Journal of Organizational and End User Computing
Volume 34 • Issue 10

An Evaluation System Based on User Big Data Management and Artificial Intelligence for Automatic Vehicles

Pei Shanshan, Macau University of Science and Technology, China
Ma Chao, Macau University of Science and Technology, China
Zhu Haitao, Beijing Smarter Eye Technology Co., Ltd., China
Luo Kun, Hisilicon (Shanghai) Technologies Co., Ltd., China*

ABSTRACT

As artificial intelligence is widely used in the automatic driving systems, the safety evaluation of automatic vehicles is considered to be the most important demand. Under this context, in this paper, an evaluation system, which is composed of several important evaluation projects, is proposed based on big data. These indicators reflect the performance of the automatic driving system. In addition, the principle of the evaluation index and the data management scheme are explained. In terms of the evaluation projects, the online test and the offline test are included, when the former focuses on the function design that is as expected, while the latter aims to ensure the actual driving experience of the automatic driving system. The evaluated results provide optimization direction of the algorithm index. Furthermore, based on AI technology and user big data management, the system saves test costs and guarantees algorithm performance and system stability.

KEYWORDS

AI Technology, Automatic Driving, Evaluation System, User Big Data Management

INTRODUCTION

Traditional automotive passive safety technologies to reduce accident hazards after an accident occurs, such as bumpers, seat belts, and airbags, cannot avoid collisions. In recent years, advanced driver assistance systems (ADAS) have received increasing attention. The ADAS is an active safety technology that uses sensor environment perception capabilities. It detects possible dangerous situations in advance through perceptual data analysis and takes actions in advance to reduce or even avoid collision damage. The ADAS is the foundation of autonomous driving and can be used with the Internet of Things (IoT; Zhou, Yang, et al., 2021). Common subdivision systems include adaptive cruise control (ACC), automatic emergency braking (AEB), the blind spot detection system (BSD), the forward collision warning system (FCW), the lane departure warning (LDW), the heads-up display (HUD), the automotive night vision system (NVS), the intelligent vehicle speed control (ISA), the intelligent headlight control (AFL), the parking assist system (PA), the pedestrian detection system (PDS), and the traffic sign recognition (TSR). This study described the evaluation system of the automatic emergency braking algorithm (AEB). The AEB function (Kim et al., 2019) is implemented to

DOI: 10.4018/JOEUC.309135 *Corresponding Author

measure the distance between the vehicle and the preceding vehicle, or the vehicle and other obstacles, while it adopts early warning and automatic braking when the distance is less than the safe distance.

The application scenarios of the automatic vehicles (Reddy et al., 2018) and in smart cities (F. Wang et al., 2022) are rich and complex, such as the diversity of weather, the change of light, and the irregular movement of vehicles and pedestrians. Therefore, in the development stage, the function test needs to go through thousands of kilometers of test distance to make a reliable evaluation of the performance. Under such testing requirements, traditional testing techniques are time-consuming and expensive. Considering that, as big data (Zhang et al., 2017) is featured with a large amount of data and many types of data. The application of user big data (Zhu, Yang, et al., 2021) testing has an excellent performance in testing comprehensiveness and diversity, and by constructing test cases, it is possible to achieve repeated testing of key scenarios.

BACKGROUND

Safety issues have always been the focus of research on autonomous driving, so that the evaluation system has received increasing attention this year. J. Han et al. (2016) proposed a vision-based FCW and AEB system. In this paper, the author installed a laser scanner synchronously with the vision sensor, and the perception data of the laser scanner was used as the ground truth to evaluate the detection results of the system, thus being able to method can evaluate the accuracy of the detected distance. Lim et al. (2017) presented an integrated system of vehicle detection and distance estimation for a real-time AEB system based on stereo vision (Xie et al., 2017). Apart from that, the authors here use the KITTI dataset, which was co-founded by Karlsruhe Institute of Technology in Germany and Toyota American Institute of Technology, as a benchmark to evaluate the detection accuracy of the verification function and estimate the processing time of the algorithm on the Titan X platform and the NVIDIA TXI platform. An et al. (2019) put forward a framework for evaluating ADAS performance when considering four aspects of efficiency, safety, cost, and driver acceptance. Furthermore, macro-level factors (i.e., traffic volume and driving time) and micro-level indicators (i.e., reaction time and driver stress) were assessed. Due to limited time and experimental conditions, only ACC was tested as the main function of ADAS. However, the evaluation methods of the above systems consume a lot of time and human resources. In functional development, the evaluation scenarios are not sufficient, resulting in the risk of functional defects. Therefore, this study presents an evaluation system based on big data (Qi et al., 2015), aiming to reduce test costs and improve test reliability.

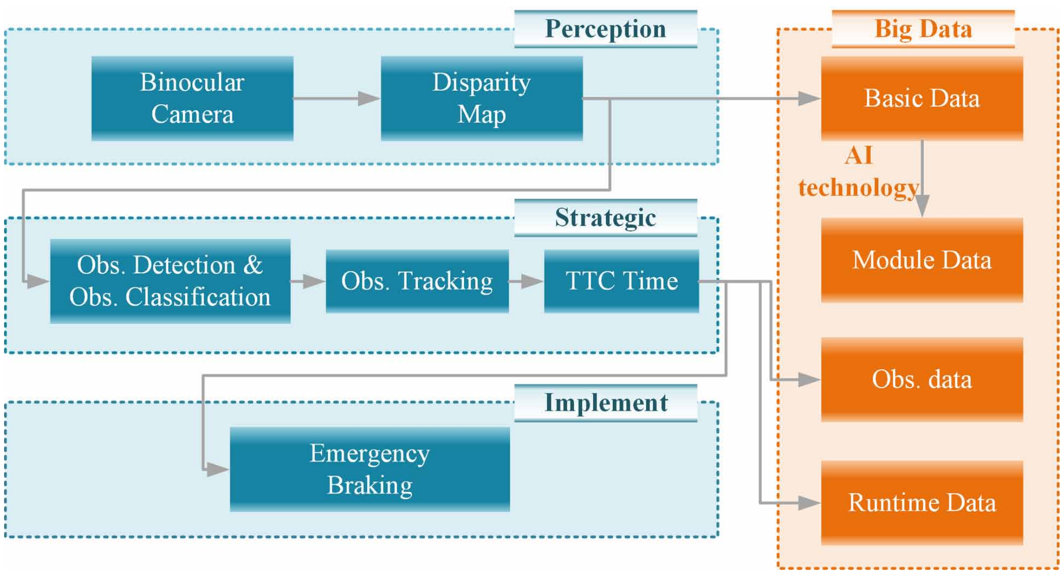
Nowadays, big data technology (E. Sun et al., 2021) and AI technology (X. Wu et al., 2022) have made rapid progress, and there are an increasing number of studies and application scenarios. George et al. (2014) proposed that the definition of big data lies in the fine-grained nature of the data itself. The focus of big data technology (Zhou, Li, & Liang, 2020) is not on the results but rather on acting on every micro-module in the process of function realization. These modules can be mapped onto factors that contribute to the success or failure of the function, while these modular tests provide rich functional microdata for helping predict and evaluate functional performance. R. Han et al. (2014) put forward a big data benchmarking system; when introducing the requirements and challenges of big data benchmarks in detail, the problems that should be solved during the execution process for fair, efficient, and successful benchmarking, and the management of big data benchmark methods. Liu et al. (2014) introduced a performance testing method in big data applications (J. Wu et al., 2022). The author focuses on the application analysis and test model of big data test design to ensure the efficient operation of the performance test method and demonstrates the effect of automated performance tests through the network public opinion monitoring system. So et al. (2019) presented a collision scene reconstruction method adopting big data technology (Salman et al., 2022). By extracting the accident description in the traffic accident report, the scene text weight was analyzed, and the test scene was constructed according to the weighted words, which helps generate more comprehensive test scenarios based on the systematic analysis of large amounts of crash data.

