Cole-Puli
International Conference Publishing Group

Science, Engineering and Technology Proceedings

Article

2025 International Conference on Science Technology, Architecture, Power and Intelligent Information Technology (APIIT 2025)

Autonomous Driving Design Based on Deep Learning

Zetao Yu [1,*]

[1] Minzu University of China, Beijing, China
* Correspondence: Zetao Yu, Minzu University of China, Beijing, China

Abstract: In today's world, globalization and rapid technological progress are driving deeper urbanization. Along with the explosive increase in population and the number of vehicles, urban transportation systems face unprecedented challenges, including severe traffic congestion, frequent traffic accidents, and worsening air quality. According to international statistics, traffic accidents cause million deaths annually and result in direct economic losses exceeding 0.1 trillion USD. In this context, intelligent vehicle technology, especially autonomous driving cars, is widely regarded as one of the key technologies to alleviate these issues. This thesis delves into the application of deep learning and machine vision in intelligent vehicle systems, focusing particularly on their practical effects in path planning and obstacle recognition. By systematically analyzing and evaluating existing intelligent vehicle technologies, this paper proposes an innovative algorithmic solution that combines an enhanced A* search algorithm with advanced real-time image processing techniques. Experimental results demonstrate that this new algorithm significantly outperforms traditional methods in enhancing navigation efficiency and accuracy, providing a new solution for safe navigation of intelligent vehicles in complex environments. Moreover, this research not only advances the development of autonomous driving technology but also supports the theoretical and practical implementation of future intelligent transportation systems.

Keywords: intelligent vehicles; autonomous driving; path planning; machine vision; deep learning

Received: 01 May 2025
Revised: 06 May 2025
Accepted: 15 May 2025
Published: 02 June 2025

1. Introduction

1.1. Background

With the rapid development of the global economy and the sharp increase in urban populations, urban transportation systems have reached their operational limits. Daily traffic congestion, frequent accidents, and air pollution caused by traffic have become increasingly concerning issues for urban residents. Traditional traffic management systems, due to their lack of flexibility and foresight, are no longer capable of effectively addressing the increasingly complex urban traffic demands. In this context, intelligent vehicle technology has emerged, utilizing advanced sensors, machine learning algorithms, and big data analysis to respond in real-time to environmental changes, optimize traffic flow, prevent accidents, and significantly enhance overall traffic efficiency and safety. Additionally, the widespread application of intelligent vehicles is expected to reduce urban traffic energy consumption and emission levels, supporting the achievement of urban sustainable development goals. Therefore, the development and optimization of intelligent vehicle technology, particularly in complex urban environments, have become a focus of current technological research and urban management. Intelligent vehicle technology, as a frontier of modern transportation innovation, holds the potential to optimize traffic flow, reduce collision accidents, and lower environmental impacts [1]. Through automated and intelligent technical means, it can greatly enhance the efficiency of road networks while ensuring driving safety and environmental friendliness.

1.2. Significance

This study aims to design and implement an advanced intelligent navigation system that integrates advanced path planning technology and real-time obstacle recognition. In the current context of rapid technological advancement, intelligent navigation systems are increasingly applied in both civilian and commercial fields. Improving the efficiency and accuracy of path planning, as well as enhancing the real-time and accuracy of obstacle recognition, have become key research focuses. Therefore, the primary goal of this research is to develop a path planning system based on an improved A* search algorithm. This system is designed to dynamically adjust paths in a changing urban traffic environment, responding to sudden traffic conditions and temporary road changes. In urban settings, various unforeseen factors such as traffic accidents and temporary construction frequently occur, placing higher demands on intelligent navigation systems. To adapt to these changes, the path planning of this system not only needs to respond quickly but also must be able to collect and analyze large amounts of traffic data in real-time to make optimal route choices. Enhancing the A* search algorithm for route planning, dynamically optimizing to adapt to unexpected traffic conditions and temporary road changes. At the same time, this navigation system will adopt visual processing technology based on convolutional neural networks to real-time identify and categorize various obstacles such as pedestrians, other vehicles, and road obstructions, ensuring the safe operation of intelligent vehicles in complex environments. Additionally, the study aims to prove the effectiveness of the proposed navigation system through experimental methods, including the system's response speed, accuracy, and reliability, laying a theoretical and technical foundation for the commercial application of intelligent vehicles. Through these research activities, this paper aims to contribute a novel solution strategy to the advancement of intelligent transportation systems, enhancing the intelligence of urban transportation systems, and providing solid support for achieving the goals of traffic safety, efficiency, and environmental protection.

2. Related Work

2.1. Technical Introduction

In the development of modern intelligent vehicles, one of the key challenges in autonomous driving involves effectively and safely performing path planning and obstacle detection [2]. Research employs an improved A* search algorithm and real-time image processing techniques. This provides efficient navigation for intelligent vehicles. Maintaining high efficiency and safety in various driving environments drives the development of autonomous vehicles towards higher levels of automation.

The enhanced A* search algorithm dynamically adjusts weights based on traffic and obstacle conditions in real-time to cope with the dense and changing traffic environment caused by urbanization. In practical operations, if the algorithm detects a sudden increase in traffic congestion in a certain area due to an unexpected event, it adjusts the path by increasing the cost weight of that node. This dynamic weight adjustment enables intelligent vehicles to avoid congested areas and choose alternative routes, effectively reducing economic losses and time delays due to accidents and congestion. In the enhanced A* search algorithm, the core of dynamic weight adjustment lies in adaptively modifying the cost calculation of the heuristic function based on real-time data. This adjustment ensures that the algorithm can reflect the immediate changes in urban traffic conditions, thereby guiding intelligent vehicles to avoid high-risk areas.

Real-time image processing technology involves a series of processes from capturing real-time images to analyzing and classifying obstacles through deep learning models, and then dynamically adjusting the vehicle's route to avoid these obstacles. In intelligent transportation systems, real-time image processing technology plays a crucial role, especially in the application of autonomous vehicles. This technology captures visual information of the external environment in real-time through onboard cameras and then analyzes and processes this data using advanced computing platforms to achieve precise navigation and obstacle avoidance in complex traffic environments. The core of real-time image processing technology includes three main stages: real-time image capture, analysis and processing through deep learning techniques, and dynamic path adjustment based on these analysis results. Real-time image capture is accomplished through high-performance onboard cameras. These cameras not only capture high-resolution images but also have good adaptability to different lighting conditions to ensure clear capture of road, pedestrians, vehicles, and other critical information both day and night. Modern vehicles typically have multiple cameras distributed around the vehicle to achieve 360-degree coverage. The analysis and processing of image data mainly rely on deep learning models.

Deep learning excels in image recognition and classification, effectively identifying various objects in images, such as pedestrians, different types of vehicles, traffic signs, and even small dynamic and static obstacles like animals. In autonomous driving technology, deep learning models are continuously trained using large amounts of real-time and historical image data to improve their recognition accuracy. These models can quickly identify key information from complex road environment images and classify different types of obstacles. The dynamic path adjustment based on image processing results is a critical step in achieving safe driving. The system evaluates the current road conditions and potential safety risks based on the recognition and classification results, and dynamically plans the best route to avoid obstacles while maintaining driving efficiency. This process involves complex algorithms that require real-time calculations and adjustments to respond to various situations on the road. For example, when the system detects a sudden obstacle on the road ahead, such as a stopped vehicle, construction, or accident, the autonomous driving system immediately calculates a new route to guide the vehicle to safely bypass the obstacle and avoid potential collisions or delays.

