Utilization of Image Analysis Technology and Case Studies of Efficiency Improvement in Automotive Testing

KOMATSU Yoji

Rohit Shiv PRASAD

YOKOTA Masashi

MICHIKITA Toshiyuki

HORIBA has been developing various solutions to improve the efficiency of automotive testing, including the test automation system “STARS” and the automatic driving system “ADS EVO.” However, when visual information, such as instrument panel (IP), needs to be incorporated into driving tests, a human driver had to directly observe the display. To address this, we developed a “Machine Vision System for Vehicle Testing” that converts IP information into numerical values using image analysis technology, which interacts with STARS and ADS EVO. In this report, we describe the design concept and structure of this system, its user interface with flexible configuration options, and the fast response numerical OCR model. We also present case studies of its application in test automation.

Keywords
Machine vision, Image analysis, Instrument panel, Chassis dynamometer testing, Test automation, OCR

1. Introduction

With the rapid advancement of vehicle electrification and intelligence, there is a growing demand for both increased development speed and quality, making the efficiency of vehicle evaluation a pressing challenge[1]. HORIBA has long provided solutions to support the streamlining of automotive testing processes. For example, by managing the status and signals of test objects, various facilities, and measurement instruments collectively via the test automation system (HORIBA product name: STARS; hereafter, STARS), a mechanism has been established to automate test execution, measurement, and monitoring, thereby contributing to the efficient development of vehicle evaluation systems[2].

Among the products HORIBA offers for complete vehicle evaluation benches is an automated driving system (HORIBA product name: ADS EVO; hereafter, ADS EVO), which operates the vehicle in place of the driver[3]. ADS EVO can perform operations such as accelerator, brake, and clutch pedals, gear shifting, ignition buttons, and air conditioning switches on behalf of the tester. These functions enable precise driving in accordance with emission measurement test modes required for vehicle certification and provide a solution that reduces tester workload during long-duration tests such as durability and energy consumption testing. However, ADS EVO does not have the capability to acquire visual information from the instrument panel (hereafter, IP) or warning lights as seen by the driver, and incorporating such information into driving required tester involvement. This background revealed the need for “ADS EVO to have eyes.”

A survey of existing technologies that could serve as the “eyes” for ADS EVO found that factory automation (FA) has progressed in manufacturing sites, with examples of image analysis technology being used for visual inspection and character recognition[4]. In these cases, inspections such as defect detection and reading printed text are performed on products flowing in known shapes at designated positions. In the automotive testing domain, there are reports of image analysis technology being used for standalone IP inspection[5]. However, these prior studies are not optimal for applications such as automotive testing, where the IP, camera positions, and inspection items change multiple times a day. Additionally, such systems differ from those that assume integration with actuators operating the vehicle, like ADS EVO.

Meanwhile, the constituent technologies of image analysis have become widely available, including as open-source libraries[6]-[8], and it was considered that these technologies could deliver sufficient performance for automotive testing applications.

Leveraging these existing technologies, a proof-of-concept (PoC) and prototype were developed for a “Machine Vision System for Vehicle Testing” (hereafter, Machine Vision System), which can be operated generically for various vehicle IPs and inspection items and processes data at human-like reaction speeds, enabling integration with ADS EVO during complete vehicle testing. This report introduces this initiative: Section 2 discusses design innovations, Section 3 evaluates the standalone Machine Vision System, Section 4 presents application cases integrated with ADS EVO, Section 5 considers the effects on test efficiency, and Section 6 summarizes and discusses future developments.

2. Development of the Machine Vision System

This section explains the requirements for the Machine Vision System when integrated with ADS EVO and describes the development of both the user interface (UI) and the numerical OCR model, which are key design aspects.

2-1 System Role and Design Policy

The Machine Vision System discussed herein is intended as an element of a system integrated with ADS EVO. This subsection outlines its role and design policy. Figure 1 shows the basic configuration of the ADS EVO-integrated system and the functions of each block. ADS EVO operates the vehicle in place of the driver, the chassis dynamometer applies loads equivalent to road driving, and the Machine Vision System captures images of the vehicle’s IP and converts them into numerical data. STARS centrally manages data from each block and, based on data from the chassis dynamometer and Machine Vision System and the prescribed driving patterns, sends operation commands to ADS EVO.