This study introduces a kind of autonomous driving based on user big data management and artificial intelligence and focuses on the application of the user big data management and artificial intelligence in the functional performance test, so as to ensure the sufficiency and accuracy of evaluation. In the next section, we introduce data collection and data preprocessing in big data management. Following that, we explain the principle and structure of the evaluation system and then provide experiments and analysis to support that the proposed method is effective. Finally, we summarize our proposed evaluation scheme.

USER BIG DATA MANAGEMENT

In this section, the multi-level data collection method is introduced for the purpose of building a user big data management system in the binocular vision-based autonomous driving (Xie et al., 2022) framework. Figure 1 shows the system framework for data construction and autonomous driving.

Figure 1. System framework for data construction and autonomous driving



The autonomous driving system is designed through the logic of perception, strategy, and execution. Vision sensors are the perception core of the autonomous driving system and are responsible for observing the environment around the vehicle. The vision sensor obtains the image data of the vehicle's front scenery (Ma et al., 2021) and calculates the three-dimensional point cloud (F. Wang et al., 2021) information through the traditional matching algorithm (Long, Xie, Mita, Ishimaru, & Shirai., 2014), which is the process of the perception module. The strategic module is completed by the controller that detects obstacles based on the three-dimensional point cloud information, obtains the position information and category information of the obstacles, adds a tracking module to the detection results of the obstacles, acquires the stable spatial position, speed, and acceleration of the obstacles, and finally calculates the time-to-collision (TTC) value of the target obstacle. When there is a risk of collision, the autonomous driving system sounds an alarm and the braking system is activated.

In terms of hardware structure, the focal length of the binocular lens is 8.26 mm, while the baseline is 12 cm, and the picture element is 4.2×10^{-3} mm. The horizontal field of view (FOV) and vertical FOV of the camera are 30° and 20°, respectively, and the image resolution is 1280 × 720. In addition, the computing chip is Xilinx Z-7020, which integrates two ARM Cortex-A9 processors and a field-programmable gate array (FPGA; X. Wang et al., 2017). Other parameters are shown in Table 1.

Table 1. System hardware parameters

Items	Parameters	Items	Parameters
Baseline	120 mm	Focus length	8.26 mm
Picture element	4.2×10^{-3} mm	Resolution	1280 × 720
Horizontal FOV	30°	Vertical FOV	20°

DATA COLLECTION

Big data collection management is divided into four data modules. The first module is the *basic data*, which is the data collected by the perception module, including the left image and right images collected by the binocular vision sensor and the disparity map generated by the matching algorithm (Long, Xie, Mita, Nikejad et al.,2014). The basic data is used for the logic design verification in the functional development stage. The second module is module data. The module data generated by AI technology using the basic data is an indispensable source of truth in the evaluation system. Here, it should be noted that in this paper, the deep learning model (Xu et al., 2021) is adopted to train the basic data and can obtain the category information, position information, tracking information, and TTC of the obstacles, which are used as standard values in the evaluation system to evaluate the module effect of automatic driving. The third module is the collection of *obstacle data* to evaluate the detection result consistency between the development platform and the actual application platform in the system. The obstacle data includes the type of obstacle, the location of the obstacle, and the TTC. The last collection module is *runtime data*, which is used to verify the real-time effect of the function (C. Hu et al., 2021).

Because the binocular vision sensor used by the autonomous driving system is adopted to monitor the driving environment ahead in real time, it is usually installed on the front windshield. To ensure that the driver's sight is not blocked, the volume of the hardware platform should not be too large, and the autonomous driving hardware design should capture the parameters of the vehicle motion, including the vehicle forward scene picture, vehicle speed information, collection time and body attitude information, etc. Apart from that, the hardware platform is connected to a personal computer (PC) through a universal serial bus (USB 3.0) high-speed data interface to transmit image data in real time and build a data set to be processed.

DATA PREPROCESSING

The data obtained in real time cannot be directly imported into the database for use. In this case, the data should be preprocessed to ensure the quality and uniformity of the data (Qi et al., 2021), as follows.

1. **Vehicle speed check:** The hardware platform of the controller can obtain the real vehicle speed through the controller area network (CAN) bus.
2. **The collection time check:** The collection device starts timing when it is powered on. After the device is connected to the computer, the timing can be synchronized with the world time. The

normally acquired timestamp can correspond to the acquisition time block after being converted by the Unix timestamp.

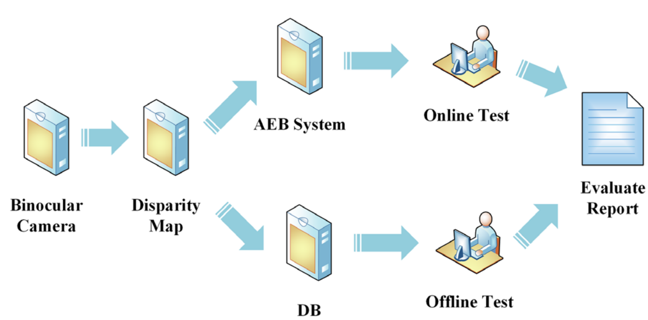
3. **File number check:** The data collected by the binocular vision sensor includes the left image, the right image, and the disparity image at the same time. File loss or data anomalies that may be caused by unstable interface connections should be checked.
4. **Data attribute confirmation:** The camera has a variety of specifications of different focal lengths, and the camera parameters are generated during calibration, and the actual camera parameters of the collected data need to be confirmed for data management.
5. **Data classification:** Scene tags are added to the collected data, and test cases are made. The tags involved are:
    a. Test scene participant *type labels*, including *cars*, *pedestrians*, *bicycles*, and *non-standard obstacles*.
    b. Test scene participant *location labels*, including *front*, *rear*, *left*, *right*, *front-left*, *rear-left*, *front-right*, and *rear-right*.
    c. Test scene participant *status labels*, such as *static*, *deceleration*, *acceleration*, *cut-in*, *cut-out*, and *traverse*.
    d. Test scene *road labels*, including *straight roads*, *curves*, *expressways*, *urban roads*, *country roads*, *mountain roads*, *multi-lane roads*, *single-lane roads*, and *tunnels*.
    e. Test scene *environment labels: sunny*, *overcast*, *rain*, *snow*, *fog*, *backlight*, and *night*.

Data preprocessing is the basis for building user big data management. The incorrect data structure or format often consumes a lot of storage space. Through data preprocessing, the quality of the data is ensured, and the storage space occupancy is reduced to a great extent, which in turn generates cost benefits. Besides this, data preprocessing guarantees that data is injected without errors so that basic verification functions can function normally.

EVALUATION SYSTEM FRAMEWORK

The evaluation work of the autonomous driving system is accompanied by the entire process of function development. The evaluation and development work needs to be carried out simultaneously. By integrating the evaluation work into the development to a certain extent, it is possible to observe whether the effect of the function module has reached the expectation very early and obviously reduces the chance of some errors. Actually, the purpose of the evaluation work is to check the effective operation of the function. To achieve the purpose, the evaluation system proposed by us consists of the offline test module and the online test module to complete the evaluation work together, as shown in Figure 2.