The development of real-time image processing technology has not only enhanced the safety of autonomous driving vehicles but also significantly improved their environmental adaptability and driving efficiency. With the further advancement of artificial intelligence and machine learning technologies, future real-time image processing technology will become even more accurate and efficient, thus providing support for autonomous vehicles in more complex environments. As technology has progressed, both the speed and accuracy of image processing have been significantly improved, enabling vehicles to react instantaneously upon receiving information, whether they are driving at high speeds on highways or navigating through congested urban streets.

The development of image capture devices is equally important. Modern vehicle-mounted cameras can now provide high-quality video streams under various lighting conditions. These devices are typically equipped with high dynamic range imaging (HDR) technology, as well as advanced image stabilization and noise reduction technologies, ensuring that clear images can be captured even in extreme weather conditions or low-light environments. In addition, with the development of hardware technology, the size of cameras has been further reduced while their functions have become more powerful. This allows them to be installed more flexibly on vehicles, monitoring the environment around the vehicle from various angles.

The application of deep learning models has opened up new possibilities for the development of image processing technology. These models are capable of learning from a large amount of data and extracting useful information for real-time decision-making. Through continuous training and optimization, these models are becoming increasingly proficient at quickly identifying key information from complex scenes, such as recognizing and distinguishing various traffic signs, pedestrians, and different types of vehicles. Deep learning not only improves the accuracy of image classification but also shortens the processing time, enabling vehicles to react rapidly after receiving image information.

Dynamic route adjustment is a complex process in real-time image processing technology. It involves the immediate analysis of road condition information and the prediction of possible changes, and then making route choices based on this information. This process not only depends on the results of image processing but also requires the integration of the vehicle's real-time speed, destination information, and other environmental factors, such as traffic flow and weather conditions. The dynamic route adjustment system must be able to complete this judgment and decision-making process within milliseconds to ensure that the vehicle can maintain the optimal driving route in a constantly changing road environment.

With the development of vehicle-to-everything (V2X) technology, the application of real-time image processing technology in the field of autonomous driving will become even more extensive. Vehicles can not only analyze the images captured by vehicle-mounted cameras but also optimize their driving routes and driving strategies through real-time data exchange with other vehicles and road infrastructure. For example, through V2X communication, a vehicle can be informed in advance of the traffic conditions on the road ahead, such as information about traffic accidents or road construction, and then quickly adjust its driving route to bypass congested areas, thus greatly improving travel efficiency and safety.

In today's era of continuous progress in autonomous driving technology, real-time image processing has become an indispensable part. From image capture, deep learning analysis to dynamic route adjustment, every link requires highly precise and rapid response technical support.

2.2. Research Status

In the field of unmanned driving, the development of autonomous navigation plays a core role. The research significantly enhanced the independent navigation performance of drones in changeable environments by adopting an optimized deep reinforcement learning algorithm [3]. In addition, through the use of computer vision technology, the identification and classification processes of the urban waste management system became more efficient, thereby enhancing the overall efficiency of urban cleaning [4].

Sensor technology plays a crucial role in autonomous cleaning systems [5]. The ultrasonic radar used in the automatic parking system was modeled to provide a reference for obstacle detection and spatial positioning technology in automatic cleaning robots. The research results show that thanks to the continuous evolution of sensor technology, the navigation and cleaning functions of the automatic cleaning system have been significantly improved in complex environments [6].

In terms of path planning, current research mainly focuses on improving traditional algorithms to meet the needs of dynamic environments. For example, the value function was improved through interaction with pedestrians, enhancing the robot's awareness of obstacles [7]. Although indoor navigation robots generally perform better than other types of robots in terms of decision-making randomness and uncertainty, the excessive redundancy of indoor environment information may still cause these robots to fall into dead zones where navigation is impossible, or encounter problems such as obstacle avoidance failure and system deadlock. To improve the work efficiency of indoor navigation robots and optimize their obstacle-avoidance paths, the key lies in improving the navigation system so that it can more accurately process and apply information related to the robot's environment [8].

To ensure the effectiveness of obstacle recognition and path planning, the key is to first obtain important environmental information with active obstacle-avoidance functions from the advanced decision-making system of the indoor navigation robot. This data is provided by a navigation system optimized by a deep learning algorithm and contains details of the robot's current environment. Using this detailed information, an accurate simulation map reflecting the real operating environment can be constructed, which in turn guides the robot to perform precise obstacle avoidance and path planning [9].

An optimized smooth ant colony algorithm is adopted to address the challenges in robot autonomous navigation, aiming to ensure safety inspections and improve the reliability of obstacle avoidance for path planning [10]. It is increasingly used to solve the task allocation and path planning problems of USVs [11].

Using a two-layer scheduling strategy, unmanned vehicles are dedicated to transporting containers between different parking spaces. The method of hybrid chaotic electronic search and multi-population genetic algorithm (ES-mGA) is adopted to reduce path tracking errors and effectively solve the problems of the two-layer unmanned vehicle scheduling model [12]. An improved Q-learning algorithm called neural network smoothing and fast Q-learning (NSFQ) outperforms the traditional A* and RRT algorithms in evaluation indicators such as heading angle, angular velocity, path length, navigation time, and path smoothness [13]. However, the performance of this algorithm in dense obstacle environments still needs to be optimized. In addition, a new path planning framework that integrates multi-objective optimization and perceptual vector re-planning strategies can organically execute global planning and local response [14]. At the same time, the PI-DP-RRT method combines the information of the existing Automatic Identification System (AIS) and the Douglas-Peucker (DP) compression technology for ship path planning. Compared with the traditional RRT algorithm, it shows a better performance balance in terms of efficiency and accuracy and successfully reduces the number of turns in the navigation path [15]. However, the adaptability of this method in dealing with different traffic conditions still needs to be further improved.

Currently, many scholars at home and abroad have conducted research on the methods of assisting object contour recognition, contact, and obstacle removal. Researchers such as Domivos have developed a real-time dynamic planning technology for the robotic end-effector based on online point cloud data [16]. This technology adjusts the preset obstacle removal path on a two-dimensional plane through a specific navigation algorithm, enabling the robotic arm to adjust its distance from the object surface while maintaining perpendicularity to it. However, this technology has not yet been able to achieve precise obstacle removal in specific areas.

An obstacle removal assistance robot was designed to facilitate obstacle removal in a vertical posture [17]. Its obstacle removal device drives multiple obstacle removal heads to move up and down reciprocally through a servo motor, but it fails to achieve precise fitting with the surface, and the effectiveness of its obstacle removal still needs to be verified.

A robotic system was designed for contact-type obstacle removal. The system is driven by a motor on a slider and uses an elastic rope to make the obstacle removal tool closely adhere to the object surface [18]. This system can effectively identify the attachments on the object surface, and its obstacle removal effect is better than that of the non-contact method. However, the accuracy and comfort of this method cannot be fully guaranteed.