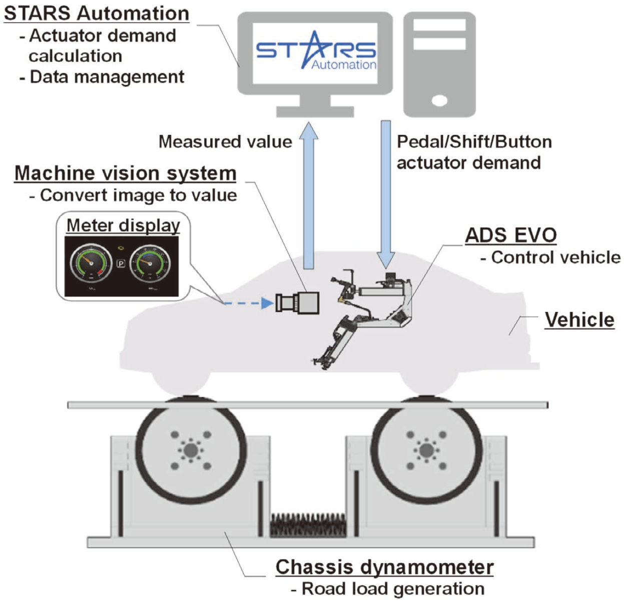


Figure 1 Diagram integrating ADS EVO and machine vision system.

Figure 2 illustrates the basic configuration of the Machine Vision System. The camera is mounted near the IP using a universal mount compatible with various vehicle shapes. The controller, equipped with a monitor for operator use, is placed in the measurement room. The controller converts camera images into numerical data and transmits them to STARS. As shown in Figure 1, STARS determines the commands for ADS EVO; thus, the Machine Vision System is designed to output numerical data read from images directly, without including detailed sequence control or error judgment functions.

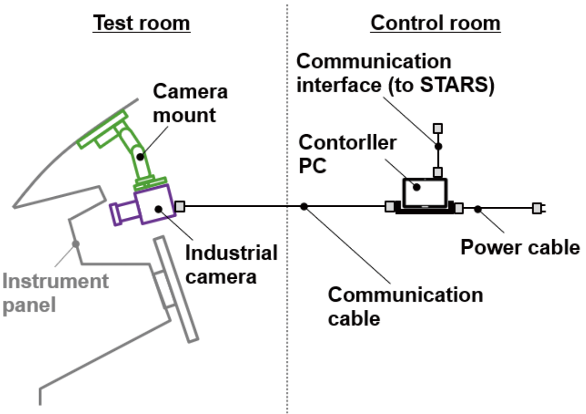


Figure 2 Machine vision system diagram.

Next, the necessary preparation steps for testing are described. Figure 3 shows the flow from preparation to measurement. First, the camera is installed at the designated position, and the controller software is launched. The camera's view is displayed on the controller screen, allowing adjustment of the viewing angle and focus. Then, a configuration file is created to define the measurement method according to the vehicle's IP. While AI-based automatic judgment could be considered, the design policy prioritized allowing testers to set parameters explicitly, addressing concerns about unexplained results.

Configuration files are created using the controller's UI. In automotive testing with a chassis dynamometer, vehicles may be swapped multiple times a day, necessitating flexible and rapid preparation for different vehicles. The UI for this purpose is explained in Section 2-2.

To ensure integration with ADS EVO, images transmitted from the camera are processed sequentially and sent immediately to STARS. Human visual reaction time is reported to be about 180 ms[9], so the Machine Vision System's processing time was targeted at 100 ms. Most constituent technologies used standard methods, but for character recognition, a custom machine learning-based model specialized for numerical recognition was developed to balance reading accuracy and responsiveness; this is outlined in Section 2-3.

2-2 User Interface Development

This subsection describes the UI for creating configuration files tailored to the vehicle, as explained in Section 2-1. First, the IPs of vehicles are described: in addition to conventional analog meter types, the spread of digital meters and added functions due to electrification and intelligence have led to diverse display content depending on manufacturer, model, and grade. However, most display methods can be broadly classified into three types: (1) pointer type (e.g., speedometer, tachometer), (2) numeric display type (e.g., odometer, clock), and (3) symbol display type (e.g., indicators, warning lights). Algorithms developed for each display type are referred to as "Circle Meter Recognition" for pointer values, "Numerical OCR (Optical Character Recognition)" for numeric values, and "Pattern Matching" for symbol presence/color, output as integer values. Testers can select the appropriate algorithm for each inspection item.