Figure 2. Evaluation system framework

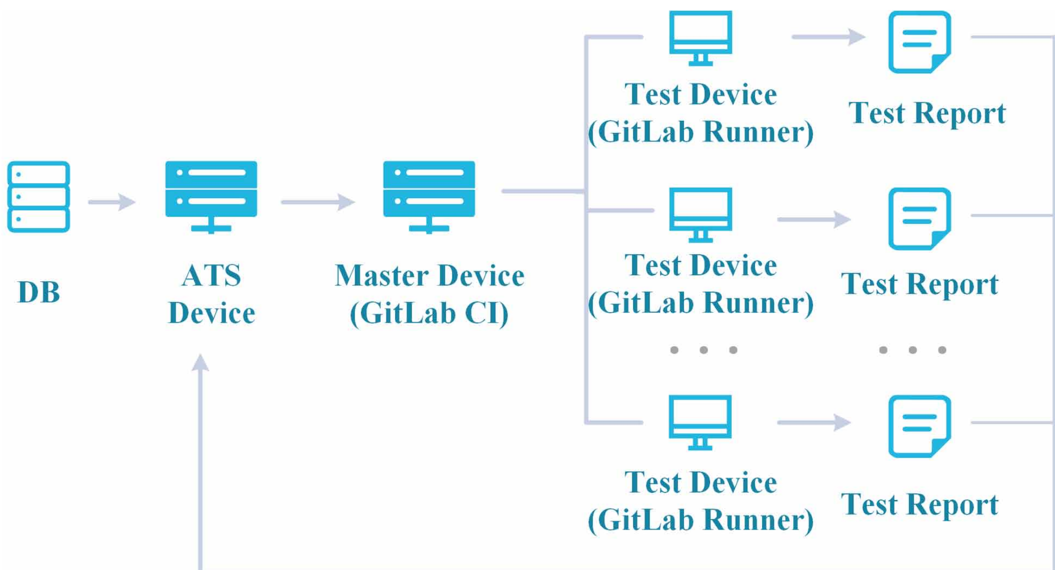


The offline test module uses the database (DB) we built to conduct local tests for the purpose of evaluating whether the function is complete and checking whether there are obvious bugs. The online test module is the actual driving environment test. Due to the rich and complex application scenarios of autonomous driving, the online test module creates a more comprehensive evaluation while ensuring the user's subjective experience.

OFFLINE TEST

The evaluation system that applies the autonomous driving of vision processing can be deployed to multiple computer nodes using distributed testing (Xu et al., 2022) to achieve test resource sharing, decentralized execution, centralized management and control, collaborative work, and load balancing. Figure 3 is the frame diagram of distributed testing.

Figure 3. Distributed testing framework of the offline test



In the main workflow of distributed testing, the user provides the content to be evaluated and the relevant test case data path to the automatic test system (ATS) device where the test framework is deployed (the path of the test case comes from the DB), and then starts and coordinates the performance evaluation experiment through the master device. In terms of the master device, it coordinates the test tasks of the test device through GitLab continuous integration (CI), and multiple cooperating test devices cooperate with GitLab CI to run different evaluation tasks through the deployed GitLab Runner. Other than that, the ATS equipment collects all distributed test results and gives an overall evaluation result of offline test performance.

The autonomous driving system is composed of multiple function modules, each of which is designed by different developers, whose understanding and programming logic are different. Therefore, the function modules need to be distributed and tested. Distributed testing can not only improve test efficiency but also verify module defects, accurately locate function errors, and ensure that the overall function can work uniformly.

Offline testing is divided into four sub-tasks to ensure the reliability of the evaluation. These four modules are evaluated from three evaluation directions. Firstly, the early warning effect test is performed to verify the final result of the autonomous driving system. Secondly, evaluation of tracking algorithm effect, evaluation of detection algorithm, and evaluation of classification algorithm effect are fine-grained effect evaluations. Finally, the algorithm stability evaluation guarantees the comprehensiveness and stability of the evaluation.

EARLY WARNING EFFECT TEST

The AEB system is the core function of autonomous driving, mainly including FCWs and automatic braking (AB), when the former is applied to the scene of automatically detecting obstacles ahead, predicting the danger of a forward collision, and issuing early warning signals in time to remind the driver. Therefore, this stage of the FCW is an essential module of the autonomous driving system. Thus, accurate early warning signals are the core test points.

The early warning signal of the system comes from the calculation of the TTC, commonly calculated as $TTC = (s - d) / V$, where s is the distance between the vehicle and the obstacle ahead, d refers to the safe stopping distance, and V denotes the relative speed of the vehicle and the forward obstacle, when the TTC is less than the preset threshold, an early warning signal is triggered.

Manual labeling is used as the true value. Firstly, the first-level labeling is performed, and the test cases are divided into two categories, namely the *triggering warning* category and the *non-triggering warning* category. Then, the second-level labeling is performed, performing fine-grained marking on test cases that have been assigned the *triggering warning* category. In specific operations, it marks the range of image sequences that should trigger warnings. In the actual test, the first-level labeling is used to verify the missing detection and false detection, as well as the early-warning accuracy of the second-level labeling verification function.

EVALUATION OF TRACKING ALGORITHM EFFECT

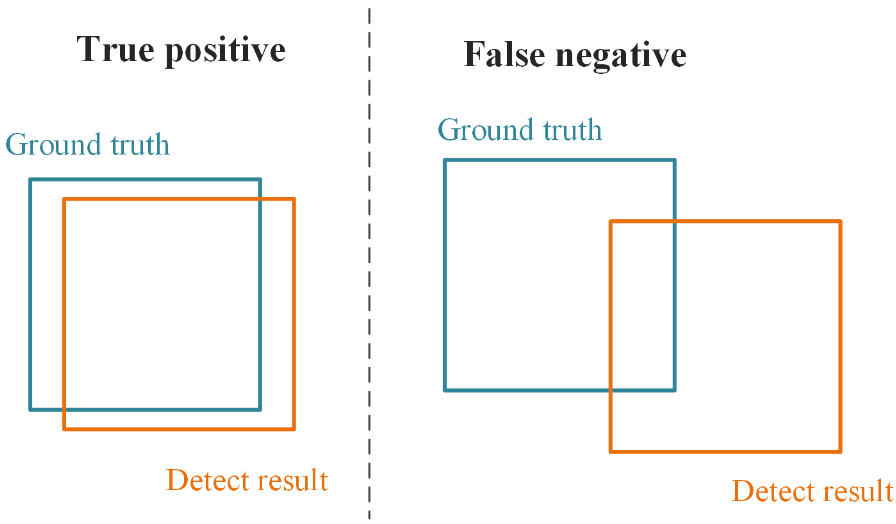
The early warning signal of the system comes from the calculation of the TTC, which depends on the position accuracy of the forward obstacle. Besides this, tracking is the module that ensures the stable spatial position, speed, and acceleration of the forward obstacle output. The evaluation system uses a deep learning network (Z. Sun et al., 2017) to train the test cases, approximate the results as the ground truth (Zhou, Li, & Liang, 2020, and compare the differences between the tracking results of the deep learning network and the actual algorithm tracking results (Zhou, Liang, et al., 2021). The specific evaluation process is as follows.

The first step is that both the deep learning network and the actual tracking module run the test case and retain the tracking result information that includes the position information, category, and tracking frame number of the target obstacle (T. Fan, 2020). The second step is to count the number of target obstacles with an intersection over union (IOU) ratio greater than 0.5 in the actual tracking module and the deep learning network tracking module, which is referred to as the NMT indicator, as shown in Equation 1.

$$NMT = \sum_{i-0}^{N} NUM, NUM = \begin{cases} 1, IOU > 0.5 \\ 0, else \end{cases} \tag{1}$$

The variable N indicates the total number of all target obstacles, and IOU is often used to measure the accuracy of target objects in detection tasks. The specific implementation is the intersection ratio of the actual tracking module obstacle frame and the deep learning network obstacle frame. As shown in Figure 4, the two boxes represent the ground truth box and the detection box, respectively, where Area_1 is the intersection of the two boxes, and Area_2 is the union of the two boxes. The ratio of the area of the intersection to the area of the union is employed to measure the deviation of the detection result from the true value. The larger the value of IOU, the smaller the deviation between the detection result and the true value. We set the threshold of IOU to 0.5 to judge whether the detection result is qualified. The variable *NUM* is the number of obstacle boxes with an IOU greater than 0.5.