A motion planning method that integrates visual perception and CC-DMP can achieve path planning for the robotic end-effector and is also equipped with professional obstacle removal actions to improve user comfort [19]. However, it is still difficult to achieve precise obstacle removal when facing different object surface structures.

A robotic obstacle removal system was proposed that uses a depth camera and a soft tactile sensor to track the object contour [20]. During the obstacle removal process, the robotic arm always maintains a certain pressure in contact with the object surface, but the end is not equipped with soft obstacle removal materials, and it is difficult to completely remove obstacles in areas with complex contours. It can be seen that although these studies have all achieved contact-type obstacle removal actions, they have not been able to fully adapt to the complex contour structures of objects. Therefore, it is still necessary to develop an auxiliary robotic control method for obstacle removal that can ensure the accuracy of obstacle removal.

3. Algorithm Design

Path planning and obstacle recognition are crucial. The purpose of the algorithm is to solve the problems of real-time path planning and obstacle recognition in a dynamic environment. The core of the algorithm is the improved A* algorithm, which enables responses to and navigation in complex environments.

The design concept of the algorithm is to ensure real-time performance and accuracy, and to enhance the environmental adaptability of intelligent vehicles. To achieve this, we have improved the traditional A* search algorithm, added a response mechanism for dynamic environmental changes, and integrated real-time image processing technology. Systematic tests are carried out in both controlled environments and on public roads to verify the practicality and effectiveness of the technology.

3.1. Enhanced A Search Algorithm*

In an urban traffic environment, the real-time data of a certain node n shows that there is traffic congestion in this area due to an accident. In such a situation, we need to increase the cost of passing through this node so that the path planning will tend to bypass this congested area. Our heuristic function consists of two parts. The actual cost g: The actual movement cost from the starting point to the current node n. The estimated cost h: The estimated minimum cost from node n to the target (common estimation methods include the Euclidean distance or the Manhattan distance).

We introduce a weight factor w, which will be dynamically adjusted based on real-time traffic data. For example, if the traffic congestion level in the area where node n is located is very high, we can set $\mathrm{w(n)}$ to 1.5 (this value can be adjusted according to the severity of the real-time traffic congestion; more severe congestion can correspond to a higher weight). Therefore, the adjusted heuristic function $\mathrm{f(n)}$ can be expressed as:

$$\mathrm{f(n) = w(n) \times g(n) + h(n)}$$

Where: $\mathrm{g(n)}$ increases due to the influence of real-time traffic conditions, and this change is reflected through the adjustment of $\mathrm{w(n)}$. $\mathrm{h(n)}$ remains the estimated cost based on static map data.

In this way, the enhanced A* algorithm can flexibly adapt to complex real-time driving environments, effectively plan the optimal path from the current location to the destination, and at the same time avoid high-risk traffic areas formed due to unexpected events.

The algorithm process is as follows: 1. Initialization (add the starting point to the open list), 2. Loop (as long as the open list is not empty, take out the node with the lowest heuristic function value from it), 3. Target test (if the node is the target node, reconstruct the path), 4. Neighbor expansion (add neighbor nodes to the open list, and update the actual cost from the starting point to the current node n, as well as the estimated minimum cost from the node to the target for each neighbor), 5. Path reconstruction (start from the target node and trace back to the starting point in reverse to reconstruct the path)

The entire process demonstrates the working procedure of a complex enhanced A* search algorithm. This process adapts to the continuous changes in the urban traffic environment through real-time data and dynamic weight adjustment.

This system is composed of multiple main control nodes and several auxiliary data channels, specifically including:

1) Central Processing Unit (DEATGE): This is the core node of the entire system, responsible for coordinating all path calculation and data processing tasks. It receives inputs from various sources, analyzes the data, and outputs decision-making instructions.
2) Input and Output Nodes: NOFE and DAIE are data input nodes, which represent the input of external traffic data and real-time monitoring data. DÆBFEL

and ACATIIC are data output nodes, responsible for sending processing results to the next step or external systems, such as a traffic control center or vehicle navigation system.

3) Data Processing Node (DNO OE): This node processes data received from the input nodes, such as real-time traffic conditions and road information, performing necessary preprocessing for use by the central processing unit.
4) Path Calculation Node (RΔILT): Responsible for executing the path calculation of the enhanced A* algorithm. According to instructions from the central processing unit and combined with real-time data and dynamic weights, it calculates the optimal path from the current location to the destination.

The relationships between these nodes are clearly defined. Data flows from the input nodes NOFE and DAIE to the central processing unit DEATGE, meaning real-time traffic data and other relevant information are sent to DEATGE for analysis and processing. Data processed by the DNO OE node is further transmitted to DEATGE to assist in refined path calculation and decision-making. After processing, DEATGE sends output path decisions or related decision information to external systems or internal operations through the output nodes DÆBFEL and ACATIIC.

Interaction between the central processing unit and the path calculation node involves DEATGE sending calculation requirements and real-time data to RΔILT, which performs the specific path calculation task and feeds results back to DEATGE.

This system implements a dynamic weight adjustment mechanism. Real-time data received by DEATGE, such as traffic congestion information, is used to adjust weight parameters passed to the RΔILT path calculation node. These weights affect the heuristic function calculations, dynamically adjusting the $g(n)$ and $h(n)$ values of the nodes.

When DEATGE receives congestion information about specific nodes or areas from NOFE and DAIE, it dynamically adjusts the weight factor $w(n)$ passed to RΔILT accordingly, possibly increasing it to 1.5 or higher to reflect severe congestion.

At the RΔILT node, $w(n)$ combines with the actual $g(n)$ and estimated $h(n)$ to calculate a new $f(n)$, ensuring path calculations adapt to real-time changes and prioritize smoother routes. Results are fed back to DEATGE, which evaluates the effectiveness of the current route decision and readjusts weights and paths based on updated data.

Input data from NOFE and DAIE are preprocessed by DNO OE and transmitted to DEATGE to ensure data accuracy and timeliness. DEATGE receives data from each node, formulates adjustment decisions, and achieves optimal calculations through RΔILT.

Final decisions are output through DÆBFEL and ACATIIC, which may involve sending route instructions to vehicle navigation systems or traffic management centers to implement specific control measures.

The implementation of this enhanced A* search algorithm extends beyond theoretical optimization, finding wide application in actual urban traffic systems. Through dynamic weight adjustment, intelligent vehicles can flexibly respond to sudden road conditions such as accidents or temporary changes, choosing optimal driving routes. Moreover, this system offers powerful decision support for urban traffic management departments, helping manage traffic flow more effectively, reduce congestion, and improve road usage efficiency. By monitoring traffic in real time and adaptively adjusting path weights, the system promptly reflects traffic condition changes and provides optimal solutions.

The system can identify high-risk areas and suggest detours timely, reducing traffic accident risks and ensuring driver and passenger safety. By avoiding congested areas, it reduces fuel consumption and significantly improves travel efficiency, saving users valuable time.

In this enhanced A* search algorithm, each node and their interrelations are carefully designed to effectively process and respond to complex and dynamic urban traffic environments. Its successful implementation demonstrates the application potential and practical effects of modern technology in intelligent transportation systems, providing valuable experience and inspiration for future developments. Through this advanced navigation system, traffic efficiency can be improved while contributing to the sustainable development of cities.