Figure 4 shows the UI for creating configuration files and the contents that can be set for each area. Area A allows selection of the video source for imaging; this is used to choose between pre-recorded videos or the currently connected camera. Area B displays the selected video source, enabling adjustment of camera angle and focus. Area C allows selection of the algorithm—Circle Meter Recognition, Numerical OCR, or Pattern Matching—for each inspection item (speed, odometer, warning lights, etc.), and specification of the analysis area in Area B. Finally, algorithm-specific settings are made: for Circle Meter Recognition, scale positions and values are set; for Pattern Matching, reference images for similarity judgment are specified in Area B; for Numerical OCR, automatic identification of character strings in the specified area is supported, so no additional settings are required. Area D shows the processing results for each inspection item, allowing adjustment of algorithm settings such as scale positions, similarity thresholds, and area settings.

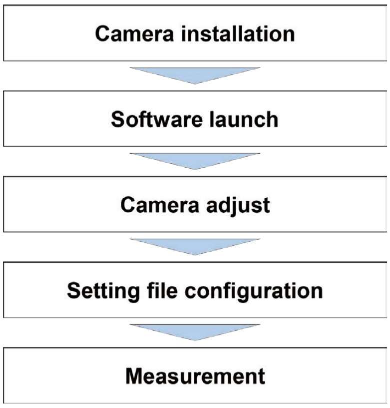


Figure 3 Operational procedure of the machine vision system.

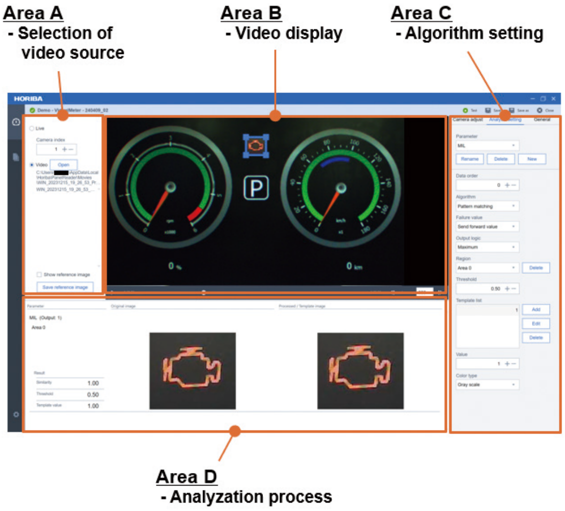


Figure 4 UI display of the setting file configuration.

Thus, while configuration of algorithm and analysis area for each inspection item is required, the objective was to enable operation in accordance with tester intent through appropriate settings[10].

2-3 Development of Numerical OCR Model

The Machine Vision System was designed for integration with ADS EVO, with the numerical OCR aiming for human-equivalent processing speed. Generally, OCR libraries are known to be computationally intensive due to image and mathematical processing[11]. Experimental implementation using the open-source OCR Tesseract[6] showed that system-wide processing time exceeded 100 ms when the number of reading targets increased. Therefore, a fast numerical OCR model was developed using a machine learning approach specialized for numerical recognition.

PyTorch™*1[12], a deep learning framework, was used for model training. Figure 5 shows the training and evaluation workflow. Considering the variety of fonts used in actual vehicle IPs, numeric images were generated using 25 fonts, with added disturbances in size, position, and inclination. Random image blending was also applied to simulate practical conditions such as IP scratches and reflections. The model was trained and validated using these images, allowing tuning for user-specific applications rather than relying on opaque library processes.

Figure 6 shows examples of training data used. Training was performed with 1.2 million images and validation with 10,000 images, resulting in a reading accuracy of 94.4%. Note that the validation included images with partially missing digits, as shown in Figure 6, so the accuracy was assessed under practical conditions, with results discussed in Section 3-2. Finally, the processing speed of the numerical OCR model is described.

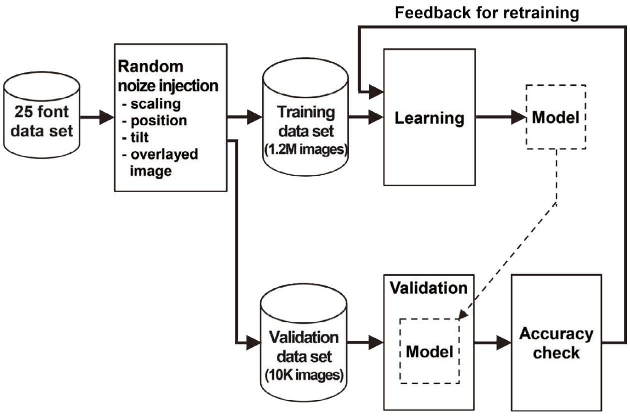
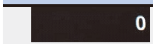
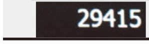


Figure 5 Process flow of model training and validation.