Figure 4. IOU explanation diagram



The third step is to count the average ratio of the actual obstacle tracking frames to the deep learning obstacle tracking frames among all NMT objects, which we record as the MAR indicator.

$$MAR = \frac{1}{NMT}\sum_{i-0}^{NMT}\frac{age_{stereo}}{age_{d}} \tag{2}$$

The NMT value is the statistical result of the second step, while age_{stereo} and age_{dl} are the tracking frame number information of the actual tracking module and the deep learning network, respectively.

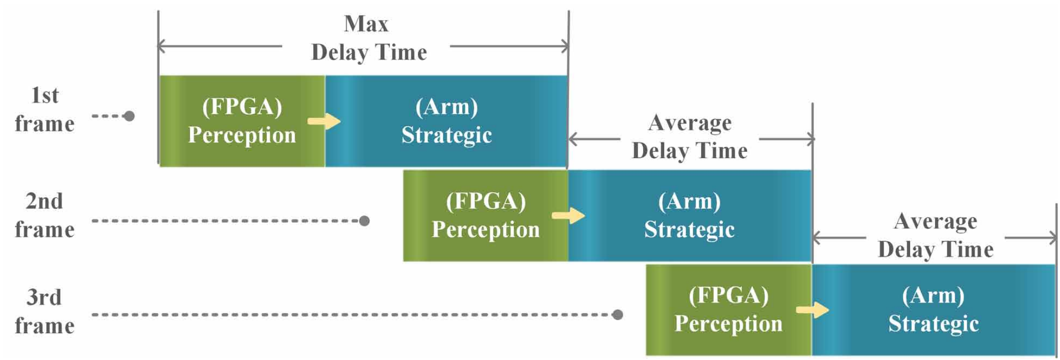
The larger the number of the NMT indicators, the more matching objects (Kasthuri et al., 2019) the actual tracking module and the deep learning network tracking barrier have and the closer the MAR indicator is to 1, while the closer the actual tracking module and the deep learning network tracking frames are. Through the above evaluation method, the effect of the tracking module can be observed according to the evaluation index. At the same time, this index can be used in the version iteration stage of the research and development process, which reflects whether the new version of the tracking strategy has significantly enhanced the effect.

EVALUATION OF DETECTION ALGORITHM AND CLASSIFICATION ALGORITHM EFFECT

The input source of the tracking module is the detection information of the target obstacle (the location and category of the target obstacle), and the stable detection result is beneficial to the accuracy of the tracking result. In addition, the evaluation system compares the detection results of the deep learning network as ground truth. The specific evaluation process is as follows. In the first step, the deep learning network and the actual detection module are tested on the same data. The detection position and the category of the target obstacle frame are recorded. Among them, the category of the target obstacle frame includes pedestrians, vehicles, cyclists, and other non-standard obstacles. The second step is to count the Mbox indicators. The specific operation sets the IOU threshold to distinguish whether the detection is successful or not. We set this value to 0.2 in the implementation. As shown in Figure 5, IOU greater than 0.2 is considered a true positive, while less than 0.2 is considered a false negative, and the statistical mark is the mean IOU of the true positive obstacle box, as shown in Equation 3.

$$Mbox = \frac{1}{n_{IOU>0.2}} \sum_{i-0}^{n_{IOU>0..2}} IOU \tag{3}$$

Figure 5. Schematic diagram of true positive and false negative



The third step is to count the $Recall_{pos}$ indicator and the $Precision_{pos}$ indicator.

$$\mathrm{Re}\,call_{pos} = \frac{Obstacle_{TP}}{Obstacle_{d}} \tag{4}$$

$$\mathrm{Pr}\,ecision_{pos} = \frac{Obstacle_{TP}}{Obstacle_{stereo}} \tag{5}$$

$Obstacle_{TP}$ is the obstacle frame identified as true positive in the second step, $Obstacle_{dl}$ refers to the number of target obstacles detected by the deep learning network, and $Obstacle_{stereo}$ indicates the number of target obstacles correctly detected by the detection algorithm. $Recall_{pos}$ indicates the closeness of the positive samples detected by the algorithm to the overall positive sample size of the ground truth, and $Precision_{pos}$ signifies the probability that the detection algorithm can correctly detect the target obstacle.

For the calculation of the classification algorithm index, it is calculated separately according to different categories. The calculation formula for each category is:

$$\mathrm{Re}\,call_{label} = \frac{Obstacle\_label_{TP}}{Obstacle\_label_{d}} \tag{6}$$

$$\mathrm{Pr}\,ecision_{pos} = \frac{Obstacle\_label_{TP}}{Obstacle\_label_{stereo}} \tag{7}$$

$Obstacle\_label_{TP}$ is the number of obstacle boxes for which both the actual classification algorithm and the deep learning network classification result are positive samples, and $Obstacle\_label_{dl}$ means the number of obstacle boxes for which the deep learning network classification result is a positive sample. $\mathrm{Obstacle\_label}_{dl}$ refers to the number of obstacle boxes for positive samples. Recall_{label} represents the percentage of correctly recognized objects that meet the requirements in the total number of objects in the test set, and $Precision_{pos}$ indicates the percentage of correctly recognized objects that meet the requirements of the total number of recognized objects.

ALGORITHM STABILITY EVALUATION

In the development stage, the visual inspection system is often developed according to some clear requirement rules. A complete system must not only perform well within the requirement rules but also ensure that scenarios outside the requirement rules will not collapse. The autonomous driving being evaluated usually focuses on the design and debugging of common road scenes during the development stage, while ignoring other unconventional scenes, such as images containing only walls, ceiling scenes, and indoor scene images (Lu et al., 2021), or unusually garbled images. Moreover, the stability test is to use the data of these scenarios to test that the function does not trigger errors and crashes, thereby ensuring the robustness of the algorithm (F. Wang et al., 2021).

ONLINE TEST

The autonomous driving application scenario uses the low-power platform FPGA (Xie et al., 2021) +Acorn RISC Machine (ARM; Linux environment), and the research and development (R&D) process adopts a PC processor (Windows environment). Online testing ensures consistency between the R&D and actual application scenarios through conformance testing and guarantees actual functional experience through time-consuming testing and online stability testing.

CONFORMANCE EVALUATION

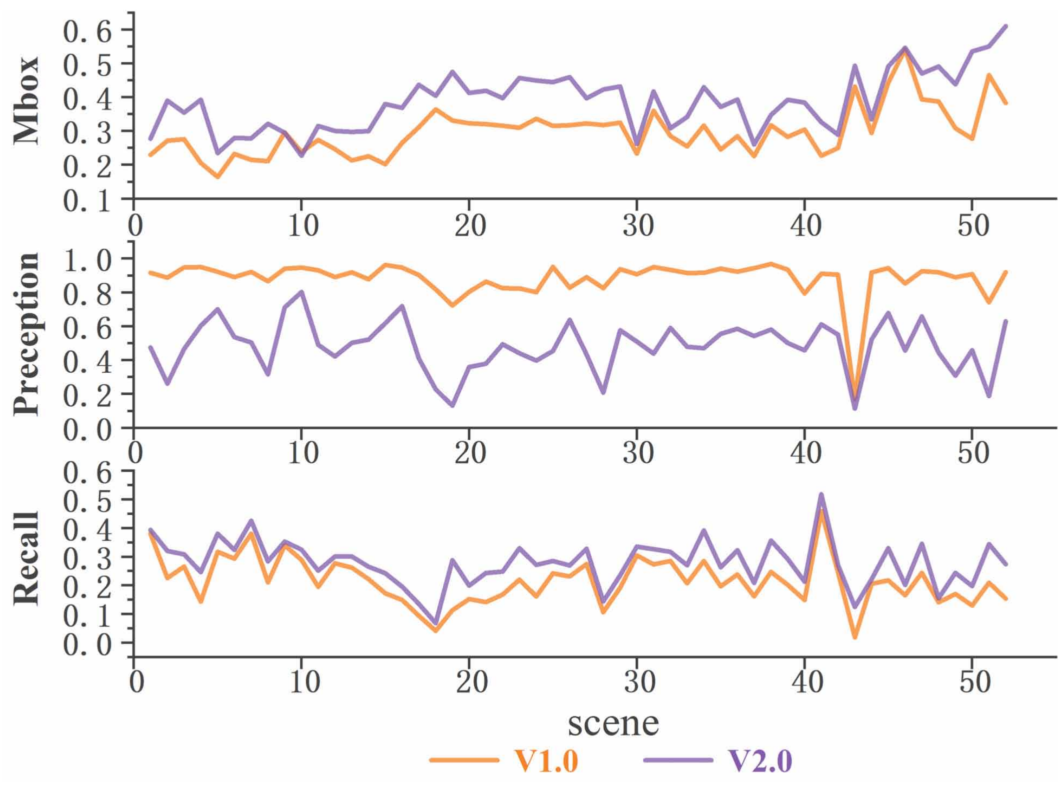
The autonomous driving platform is different in development and actual use. The identical line of codes or the same function calling may return different results, the cause of which should be identified. In this case, the test is implemented by the model-based definition (MBD) method. Firstly, the software-in-the-loop (SIL) test is performed to verify the consistency of the auto-generated code and the code-generated model. The code is compiled into a dynamic-link library (DLL) file, while the test case is used as input, and whether the output data of the model and the DLL file are consistent is observed. Among them, test cases can reuse the data employed in offline testing. If the tests are sufficient and the output is consistent, it can be roughly assumed that the code and the model behave in the same way.