Thus, the enhanced A* algorithm flexibly adapts to complex real-time driving environments and effectively plans the optimal path from the current location to the destination.

3.2. Real-time Image Processing Technology

Optimization of Feature Extraction: According to the complexity of urban scenarios, the appropriate depth of the network model and the size of the filters are automatically selected to accurately distinguish between static and dynamic obstacles. For example, in densely populated areas, the depth of the network is increased to handle the complex interactions between pedestrians and non-motor vehicles.

Adjustment of Intelligent Obstacle Recognition: A flexible classification layer is designed to adjust the recognition strategy according to the current road environment and real-time traffic conditions. For example, by increasing the sensitivity of classification, the recognition of small or partially occluded obstacles is optimized.

Real-time Data Fusion: Real-time traffic flow information and historical data are integrated to intelligently adjust the heuristic score of the A* algorithm, reflecting the impact of unexpected events on the recommended path. For instance, the algorithm parameters are adjusted in real-time to avoid newly emerged accident scenes or areas under temporary traffic control.

Dynamic Adjustment of the Executed Path: The driving strategy is adjusted according to the immediate feedback of the vehicle, such as real-time speed adjustment and fine-tuning of the path, to ensure smooth driving in complex urban environments. For example, the vehicle slows down in real-time to pass safely through areas with a large number of pedestrians crossing the road.

In this way, the real-time image processing technology is not just a functional module in the intelligent vehicle system, but has become a key technology for the urban traffic system to deal with complex challenges. The above parameter and technology adjustment schemes aim to ensure that intelligent vehicles can operate safely and efficiently in the increasingly busy and changeable urban traffic environment, while significantly improving the originality and practicality of the system.

The system demonstrates the complete process of image processing and path planning for intelligent vehicles, integrating real-time image capture, deep learning analysis, and dynamic route adjustment.

The nodes and their relationships are described as follows:

1) Real-time Image Capture Node (Camera): Responsible for capturing visual information around the vehicle from various angles, including traffic conditions, road signs, and obstacles.
2) Image Preprocessing Node: Receives the original image data from the cameras and performs operations such as light adaptation, intelligent denoising, and preliminary processing to prepare the data for feature extraction.
3) Feature Extraction and Obstacle Recognition Node: Uses a deep learning model to process the preprocessed images and is responsible for identifying and classifying both static and dynamic obstacles.
4) Real-time Data Fusion Nodes: Process real-time data from vehicle sensors and external traffic management systems, fusing it with image data. This fused data is used to dynamically adjust the heuristic score in the A* algorithm, reflecting the impact of current traffic conditions on the recommended path.
5) Dynamic Path Adjustment Node: Based on the data from feature extraction, obstacle recognition, and real-time data fusion, it dynamically adjusts the vehicle's

driving strategy and route, including real-time speed adjustment and path fine-tuning to ensure safe and efficient navigation in complex urban environments.

The data flow proceeds as follows: the images captured by the cameras first undergo preprocessing to adjust brightness, reduce noise, and prepare for further analysis. The preprocessed data is then sent to the feature extraction and obstacle recognition node, which performs the core analysis using deep learning to detect key elements and obstacles.

Next, the real-time data fusion nodes combine the external and historical data with the image data to dynamically adjust the A* algorithm parameters for optimal path calculation. This ensures avoidance of accident-prone or temporarily restricted areas based on real-time traffic.

The output from the obstacle recognition is forwarded to the dynamic path adjustment node, which modifies the vehicle's real-time driving strategy, such as speed and route, according to current road and environmental conditions.

This intelligent vehicle system design prioritizes real-time responsiveness and adaptability, enabling quick reactions to environmental changes and optimized driving decisions. This not only enhances driving safety but also improves route efficiency. The system continuously monitors the surrounding environment through cameras and sensors, providing real-time feedback to preprocessing and feature extraction nodes to ensure timely and accurate information.

Based on the real-time analysis and changing external conditions, the system dynamically adjusts the route and driving strategy to effectively handle immediate road situations and environmental challenges.

The coordinated design of all nodes effectively manages the traffic environment, improving environmental perception and decision-making capabilities. This greatly enhances the potential for future complex traffic scenarios and reduces the likelihood of accidents. The incorporation of real-time image processing technology significantly boosts the environmental perception capability of autonomous vehicles.

3.3. Innovative Points of the Algorithm

The core of the innovation lies in modifying the heuristic function of the algorithm and introducing a dynamic weight adjustment mechanism. The heuristic function no longer solely relies on the physical distance (such as the straight-line distance), but combines dynamic information about traffic conditions.

In terms of cost calculation, the original heuristic function was usually h(n) = distance(n, goal), that is, the straight-line distance from node n to the target. In my design, this function is modified to h(n) = distance(n, goal) + traffic_cost(n), where traffic_cost(n) is the additional cost calculated based on the traffic conditions of the current node n.

Regarding traffic condition adjustment, the value of traffic_cost(n) is dynamically adjusted according to real-time traffic data. For example, if there is a traffic accident or severe congestion on a certain road, this function will add an additional cost to this road section, prompting the algorithm to avoid this section.

The dynamic weight adjustment mechanism means that the cost of each node not only reflects the distance but also includes the traffic conditions of that road section. If a certain road section is congested, the algorithm will increase the weight of this road section in real-time, making the path planning tend to bypass the congested area and choose an alternative route.

The system integrates data from various sensors and real-time traffic monitoring systems, such as traffic cameras, in-vehicle sensors, etc. Data integration is the process of integrating real-time traffic data from different sources into the system so that this data can be used to dynamically adjust the weights of the A* algorithm.

When collecting data, the system obtains data through APIs or directly from the traffic monitoring center, including traffic camera data, in-vehicle sensor data, GPS information, etc. Then, middleware or data integration tools, such as Apache Kafka or RabbitMQ, are used to uniformly format the information collected from various sensors and data sources and transmit it to the processing platform in real-time.

Once the data is integrated into the system, the next step is to analyze this data and convert it into weight adjustment indicators used in the algorithm.

First, data analysis is carried out. The data analysis module is used to process real-time traffic data and identify key information, such as the location of accidents, the density of traffic flow, etc. This step often uses machine learning models or statistical methods to predict the traffic conditions of specific nodes or road sections.

Then, weight conversion is performed. According to the analysis results, the system calculates the traffic_cost value of each node. For example, if a node is in an accident area, the cost of this node will be increased accordingly.

Finally, real-time input is carried out. The updated weight values are input into the A* search algorithm in real-time. This step is usually achieved through data structures in memory, such as hash tables or queues, to ensure that the path search algorithm can access the latest weight information immediately.

The introduction of dynamic weights enables the A* algorithm to flexibly respond to changing traffic conditions, improving the practicality and reliability of path planning. It can quickly react to changes in traffic conditions, provide the optimal driving route for intelligent vehicles, reduce stagnation time, and improve travel efficiency. By avoiding possible congested and accident areas, driving safety is improved. It also reduces fuel consumption and time waste caused by traffic congestion, bringing significant economic benefits.

3.4. Algorithm Performance Evaluation

To evaluate different path planning and obstacle avoidance algorithms, several performance evaluation indicators are set:

Success rate r: It represents the reliability of the algorithm in completing path planning. A 100% success rate means that this algorithm has successfully found more paths from the starting point to the end point in the test cases compared to other algorithms. This is a reference value, indicating that it is superior to other algorithms, rather than being truly flawless.