Table 1 shows the reading processing speed for the model described. Experiments used a laptop PC with a 13th Gen Intel® Core™*2 i7-1365U 1.80 GHz and 32GB RAM. Reading speeds for single-digit and five-digit numbers were measured, with processing times of approximately 30 μs and only a 5 μs difference between the two, confirming that digit length had minimal impact. Thus, even with increased reading targets, the system-wide processing speed goal of 100 ms can be met, and this numerical OCR model was adopted for the system.

Figure 6 Training and validation data set example.

Table 1 Processing speed performance of the digit-specific OCR model.

Displayed image	Processed time
	27 μs
	32 μs

*1 Registered trademark or trademark of The Linux Foundation

*2 Registered trademark or trademark of Intel Corporation

3. Evaluation of the Standalone Machine Vision System

This section presents the evaluation results for the reading accuracy of the standalone Machine Vision System. Section 3-1 reports the accuracy of each of the three reading algorithms; Section 3-2 presents additional evaluation of the numerical OCR model using various fonts from actual vehicle IPs.

3-1 Reading Accuracy Evaluation

Due to the diversity of IP displays, it is not easy to set a uniform reading accuracy target for the Machine Vision System. Therefore, a simulated IP device capable of outputting specified values was used for evaluation, with a reading success rate target of 99%. Accuracy was assessed by first confirming whether each algorithm's reading values continuously tracked the command values, then considering system processing time and responsiveness, and finally defining reading success for each algorithm and verifying accuracy.

Figure 7 shows a schematic of the evaluation device. A simulated IP displaying speed, odometer, and malfunction indication lamp (MIL) was prepared, and readings were compared to controller command values. Circle Meter Recognition, Numerical OCR, and Pattern Matching were applied to speed, odometer, and MIL, respectively. WLTC speeds were used for speed command values; odometer readings were set to increase from 0 to 999,999 during the WLTC1 cycle to test up to six digits. MIL was tested with frequent switching (off, yellow, red every 5 seconds), with reading values of 0, 1, and 2, respectively, repeating the cycle three times for reproducibility.

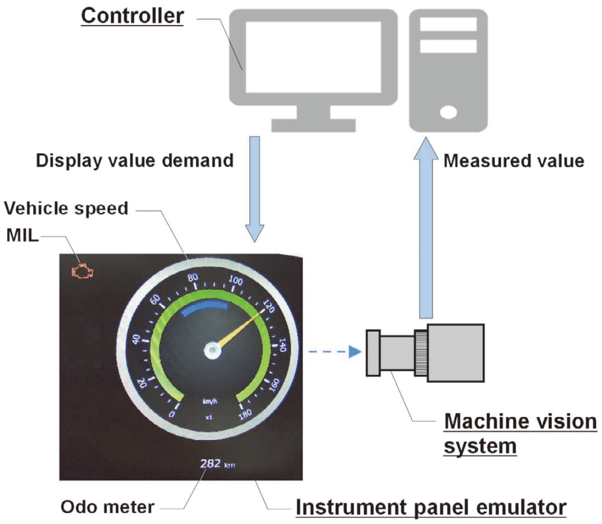


Figure 7 Performance test equipment diagram.

Figure 8 shows the command and measured values for speed using Circle Meter Recognition and their correlation. Across low to high speeds and three measurements, command and measured values tracked similarly; the regression slope was 0.983 and R^2 was 0.998, indicating good tracking. Minor discrepancies were attributed to needle shape and scale position settings.

Figure 9 shows command and measured values for the odometer using Numerical OCR and their correlation. Across one to six digits, values tracked well; regression slope was 1.00 and R^2 was 0.999, though a few outliers were observed. Table 2 lists image changes that contributed to these outliers, such as intermediate images captured during number transitions, which are difficult for humans to judge. Future work includes implementing outlier removal and interpolation using multiple frames.