The SIL test code runs on the Windows platform and cannot guarantee the effect on the target processor. Therefore, the processor-in-the-loop (PIL) tests are conducted to verify the consistency of the target processor. In terms of the PIL test, it is to compile the code into the form required by the target processor and then download it together with the test case to the target processor to run. Besides this, the same test case is used as the SIL test, and the output results are observed. The data observed and compared between the SIL test and the PIL test include obstacle information and the TTC. If the difference is within the allowable range, the test is passed.

RUNNING TIME EVALUATION

The autonomous driving platform is an FPGA+ARM architecture, which has combined advantages in performance, cost, and power consumption. The perception module is deployed on the FPGA, including the acquisition of the left image and the right image and the calculation of stereo matching. The strategic module deployed on the ARM is divided for obstacle detection, obstacle tracking, and TTC calculation. The specific deployment strategy is shown in Figure 6.

Figure 6. The deployment strategy



The perception module and the strategic module use distributed computing. After the ARM has processed the data of the current frame and the FPGA has processed the disparity map of the next frame, the ARM will immediately start the detection of the next frame. This deployment method has not yet entered parallel computing in the first frame, and there is a maximum delay time. After the function runs, the average delay time is maintained. Here, it should be mentioned that our running time test is used to evaluate the average delay time. The sampling frame rate of the system perception module we evaluate is about 12.5 fps. Thus, the state of evaluating the average delay time of fewer than 80 ms is considered acceptable.

ONLINE STABILITY EVALUATION

The autonomous driving system will eventually need to be tested on public roads. Due to the ever-changing real road scenes, the test cases cannot completely cover all driving situations, so that the evaluation results may deviate from the actual situation. Therefore, road tests are carried out for a longer period of time to ensure that the functions are functioning properly under normal driving conditions. In the test, the functional status is displayed in real time through the user interface, which is monitored by the tester. Under the specified test duration, it is ensured that the function has no memory leaks and no crash information.

RESULTS AND DISCUSSION

In this section, the evaluation schemes mentioned are analyzed with practical tests. Besides this, the big data set has a total of 400,000 test images, covering driving scenes in different weather, different light, different seasons, and different road conditions.

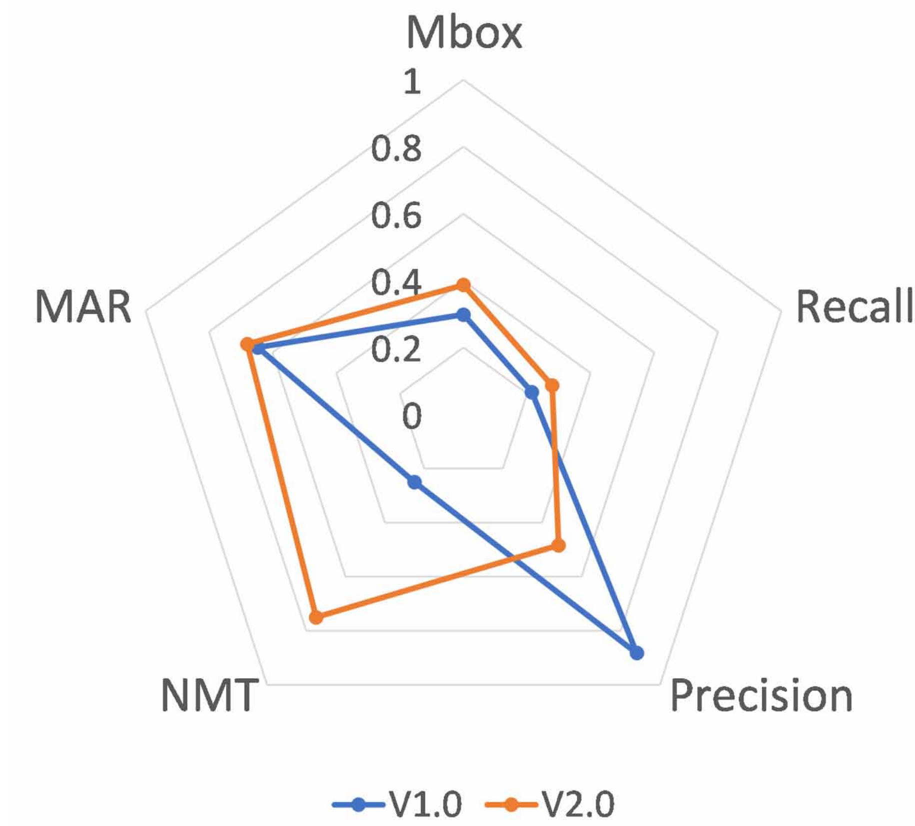
EARLY WARNING EFFECT EVALUATION EXPERIMENT (OFFLINE TEST)

Considering the final results of the autonomous driving system, firstly, the warning effect was quantitatively evaluated, as shown in Table 2. The test data of the two development versions were compared, and the correct rate of positive samples (should be warned) and that of negative samples (should not be warned) were counted. Among them, the correct rate (84.02%) of the scenarios that should be warned in Version 2.0 is higher than that of Version 1.0, but the number of scenarios that should not be warned is lower than that of Version 1.0. On the whole, Version 2.0 improves the underreporting situation more than Version 1.0 but aggravates the false-positive rate.

Table 2. Early warning assessment results

	Early Warning Scene			No Warning Scene		
Version	**Label Number**	**Detect Number**	**Correct Rate (%)**	**Label Number**	**Detect Number**	**Correct Rate (%)**
1.0	2654	1922	70.42	410	328	80.00
2.0	2654	2230	84.02	410	250	60.98

Figure 7. The evaluation effect of each scene



TRACKING EFFECT EVALUATION EXPERIMENT (OFFLINE TEST)

In the tracking module, the two metrics, namely the NMT and the MAR, are employed to evaluate the performance. To simplify the evaluation steps without losing the evaluation effect, we sampled and extracted the multi-scene data of tracking 15 frames, 25 frames, and 50 frames for comparison. Figure 7 displays the evaluation effect of each scene. The above test data are compared according to the scene, and the tracking performance of the two scenes is also counted. Table 3 counts the differences between the two versions under the tracking module.

The NMT indicator in Version 1.0 has dominant group ratios of 78.85%, 71.15%, and 51.92% for 15 frames, 25 frames, and 50 frames, respectively, which is higher than the 21.15%, 28.85%, and 48.08% ratios in Version 2.0. The proportion of the dominant group in the MAR indicator is not very different between the two versions. Beyond that, the average MAR values of Version 2.0 (0.6859%, 0.7233%, and 0.7337%) are slightly higher than those of Version 1.0 (0.6696%, 0.6988%, and 0.7401%).