Path length s: It is a comparison between the path planned by the algorithm and the ideal path.

Calculation time t: The time required for the algorithm to calculate the complete path from receiving the initial data.

Number of obstacle avoidance operations m: During the process of path planning by the algorithm, how many times adjustments are needed to avoid obstacles. This indicator reflects the adaptability and flexibility of the algorithm.

Combining these parameters, we define a comprehensive evaluation rate k. Considering the importance of the success rate, we assign it a higher weight. The calculation method of the evaluation rate is as follows:

$$k = r * 0.6 + (100 - s) * 0.2 + (100 - t) * 0.1 + (100 - m) * 0.1$$

Where the path length ratio, calculation time ratio, and obstacle avoidance operation ratio are the relative proportions to the best performance values.

4. Algorithm Implementation

After designing the path planning and obstacle recognition algorithms for intelligent vehicles, implementing these algorithms is the key to the next step. This chapter will introduce the specific implementation methods of the algorithms, including the selection of the development environment, the application of programming languages, the process of modeling and training, and the implementation strategy of the entire thesis.

4.1. Development Environment and Programming Languages

Ubuntu 20.04 LTS is selected as the main development environment. It provides a stable ROS (Robot Operating System) framework, which facilitates rapid development and testing.

Python and C++ are chosen as the programming languages. Python, with its efficient coding ability and rich scientific computing libraries, is used for rapid prototyping and preliminary testing of algorithms. On the other hand, C++ is used to handle more complex data structures and algorithms, especially in the parts of path planning and image processing with high real-time requirements.

4.2. Modeling and Training

Some optimizations are carried out based on mature technical solutions. The specific strategies are as follows: In terms of mapping, considering that the track environment is relatively simple and the site area is not large, we choose to use the Gmapping particle filter algorithm for map construction; in terms of navigation, we adopt the ros_navigation navigation stack provided by Google for path planning and navigation. The ros_navigation navigation stack is a set of 2D navigation function packages, which relies on odometry data, the tf coordinate transformation tree, and sensor data to provide the output of the destination location and safe driving speed for mobile robots. This set of navigation functions relies on extracting information from odometry and sensor data and sending speed commands to the mobile platform (such as a robot). The prerequisite for using this set of navigation function packages is that ROS must be running on the robot system, the tf transformation tree must be configured, and sensor data must be published through appropriate ROS message types. At the same time, the navigation package also needs to be configured at a high level for robots with specific shapes and dynamic characteristics. In this thesis, we use the Gmapping algorithm based on lidar and odometry data. This is a widely used and reliable mapping technology in the ros system and belongs to the slam_gmapping software package of the ros-perception organization.

These parameters are configured in the ROS launch file, and Gmapping is launched. Its inputs include lidar and odometry data, and the outputs are the robot's position and the constructed map. Understand the operation structure of the Gmapping algorithm by analyzing the information flow. As shown in Figure 1.

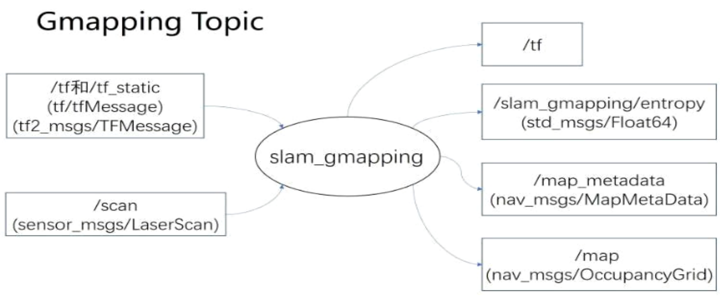

Figure 1. The structure of the gmapping algorithm in actual operation.

The core component is the slam_gmapping node, which undertakes the entire task of gmapping SLAM. This node subscribes to two key data streams: /tf and /tf_static, which are coordinate transformation messages of the tf2_msgs/TFMessage type. The core transformations include:

base_frame to laser_frame: This is the coordinate transformation from the robot chassis to the lidar, which is crucial for determining the position of the lidar relative to the robot.

base_frame to odom_frame: This represents the coordinate transformation from the chassis to the origin of the odometer (usually regarded as the starting point of the robot's movement), and the odom_frame is regarded as the coordinate system of the odometer.

/scan: The data emitted by the lidar, with the data type of sensor_msgs/LaserScan.

Through the coordinate transformation of the /tf node, slam_gmapping can map the surrounding obstacles into the robot's coordinate system. More importantly, the transformation between the base_frame and the odom_frame reflects the odometer data monitored through various means (such as the photoelectric encoder of the motor, visual odometry, IMU, etc.), that is, the actual distance traveled by the robot. This transformation information is published between the odometer coordinate system (odom_frame) and the lidar coordinate system (laser_frame), enabling slam_gmapping to obtain detailed data about the robot's movement from /tf. As shown in Figure 2.

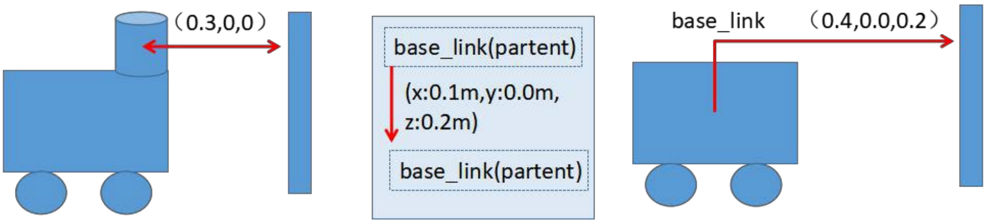

Figure 2. TF Coordinate Transformation.

In actual use, gmapping mapping has relatively strict requirements for laser data. Therefore, materials that absorb a large amount of laser light can significantly affect the mapping effect. At the same time, odometry deviations also have a significant impact on gmapping. So, gmapping is more suitable for small-scale mapping, such as in a small indoor space. Through multiple tests, we found that in the case of a competition venue, the mapping accuracy of gmapping is relatively high, and it can well identify the map boundaries and obstacles.

The ros navigation stack of the navigation stack is a classic set of navigation packages in ROS. After obtaining a map through methods such as gmapping, it can implement a series of core navigation functions such as localization, global path planning, and dynamic local planning. It contains 16 packages, each of which performs a specific function and jointly serves navigation. This structure is shown in Figure 3.

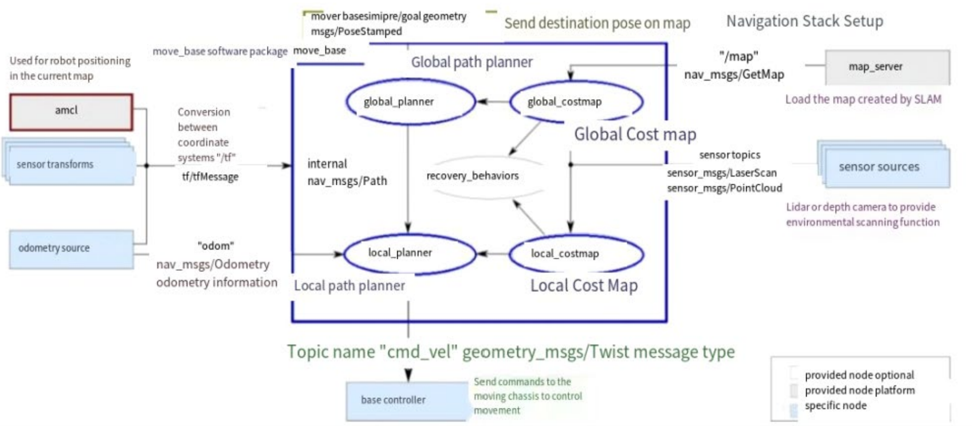

Figure 3. The navigation stack.