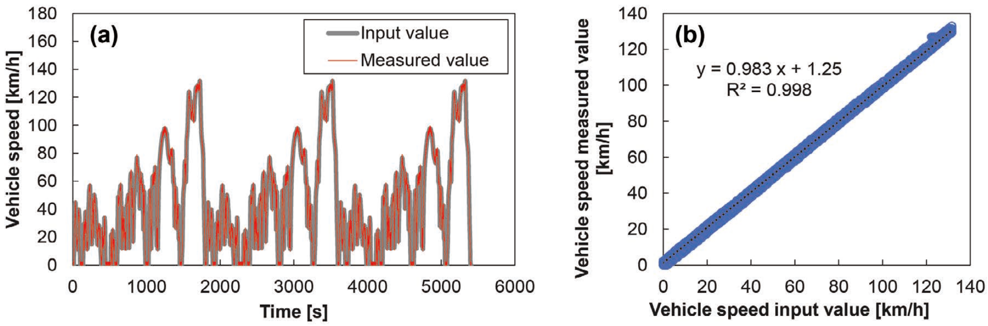


Figure 8 Displayed value vs. recognized value of (a) vehicle speed in time-series comparison and (b) correlation plot.

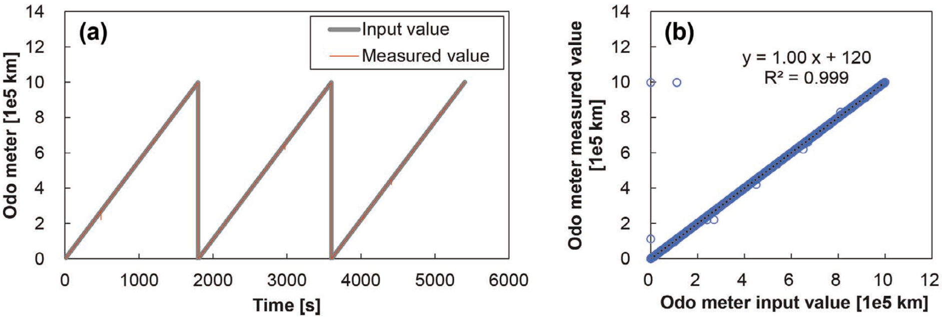


Figure 9 Displayed value vs. recognized value of (a) odo-meter in time-series comparison and (b) correlation plot.

Figure 10 shows command and measured values for MIL using Pattern Matching, with expanded views of the start, middle, and end of three test cycles. All states (off, yellow, red) were accurately read.

Next, processing and response times during measurement were examined. Figure 11 shows time-series changes for speed, odometer, and MIL. Speed was updated at 100 ms intervals, meeting the target. Odometer and MIL lagged by 300–400 ms and 300–700 ms, respectively, relative to command values, but these include controller and simulated IP display delays. Standalone system response time will be separately verified.

Finally, definitions of reading success were set for each algorithm. For Circle Meter Recognition, success was defined as a difference of ≤2.5 km/h (half a scale) between command and measured values. For Numerical OCR and Pattern Matching, success was defined as matching command and measured values within the observed lag time. Based on these definitions, reading accuracy was 99.8% for Circle Meter Recognition, 97.7% for Numerical OCR, and 99.1% for Pattern Matching. The lower accuracy for Numerical OCR was attributed in part to difficult images (Table 2), and will be further investigated.

Table 2 Fail case example in value recognition.

Displayed image	Demand value	Measured value	Result
	829	829	OK
	830	880	NG
	830	830	OK

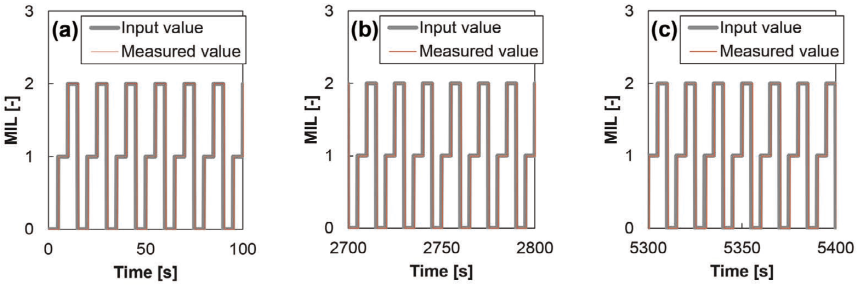


Figure 10 Displayed value vs. recognized value of (a) MIL in 1st cycle, (b) 2nd cycle and (c) 3rd cycle