Table 3. Tracking assessment results

	15-NMT		25-NMT		50-NMT	
Version	Advantages Number	Ratio (%)	Advantages Number	Ratio (%)	Advantages Number	Ratio (%)
1.0	41	78.85	37	71.15	27	51.92
2.0	11	21.15	15	28.85	25	48.08
	15-MAR		25-MAR		50-MAR	
Version	Advantages Number	Mean	Advantages Number	Mean	Advantages Number	Mean
1.0	28	0.6696	27	0.6988	26	0.7401
2.0	24	0.6859	25	0.7238	26	0.7337

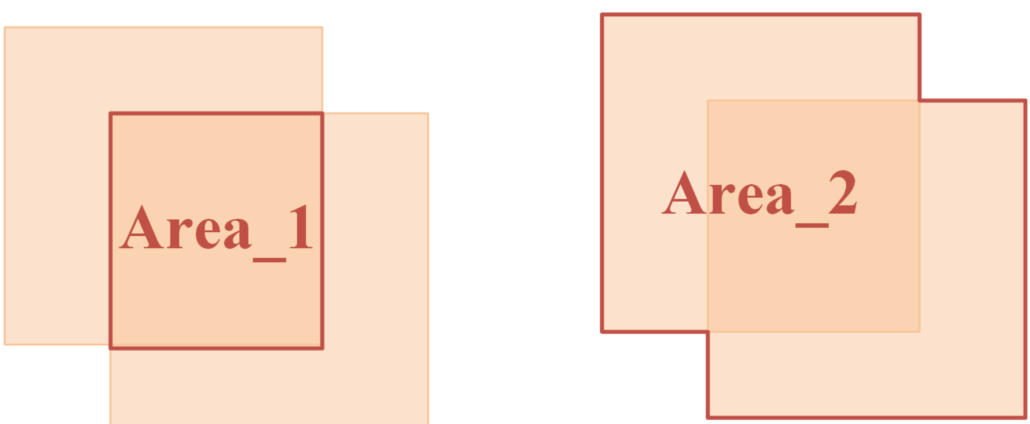
OBSTACLE DETECTION AND CLASSIFICATION MODULE EVALUATION EXPERIMENT (OFFLINE TEST)

In the detection module, the Mbox index, the Precision index, and the Recall index are evaluated. Figure 8 shows the performance in different scenarios, and Table 4 presents the statistics.

Table 4. Obstacle detection assessment results

Version	Mbox (%)	Precision (%)	Recall (%)
1.0	35.18	45.63	26.29
2.0	29.48	75.76	19.99

Figure 8. Obstacle detection assessment results of different scenarios



Compared with the Version 1.0 test, the Version 2.0 test reduces the Recall rate by 6.3%, from 26.29% to 19.99%, but the Precision rate increases by 30.13%, from 45.63% to 75.76%.

In the classification evaluation, the obstacles are classified into vehicles, pedestrians, and others. The Precision index and the Recall index are used for quantitative evaluation (see Table 5).

Table 5. Obstacle classification assessment results

Version		Vehicle	Pedestrian	Other
1.0	Precision (%)	85.40	39.92	87.14
	Recall (%)	95.65	37.46	72.96
2.0	Precision (%)	85.83	41.11	82.85
	Recall (%)	96.01	37.31	69.44

Judging from the classification statistics, the two comparative versions have little difference in the classification performance of vehicles. In the classification effect of pedestrians, the Precision of Version 2.0 (41.11%) is slightly higher than that of Version 1.0 (39.92%). For the classification effect of other categories, the Precision (87.14%) and Recall (72.96%) of Version 2.0 are both lower than the Precision (82.85%) and the Recall (69.44%) of Version 1.0, respectively. In general, the classification effect of \Version 2.0 is more obvious than that of Version 1.0 on pedestrians, but the recognition effect of other categories is worse.

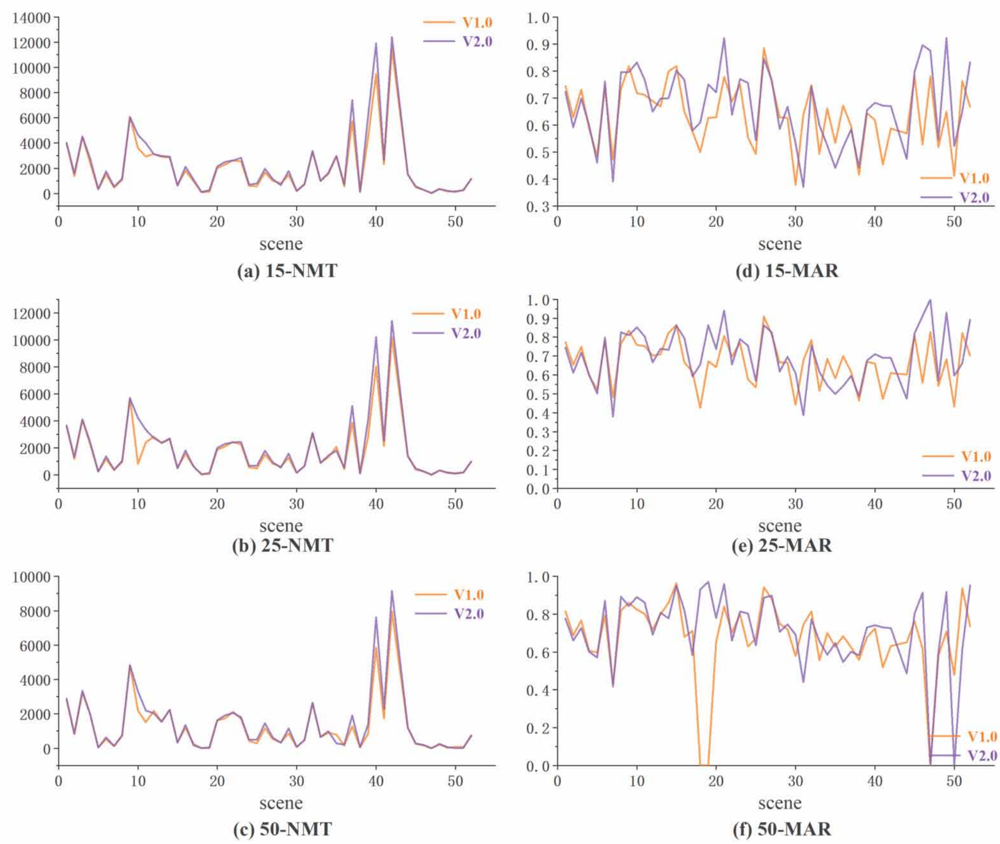
In the above experiments, the performance of the autonomous driving system was evaluated by the offline test module in all aspects. To make the comparison of the two versions more intuitive, the test results were partially integrated, as shown in Table 6 and Figure 9.

Table 6. Summary of various indicators (data)

Version	Mbox (%)	Recall (%)	Precision (%)	NMT (%)	MAR (%)
1.0	0.2980	0.2149	0.8819	0.2500	0.6469
2.0	0.3867	0.2787	0.4838	0.7500	0.6798

The mean values of the MAR and the NMT were calculated in the tracking evaluation experiment, and the mean Mbox value and the Recall and the Precision of the detection module were also measured. It can be seen from the data that Version 2.0 (0.8819%) is significantly better than Version 1.0 (0.4838%) in terms of accuracy, while in other indicators, the performance of Version 2.0 has decreased.

Figure 9. Summary of various indicators (map)



MODEL QUANTIZATION ACCURACY EXPERIMENT

Deploying deep learning models on FPGAs requires model quantization. Model quantization converts floating-point calculations into low-ratio-specific point calculations. It reduces the computational intensity, parameter size, and memory consumption, which helps improve operating efficiency, but the disadvantage is that there is an accuracy loss. In this experimental module, we compare the quantification results of the two versions of the model.