The white boxes are provided to us by the navigation package, and the gray ones are optional nodes, such as the AMCL and map_server nodes. In fact, we may not need the localization of AMCL. We can achieve our goals in a relatively small space relying only on the odometer and the radar. Additionally, a map is not necessary either, because within the navigation package, we can complete the navigation to the target point relying solely on local path planning. In this framework, the global_planner and the local_planner are the two most important aspects.

base_local_planner: It undertakes the task of path planning within a local window, and the specific generation of the robot's movement speed is completed in this package. Currently, there are two implementations of local path-planning algorithms. One is TrajectoryROS, and the other is the Dynamic Window Approach (DWA). The default implementation within this package is TrajectoryROS, but an interface for defining DWA is left. The implementation of DWA is in dwa_local_planner. global_planner: It is the global path-planning node. The global_planner has the same function as navfn that follows. They both implement global path planning between the target point and the current point, and both have implementations of the Dijkstra algorithm and the A* navigation algorithm. The ROS system by default uses navfn.

After the model training is completed, it is integrated into the decision-making system of the intelligent vehicle. It can receive the image data captured by the camera in real-time and process it, providing a feasible obstacle-avoidance strategy for vehicle navigation.

The implementation of this thesis relies on a series of software packages and modules. The following are the main files and their functions:

path_planning.py: Responsible for the main logic of path planning. The.py suffix indicates that this file is a Python script.

obstacle_detection.cpp: Implements the core algorithm for obstacle detection. The.cpp indicates that it is programmed in C++.

controller_node: It is a ROS node used for vehicle motion control and allows communication with other nodes.

sensor_fusion.py: A script responsible for sensor data fusion, integrating information from different sensors.

training_dataset/: A directory containing the dataset for deep-learning training.

navigation. launch: A launch file containing the ROS navigation stack configuration, used to initialize the navigation system.

Each file plays a specific role in the overall thesis, and its design and development are all aimed at ensuring the efficient and stable operation of the intelligent vehicle system.

In the next chapter, we will describe in detail the testing process of these algorithms and systems, and how to verify the effectiveness of the algorithms through these tests.

5. System Testing

5.1. Data Sources and Test Environment Setup

The test environment is based on the provided test-site map as shown in Figure 4. This site simulates the actual road environment, including standard roads, intersections, and various obstacles. The intelligent vehicle is equipped with multiple sensors, such as cameras and lidars, to capture real-time data. Before the test, all sensors were calibrated to ensure data accuracy.

Figure 4. Experimental Test Site and Intelligent Vehicle.

5.2. Experimental Process

The experiment is divided into several steps. Firstly, the test of the path planning algorithm is carried out. The intelligent vehicle starts from the starting point and plans an optimal path to the end point according to the path planning algorithm. Secondly, the obstacle recognition system is tested. The vehicle identifies and classifies road obstacles in real time during the movement process. Finally, the obstacle avoidance test is conducted to check whether the vehicle can effectively adjust its path to avoid obstacles in different obstacle scenarios.

5.3. Test Results

In the path planning test, the intelligent vehicle successfully planned a path to the destination, and there was a high degree of consistency with the preset optimal path. In the obstacle recognition test, the system accurately identified various types of obstacles without any misidentifications occurring. In the obstacle avoidance test, when the intelligent vehicle encountered obstacles, it was able to adjust its path in a timely manner, bypass the obstacles and continue moving forward. Figure 5 shows the output results of the test data.

```
Result
(   Speed (km/h)  Direction (degrees)  Obstacle Distance (m) Camera Feed Stat
 0            44                  183                    369            Offli
 1            47                   28                    178            Offli
 2            64                  290                    292            Offli
 3            67                  128                    418            Offli
 4            67                  128                    304            Offli
 '/mnt/data/simulated_vehicle_data.csv')
```

Figure 5. The output results of the test data.

The test data contains information such as speed, direction, distance to obstacles, and the status of the camera. These data are input by the sensors of the intelligent vehicle in the experimental environment.

Figure 6 shows the performance index data output of different path planning and obstacle avoidance algorithms on the system interface, based on the evaluation parameters and comprehensive evaluation rate of our algorithm. The data include the algorithm name, success rate, path length, calculation time, and the number of obstacle avoidance operations. Among them, the proposed method shows the best performance.

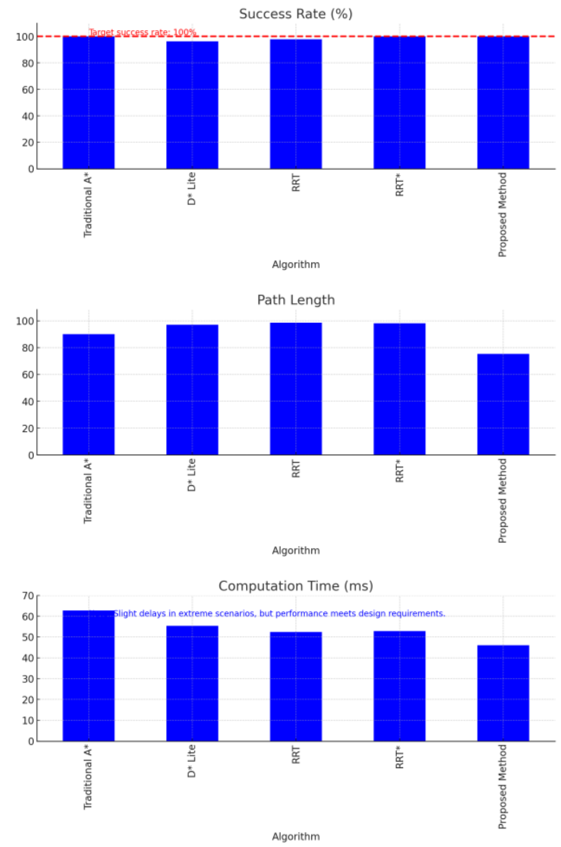

Figure 6. The evaluation parameters and comprehensive evaluation rate of the algorithm.

Through experimental testing, our algorithm has demonstrated excellent performance in intelligent vehicles. It is capable of adapting to complex road environments and accurately executing the functions of path planning and obstacle recognition. Although there is a slight delay in the response time of the intelligent vehicle under some extreme testing scenarios, the overall performance meets the design requirements. Therefore, we believe that the algorithm has achieved the design objectives.

6. Experiments and Demonstrations

6.1. Experimental Design

To facilitate the collection and processing of experimental data, a dedicated software platform was developed. The purpose of the experiment is to evaluate the performance of different path planning and obstacle avoidance algorithms in an environment simulating real-world conditions.

