE Development Frameworks. Computer, 38(1): 107–110.

[13] Zhang NP, Zhu FL, 2006, Research and Application of Struts Framework Based on MVC Pattern. Computer Technology and Development, 16(3): 229–231.

[14] Cai QY, 2014, Development of Message Board Based on Struts2+Hibernate+Spring. Computer Knowledge & Technology, 10(24): 5056–5058.

[15] Zhou Y, 2013, Implementation of Cell Phone Map System Based on Android Platform. Modern Computer, 2013(15): 74–76.

[16] Zou M, Zheng JH, 2012, Analysis and Application of JavaScript API in Android. Software Guide, 2012(10): 19–21.

[17] Wang W, Godfrey MW, 2013, Detecting API Usage Obstacles: A Study of IOS and Android Developer Questions. IEEE Software, 61(1): 61–64.

[18] Cheng CF, Lin DS, 2009, Research on the Application of Cloud Computing in LBS System. Fujian Computer, 2009(12): 54–55.

[19] Taly A, Erlingsson U, Mitchell JC, et al., 2011, Automated Analysis of Security-Critical JavaScript APIs. IEEE Symposium on Security and Privacy, 42(12): 363–378.

[20] Ye B, 2015, Research on Cell Phone Software Development for Android System. China New Technology and New Products, 2015(10): 12–13.

[21] Miller C, Castonguay A, Teel RW, 2015, Measuring the Impact of the Approach to Migration in the Quality of Web

Service Interfaces. Enterprise Information Systems, 9: 58–85.
[22] Cheng Y, Shen J, Mao H, et al., 2015, Research on Interface Data Integration Application Based on Web Services. Dual-Use Technology and Products, 2015(12): 109–112.

Publisher's note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.