Table 7. Model quantization results

Version		Pedestrian		Vehicle	
		Before Quantification	After Quantification	Before Quantification	After Quantification
1.0	Number	19072	18088	98995	97979
	AP (%)	0.57	0.50	0.74	0.75
2.0	Number	18641	18232	98958	97776
	AP (%)	0.57	0.57	0.75	0.67

As shown in Table 7, the two versions of the model are tested on the same dataset before and after quantization. On the one hand, we count the number of detections of pedestrians and vehicles, and on the other hand, we calculate the average precision (AP) metric of the model. It can be seen that there is little difference in the indicators of the two models before quantification, and there is a difference after quantification. The model of Version 1.0 has a large loss of accuracy in the pedestrian category, and the AP indicator has dropped from 0.57% to 0.50%. On the contrary, the AP index of the model in Version 2.0 changes from 0.75% to 0.67% in the vehicle category, and there is a large loss of accuracy.

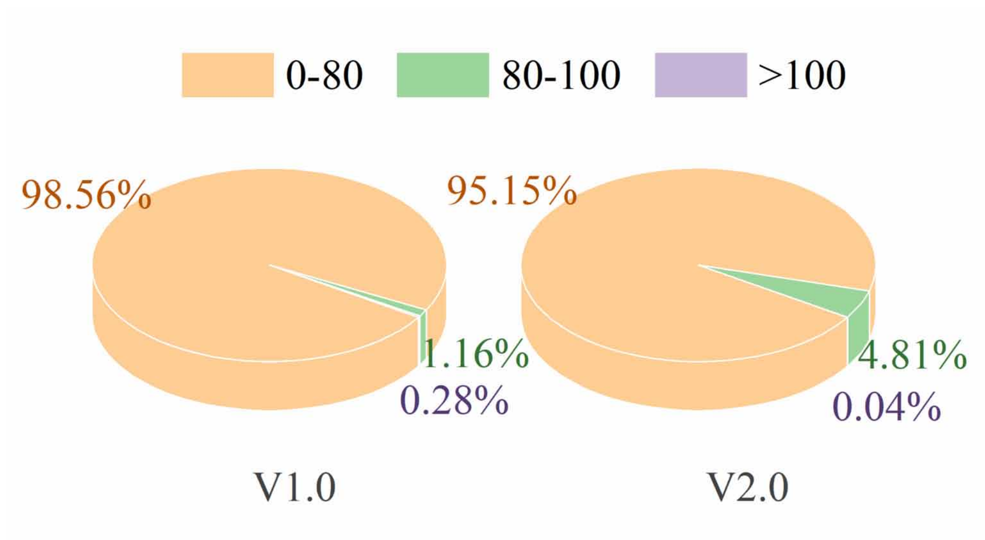
RUNNING TIME EVALUATION EXPERIMENT (ONLINE TEST)

In this experiment, the average delay time is evaluated in the actual drive test scenario, and the test time is not less than one hour. Besides this, the statistics are calculated according to the two standards of less than 80 ms and less than 100 ms, as shown in Table 8 and Figure 10. For the tests of these two different versions, it can be concluded that 95% of the results of the two meet the requirement of 12.5 fps (80 ms/frame) under normal scenarios.

Table 8. Running time assessment results

Version	Statistics Time (h)	Less Than 80 ms (%)	Less Than 100 ms (%)
1.0	1.38	98.56	99.72
2.0	1.00	95.15	99.96

Figure 10. Running time distribution (map)



CONCLUSION

In this paper, an autonomous driving evaluation system based on user big data management and AI technology is presented for the purpose of improving automotive active safety technologies. Besides this, the user big data management method, the automatic driving offline evaluation scheme, and the automatic driving online evaluation scheme are respectively introduced. Importantly, the user big data management solution is taken to enhance the data processing capability of the evaluation system, including the whole process processing capability of data adoption, data analysis, and data release. Apart from that, the combination of AI technology and the evaluation system effectively solves the problem of true value acquisition in the process of applying big data evaluation, thereby enhancing the automation of the evaluation system. The experimental results on real datasets indicate that our proposed evaluation system has achieved effective results. In the future, user big data management and AI technology will be further developed to improve the automation of the evaluation system.

ACKNOWLEDGMENT

This work was supported by the National Key R&D Program of China (2018AAA0103103); the Science and Technology Development Fund, Macao SAR (No. 0024/2018/A1); the Science and Technology Development Fund, Macao SAR (No. 0019/2021/A1).

REFERENCES

An, Y., Tang, X., Lin, X., Zheng, J., He, F., Ma, L., & Li, M. (2019). An evaluation framework to measure the performance of ADAS. *CICTP*, *2019*, 3712–3723. doi:10.1061/9780784482292.322

Fan, T. (2020). Research and realization of video target detection system based on deep learning. *International Journal of Wavelets, Multresolution, and Information Processing*, *18*(1), 1941010. doi:10.1142/S0219691319410108

George, G., Haas, M. R., & Pentland, A. (2014). Big data and management. *Academy of Management Journal*, *57*(2), 321–326. doi:10.5465/amj.2014.4002

Han, J., Heo, O., Park, M., Kee, S., & Sunwoo, M. (2016). Vehicle distance estimation using a mono-camera for FCW/AEB systems. *International Journal of Automotive Technology*, *17*(3), 483–491. doi:10.1007/s12239-016-0050-9

Han, R., Lu, X., & Xu, J. (2014, March). On big data benchmarking. *Workshop on Big Data Benchmarks, Performance Optimization, and Emerging Hardware*, 3-18. doi:10.1007/978-3-319-13021-7_1

Hu, C., Fan, W., Zeng, E., Hang, Z., Wang, F., Qi, L., & Bhuiyan, M. Z. A. (2021). Digital twin-assisted real-time traffic data prediction method for 5G-enabled internet of vehicles. *IEEE Transactions on Industrial Informatics*, *18*(4), 2811–2819. doi:10.1109/TII.2021.3083596

Kasthuri, A., Suruliandi, A., & Raja, S. P. (2019). Gabor-oriented local order feature-based deep learning for face annotation. *International Journal of Wavelets, Multresolution, and Information Processing*, *17*(5), 1950032. doi:10.1142/S0219691319500322

Kim, G., Mun, H., & Kim, B. (2019). Performance of AEB system on a slope using an extended Kalman filter. *International Journal of Software Engineering and Knowledge Engineering*, *29*(7), 955–969. doi:10.1142/S0218194019400084

Lim, B., Woo, T., & Kim, H. (2017, April). Integration of vehicle detection and distance estimation using stereo vision for real-time AEB System. *Vehicle Technology and Intelligent Transport Systems*, 211-216.