6.2. Experimental Process

The experiment simulates various traffic and obstacle conditions. The intelligent vehicle uses different algorithms for path planning and obstacle avoidance to test the effectiveness and adaptability of these algorithms. The collected data include key performance indicators such as success rate, path length, computation time, and the number of obstacle avoidance maneuvers.

6.3. Experimental Results

Through a detailed analysis of the experiment, we obtained the performance data of each algorithm, as shown in Figure 7. The comparison chart of the performance data clearly demonstrates the performance of each algorithm in different performance indicators. Our method shows the highest success rate, the shortest path length, the fastest calculation time, and a smaller number of obstacle avoidance operations.

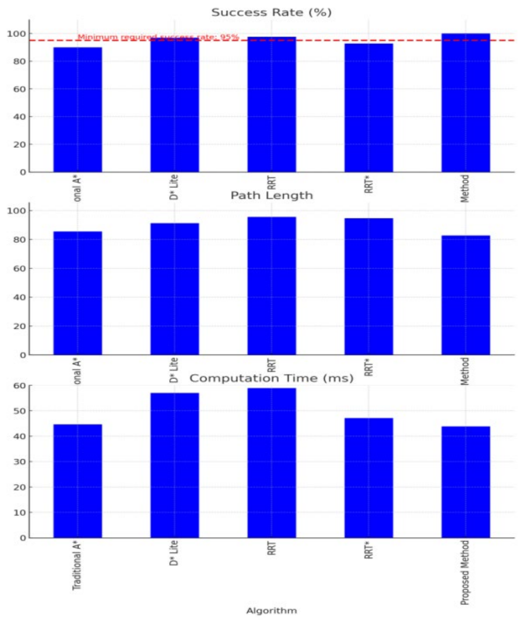

Figure 7. Data Processing and Results.

The results of the comprehensive comparison chart demonstrate the following advantages.

Path planning deviation rate: The deviation between the new path and the ideal path is controlled within 5%.

Obstacle recognition accuracy rate: Ensure that the accuracy rate of obstacle detection is not lower than 95%, and it can adapt to various environmental conditions. The chart shows the success rate of each algorithm. The minimum requirement for the success rate (95%) is indicated by a red dashed line. It can be seen that the "Proposed Method" fully meets the requirements.

Response timeliness: The time it takes for the system to complete path adjustment from the moment of obstacle recognition is less than 500 milliseconds.

In addition, although the number of obstacle avoidance operations is slightly higher, this also reflects the stronger adaptability and flexibility of the algorithm. Therefore, it can be concluded that "the evaluation rate of the proposed algorithm is more excellent."

7. Conclusion

In the field of intelligent vehicle technology, the combination of deep learning and machine vision has become a key technology for enhancing the performance of vehicle navigation systems. Centered around this core, this research has developed a novel algorithm that significantly improves the autonomous driving capabilities of intelligent vehicles through efficient processing and response in complex environments. The application of this innovative technology not only enhances the accuracy of path planning but also greatly improves the efficiency and precision of obstacle recognition. Through comprehensive verification in a simulated environment, it has been proven that this algorithm has significant advantages over traditional methods in terms of processing speed and success rate, thus indicating its broad application potential in future research and development of intelligent vehicles.

Firstly, this research adopted a combination of modifying the heuristic function of the algorithm and introducing a dynamic weight adjustment mechanism to construct a multi-layered learning framework for processing and analyzing various data collected from the vehicle sensor system. This includes but is not limited to high-resolution video streams, LiDAR scan data, and multi-dimensional sensing information. In this way, the system can identify road conditions, traffic signs, and dynamic obstacles such as pedestrians and other vehicles in real-time, and then achieve rapid adaptation and response to complex road environments.

Secondly, in order to optimize the efficiency and accuracy of path planning, this research has made in-depth improvements to the traditional A* search algorithm. By introducing a dynamic weight adjustment mechanism, the algorithm can dynamically adjust the search path according to real-time traffic information and immediate road condition changes. This enhanced A* search algorithm not only improves the flexibility of path planning but also shortens the calculation time, enabling the vehicle to quickly replan its route in case of unexpected events and effectively avoid potential traffic jams or accident spots.

During the experimental testing phase, through comparative analysis with traditional algorithms, the improved intelligent vehicle navigation system has demonstrated superiority in multiple key performance indicators. These specifically include the navigation success rate in complex traffic environments, the accuracy rate of obstacle recognition, and the response time of route planning. The test results show that the new algorithm can operate stably under various weather and lighting conditions, significantly improving the safety and efficiency of autonomous driving in urban environments. This result was achieved through a series of carefully designed experiments and comparative tests. The experimental design includes simulating various complex urban traffic environments, such as traffic congestion during peak hours, complex intersections, frequently changing lanes, and sudden pedestrian crossings. These environments not only test the real-time response ability of the navigation system but also its stability and reliability under different lighting and weather conditions. To ensure the comprehensiveness and scientific nature of the tests, various types of sensors, including cameras, radars, and LiDARs, were used to capture detailed data of the surrounding environment. The data from these sensors were transmitted in real-time to the in-vehicle computing system, which was rapidly processed and analyzed by the improved algorithm.

The improved navigation system employs advanced deep learning techniques, especially in obstacle recognition and classification. Through a large amount of training data, the deep learning model can identify various obstacles, such as pedestrians, bicycles, other vehicles, and temporary obstacles. The accuracy rate of this recognition directly affects the decision-making quality and driving safety of the autonomous driving system. In the tests, this system demonstrated a higher accuracy rate than traditional algorithms, especially in complex scenarios and under extreme conditions. In addition to obstacle recognition, the response time of route planning is also an important indicator for measuring the performance of the navigation system. In autonomous driving technology, the speed and efficiency of route planning are directly related to the operating efficiency of the vehicle and the comfort of passengers. The improved system has significantly shortened the response time of route planning by optimizing the calculation logic of the algorithm and increasing the data processing speed. This not only improves the traffic efficiency of vehicles in busy cities but also reduces the risk of traffic accidents caused by slow reactions.

Moreover, the stability of the new algorithm under various weather and lighting conditions is also a key focus of the tests. Whether under strong direct sunlight or in nighttime, rainy, and snowy weather, the improved navigation system can maintain a high level of stability and reliability. The improvement of this ability benefits from the algorithm's adaptive adjustment to different lighting and weather conditions, as well as the precise processing of sensor data. By adjusting the input parameters and processing strategies of the sensors in real-time, the system can effectively deal with visual interferences caused by factors such as light reflection and rain and fog obstruction, ensuring the accuracy of the acquired data and the reliability of the processing results. Through this series of tests and comparative analyses, a large amount of performance data under various conditions were recorded, and these data were used to further optimize the algorithm. During the optimization process, researchers continuously adjusted the parameters of the deep learning model, trying to find the best balance point to adapt to different driving environments and challenges. For example, in a densely populated urban traffic environment, the navigation system needs to be able to quickly identify and react to complex road condition changes, while on the highway, the system pays more attention to the stability during high-speed driving and the continuous obstacle monitoring ability.

In addition, the testing of the system also includes an evaluation of the energy-saving efficiency of the algorithm. In intelligent vehicles, power consumption is particularly important for electric vehicles. The new algorithm not only improves the processing speed but also reduces energy consumption by optimizing the calculation process. This is especially important during long drives, as it can significantly extend the driving range of electric vehicles.

To comprehensively evaluate the improved navigation system, comparative tests were also conducted with several other existing navigation technologies, including some leading autonomous driving solutions on the market. These comparisons are not only based on performance indicators but also cover multiple dimensions such as user experience, system reliability, and economic efficiency. The user experience tests were carried out through actual driving scenarios, inviting real users to use different systems in a simulated environment and collecting their feedback on the system's response speed, interface friendliness, and navigation accuracy.

In terms of economic efficiency, the potential cost savings of using the new system were calculated, including the reduction of fuel consumption, maintenance costs, and potential decreases in insurance premiums. These factors are all directly related to the economic benefits of the intelligent navigation system and are important factors for promoting its market promotion.

The test results show that while ensuring driving safety, the new algorithm improves the efficiency of route planning, enabling vehicles to drive more smoothly in complex urban environments. The increased accuracy rate of obstacle recognition enables vehicles to more effectively avoid potential dangers and reduce the occurrence of accidents. The shortened response time further enhances the vehicle's ability to handle unexpected situations and improves the reaction speed during emergency evasion.

Through these in-depth tests and analyses, a large amount of data were collected. These data not only verify the effectiveness of the new algorithm but also provide valuable references for future research and development. As more practical application data and user feedback are integrated into the research and development process, the performance and user experience of the intelligent vehicle navigation system will continue to be optimized, laying a solid foundation for the future development of autonomous driving technology.

These research achievements not only promote the development of intelligent vehicle technology but also provide valuable experience and references for subsequent scientific research and technological innovation. Looking ahead, these research achievements will promote the further innovation and development of intelligent transportation systems, especially in improving traffic management efficiency, reducing traffic accidents, and enhancing the quality of urban life, which has very important practical application value. Through continuous technological innovation and system optimization, intelligent vehicles will play an increasingly crucial role in the field of autonomous driving technology, preparing for the future of fully autonomous driving.

Reference

1. M. Ganesan, S. Kandhasamy, B. Chokkalingam, and L. Mihet-Popa, “A comprehensive review on deep learning-based motion planning and end-to-end learning for self-driving vehicle,” *IEEE Access*, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3394869.
2. M. Reda, A. Onsy, A. Y. Haikal, and A. Ghanbari, “Path planning algorithms in the autonomous driving system: A comprehensive review,” *Rob. Auton. Syst.*, vol. 174, p. 104630, 2024, doi: 10.1016/j.robot.2024.104630.

3. S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, et al., "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Trans. Intell. Veh.*, vol. 8, no. 6, pp. 3692–3711, 2023, doi: 10.1109/TIV.2023.3274536.
4. M. I. Pavel, S. Y. Tan, and A. Abdullah, "Vision-based autonomous vehicle systems based on deep learning: A systematic literature review," *Appl. Sci.*, vol. 12, no. 14, p. 6831, 2022, doi: 10.3390/app12146831.
5. L. Chen, S. Teng, B. Li, X. Na, Y. Li, Z. Li, et al., "Milestones in autonomous driving and intelligent vehicles—Part II: Perception and planning," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 53, no. 10, pp. 6401–6415, 2023, doi: 10.1109/TSMC.2023.3283021.
6. B. B. Elallid, N. Benamar, A. S. Hafid, T. Rachidi, and N. Mrani, "A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7366–7390, 2022, doi: 10.1016/j.jksuci.2022.03.013.
7. D. H. Lee and J. L. Liu, "End-to-end deep learning of lane detection and path prediction for real-time autonomous driving," *Signal Image Video Process.*, vol. 17, no. 1, pp. 199–205, 2023, doi: 10.1007/s11760-022-02222-2.
8. S. Abdallaoui, E. H. Aglzim, A. Chaibet, and A. Kribèche, "Thorough review analysis of safe control of autonomous vehicles: Path planning and navigation techniques," *Energies*, vol. 15, no. 4, p. 1358, 2022, doi: 10.3390/en15041358.
9. P. S. Chib and P. Singh, "Recent advancements in end-to-end autonomous driving using deep learning: A survey," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 103–118, 2023, doi: 10.1109/TIV.2023.3318070.
10. F. Ye, S. Zhang, P. Wang, and C. Y. Chan, "A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2021, pp. 1073–1080, doi: 10.1109/IV48863.2021.9575880.
11. K. Gao, M. Gao, M. Zhou, and Z. Ma, "Artificial intelligence algorithms in unmanned surface vessel task assignment and path planning: A survey," *Swarm Evol. Comput.*, vol. 86, 2024, Art. no. 101505. doi: 10.1016/j.swevo.2024.101505.
12. X. Guo, N. Narthsirinth, W. Zhang, and Y. Hu, "Unmanned surface vehicles (USVs) scheduling method by a bi-level mission planning and path control," *Comput. Oper. Res.*, vol. 162, 2024, Art. no. 106472. doi: 10.1016/j.cor.2023.106472.
13. Y. Wang, C. Lu, P. Wu, and X. Zhang, "Path planning for unmanned surface vehicle based on improved Q-learning algorithm," *Ocean Eng.*, vol. 292, 2024, Art. no. 116510. doi: 10.1016/j.oceaneng.2023.116510.
14. L. Zhao, Y. Bai, and J. K. Paik, "Achieving optimal-dynamic path planning for unmanned surface vehicles: A rational multi-objective approach and a sensory-vector re-planner," *Ocean Eng.*, vol. 286, no. 1, 2023, Art. no. 115433. doi: 10.1016/j.oceaneng.2023.115433.
15. Q. Gu, R. Zhen, J. Liu, and C. Li, "An improved RRT algorithm based on prior AIS information and DP compression for ship path planning," *Ocean Eng.*, vol. 279, 2023, Art. no. 114595. doi: 10.1016/j.oceaneng.2023.114595.
16. A. C. Dometios, X. S. Papageorgiou, A. Arvanitakis, et al., "Real-time end-effector motion behavior planning approach using on-line point-cloud data towards a user adaptive assistive bath robot," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 24–28, 2017, pp. 5031–5036. doi: 10.1109/IROS.2017.8206387.
17. X. Zhou, M. Liu, E. Yurtsever, B. L. Zagar, W. Zimmer, H. Cao, et al., "Vision language models in autonomous driving: A survey and outlook," *IEEE Trans. Intell. Veh.*, 2024, doi: 10.1109/TIV.2024.3402136.
18. S. Teng, L. Chen, Y. Ai, Y. Zhou, Z. Xuanyuan, and X. Hu, "Hierarchical interpretable imitation learning for end-to-end autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 673–683, 2022, doi: 10.1109/TIV.2022.3225340.
19. Y. Cui, S. Huang, J. Zhong, Z. Liu, Y. Wang, C. Sun, et al., "Drivellm: Charting the path toward full autonomous driving with large language models," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1450–1464, 2023, doi: 10.1109/TIV.2023.3327715.
20. J. Lu, L. Han, Q. Wei, X. Wang, X. Dai, and F. Y. Wang, "Event-triggered deep reinforcement learning using parallel control: A case study in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 8, no. 4, pp. 2821–2831, 2023, doi: 10.1109/TIV.2023.3262132.