Liu, Z. (2014, July). Research of performance test technology for big data applications. *2014 IEEE International Conference on Information and Automation*, 53-58. doi:10.1109/ICInfA.2014.6932625

Long, Q., Xie, Q., Mita, S., Ishimaru, K., & Shirai, N. (2014, October). A real-time dense stereo matching method for critical environment sensing in autonomous driving. *17th International IEEE Conference on Intelligent Transportation Systems*, 853-860. doi:10.1109/ITSC.2014.6957796

Long, Q., Xie, Q., Mita, S., Niknejad, H. T., Ishimaru, K., & Guo, C. (2014, September). Real-time dense disparity estimation based on multi-path Viterbi for intelligent vehicle applications. *The British Machine Vision Conference*. doi:10.5244/C.28.127

Lu, H., Liu, Q., Liu, X., & Zhang, Y. (2021). A survey of semantic construction and application of satellite remote sensing images and data. *Journal of Organizational and End User Computing*, *33*(6), 1–20. doi:10.4018/JOEUC.20211101.oa6

Ma, C., Sun, Z., Pei, S., Liu, C., & Cui, F. (2021). A road environment prediction system for intelligent vehicle. *Wireless Communications and Mobile Computing*, *2021*, 5569295. Advance online publication. doi:10.1155/2021/5569295

Qi, L., Dou, W., Hu, C., Zhou, Y., & Yu, J. (2015). A context-aware service evaluation approach over big data for cloud applications. *IEEE Transactions on Cloud Computing*, *8*(2), 338–348. doi:10.1109/TCC.2015.2511764

Qi, L., Yang, Y., Zhou, X., Rafique, W., & Ma, J. (2021). Fast anomaly identification based on multi-aspect data streams for intelligent intrusion detection toward secure Industry 4.0. *IEEE Transactions on Industrial Informatics*. Advance online publication. doi:10.1109/TII.2021.3139363

Reddy, M. R., Srinivasa, K. G., & Reddy, B. E. (2018). Smart vehicular system based on the Internet of Things. *Journal of Organizational and End User Computing*, *30*(3), 45–62. doi:10.4018/JOEUC.2018070103

Salman, R., Myeongbae, L., Jonghyun, L., Cho, Y., & Changsun, S. (2022). A comparative study of energy big data analysis for product management in a smart factory. *Journal of Organizational and End User Computing*, *34*(2), 1–17. doi:10.4018/JOEUC.291559

So, J. J., Park, I., Wee, J., Park, S., & Yun, I. (2019). Generating traffic safety test scenarios for automated vehicles using a big data technique. *KSCE Journal of Civil Engineering*, *23*(6), 2702–2712. doi:10.1007/s12205-019-1287-4

Sun, E., Meng, K., Yang, R., Zhang, Y., & Li, M. (2021). Research on distributed data sharing system based on Internet of Things and blockchain. *Journal of Systems Science and Information*, *9*(3), 239–254. doi:10.21078/JSSI-2021-239-16

Sun, Z., Ma, C., Wang, L., Meng, R., & Pei, S. (2021). A deep learning-based binocular perception system. *Journal of Systems Engineering and Electronics*, *32*(1), 7–20. doi:10.23919/JSEE.2021.000002

Wang, F., Li, G., Wang, Y., Rafique, W., Khosravi, M., Liu, G., Liu, Y., & Qi, L. (2022). Privacy-aware traffic flow prediction based on multi-party sensor data with zero trust in smart city. *ACM Transactions on Internet Technology*, *2022*, 3511904. Advance online publication. doi:10.1145/3511904

Wang, F., Wang, L., Li, G., Wang, Y. L. C., & Qi, L. (2021). Edge-cloud-enabled matrix factorization for diversified APIs recommendation in mashup creation. *World Wide Web (Bussum)*, 1–21. doi:10.1007/s11280-020-00825-8

Wang, F., Zhu, H., Srivastava, G., Li, S., Khosravi, M. R., & Qi, L. (2021). Robust collaborative filtering recommendation with user-item-trust records. *IEEE Transactions on Computational Social Systems*, *2021*. Advance online publication. doi:10.1109/TCSS.2021.3064213

Wang, X., Li, J., An, X., & He, H. (2017). FPGA based curve lane marking detection for ADAS. *Artificial Intelligence Science and Technology*: *Proceedings of the 2016 International Conference*, 243-251. doi:10.1142/9789813206823_0032

Wu, J., & Zhang, K. (2022). Machine learning algorithms for big data applications with policy implementation. *Journal of Organizational and End User Computing*, *34*(3), 1–13. doi:10.4018/JOEUC.287570

Wu, X. (2022). Analysis of environmental governance expense prediction reform with the background of artificial intelligence. *Journal of Organizational and End User Computing*, *34*(5), 1–19. doi:10.4018/JOEUC.287874

Xie, Q., Hu, X., Ren, L., Qi, L., & Sun, Z. (2022). A binocular vision application in IoT: Realtime trustworthy road condition detection system in passable area. *IEEE Transactions on Industrial Informatics*, 1. Advance online publication. doi:10.1109/TII.2022.3145858

Xie, Q., Liu, R., Sun, Z., Pei, S., & Cui, F. (2021). A flexible free-space detection system based on stereo vision. *Neurocomputing*, *485*, 252–262. doi:10.1016/j.neucom.2021.05.115

Xie, Q., Long, Q., & Mita, S. (2017). Integration of optical flow and multi-path-Viterbi algorithm for stereo vision. *International Journal of Wavelets, Multresolution, and Information Processing*, *15*(3), 1750022. doi:10.1142/S0219691317500229

Xu, X., Fang, Z., Zhang, J., He, Q., Yu, D., Qi, L., & Dou, W. (2021). Edge content caching with deep spatiotemporal residual network for IoV in smart city. *ACM Transactions on Sensor Networks*, *17*(3), 1–33. doi:10.1145/3447032

Xu, X., Tian, H., Zhang, X., Qi, L., He, Q., & Dou, W. (2022). DisCOV: Distributed COVID-19 detection on X-ray images with edge-cloud collaboration. *IEEE Transactions on Services Computing*, *15*(3), 1206–1219. Advance online publication. doi:10.1109/TSC.2022.3142265

Zhang, X., Qi, L., Dou, W., & He, Q. (2017). MRMondrian: Scalable multidimensional anonymisation for big data privacy preservation. *IEEE Transactions on Big Data*. Advance online publication. doi:10.1109/TBDATA.2017.2787661

Zhou, X., Li, Y., & Liang, W. (2020). CNN-RNN based intelligent recommendation for online medical pre-diagnosis support. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *18*(3), 912–921. doi:10.1109/TCBB.2020.2994780 PMID:32750846

Zhou, X., Liang, W., Kevin, I., Wang, K., & Yang, L. T. (2020). Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations. *IEEE Transactions on Computational Social Systems*, *8*(1), 171–178. doi:10.1109/TCSS.2020.2987846

Zhou, X., Liang, W., Li, W., Yan, K., Shimizu, S., Kevin, I., & Wang, K. (2021). Hierarchical adversarial attacks against graph neural network based IoT network intrusion detection system. *IEEE Internet of Things Journal*. Advance online publication. doi:10.1109/JIOT.2021.3130434

Zhou, X., Yang, X., Ma, J., Kevin, I., & Wang, K. (2021). Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT. *IEEE Internet of Things Journal*, 1. Advance online publication. doi:10.1109/JIOT.2021.3077937

Zhu, X., & Yang, Y. (2021). Big data analytics for improving financial performance and sustainability. *Journal of Systems Science and Information*, *9*(2), 175–191. doi:10.21078/JSSI-2021-175-17

Pei Shanshan was born in 1993. She received her master's degree from Tianjin University of Science & Technology, China, in 2018. She is currently pursuing her Ph.D. degree in Faculty of Information Technology, Macau University of Science and Technology, Macau, China. Her research interests include computer vision, pattern recognition, machine learning and deep learning.

Ma Chao was born in 1975. Ph.D. degree in applied mathematics from Wuhan University, China, in 2005. He is currently an as sociate professor in Faculty of Information Technology, Macau University of Science and Technology, Macau, China. His research interests include Diophantine approximation, fractal geometry and applied mathematics.

Haitao Zhu received the M.S. degree in software engineering from Beihang University, Beijing, China, in 2009. From 2014, as a co-founder, he founded Beijing Smarter Eye Technology Co., Ltd. and served as R&D (Research & Development) director. Based on the international leading binocular stereo vision algorithm, the company provides vision solutions for vehicle active safety. He has applied for more than 20 international and domestic invention patents in Smarter Eye.

Luo Kun graduated with a master's degree in electronic engineering from the School of Electronic Information, Wuhan University, China. And he graduated with a Ph.D. degree in Radio Physics from the School of Electronic Information, Wuhan University, China. He is currently the general manager of the display driver product field of HiSilicon Technology Co., Limited, Shanghai, China. His research interests include computer vision, video display processing.