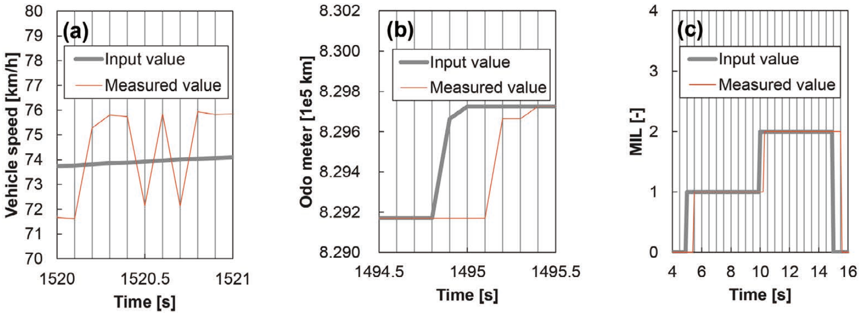


Figure 11 Displayed value vs. recognized value of (a) vehicle speed, (b) odo-meter and (c) MIL in an enlarged time window

3-2 Additional Evaluation of Numerical OCR Mode

Among the three algorithms, only Numerical OCR uses machine learning, raising concerns about unexplained outputs for certain targets. This section reports robustness tests using various numeric fonts from actual vehicle IPs. From internal image data, 50 numeric samples from 13 vehicle models were selected, excluding images with severe tilt, distortion, or focus errors. Table 3 shows sample images, their numeric values, measured results, and judgment. The model targeted only Arabic numerals; symbols like dots or colons were not included. When such symbols appeared, only the Arabic numerals were output, and all 50 samples produced the intended results. Conversion to physical values considering these symbols is assumed to be handled by higher-level systems.

4. Applications Integrated with Automated Driving Systems

The Machine Vision System described in this report was developed with the aim of integration into test sequences involving ADS EVO. This section presents examples of ADS EVO-integrated applications for which proof-of-concept (PoC) was conducted in HORIBA's research and development.

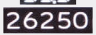
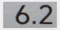
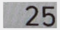
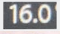
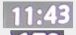
4-1 Gear Shift Indicator-Integrated Application

ADS EVO is capable of handling not only automatic transmission vehicles but also manual transmission gear shift operations. Some vehicles are equipped with a gear shift support function known as the Gear Shift Indicator (GSI). GSI notifies the driver of the recommended gear via instrument panel display according to driving conditions. UN Regulation No. 154 stipulates that, for electrified vehicles equipped with GSI, gear shifting must follow the recommended gear display[13], making GSI-compliant driving essential for accurate measurement of fuel economy, energy consumption, and emissions. When using ADS EVO alone, instrument panel displays could not be read, so GSI-compliant driving required manual operation by the tester. This section introduces a case where automatic gear shifting was achieved by combining ADS EVO with the Machine Vision System to respond to GSI displays.

Figure 12 shows the operation method for the test vehicle's GSI. Moving the shift knob left from D allows upshifting/downshifting by moving it forward or backward. In this state, ADS EVO performs gear shifts according to the instrument panel display.

Figure 13 shows the instrument panel display of the test vehicle. During GSI shifting, the current gear is displayed

Table 3 Evaluation results of the font recognition test from various vehicle instrument panels.

Displayed image	Displayed value	Measured value	Result
	28384	28384	OK
	35.3	353	OK
	26250	26250	OK
	289	289	OK
	76	76	OK
	23	23	OK
	9:32	932	OK
	23816	23816	OK
	678	678	OK
	16:14	1614	OK
	24.7	247	OK
	21458	21458	OK
	19	19	OK
	400	400	OK
	14.0	140	OK
	040599	40599	OK
	16:10	1610	OK
	16	16	OK
	9348	9348	OK
	432	432	OK
	6.2	62	OK
	25	25	OK
	99	99	OK
	37754	37754	OK
	53365	53365	OK
	164	164	OK
	65	65	OK
	15:48	1548	OK
	8	8	OK
	14497	14497	OK
	467.1	4671	OK
	76	76	OK
	16:07	1607	OK
	16.0	160	OK
	049348	49348	OK
	267.2	2672	OK
	68545	68545	OK
	710	710	OK
	373.7	3737	OK
	7:59	759	OK
	19.0	190	OK
	37708	37708	OK
	16:30	1630	OK
	24	24	OK
	29.7	297	OK
	34	34	OK
	27008	27708	OK
	11:43	1143	OK
	170	170	OK
	23	23	OK

(Figure 13a). As speed increases, a gear shift request symbol and recommended gear are displayed (Figure 13b).

The system shown in Figure 1 of Section 2-1 was used for testing. The Machine Vision System reads the current gear and gear shift request using pattern matching and the recommended gear using numerical OCR, transmitting results to STARS. Upon recognizing a gear shift request, STARS sends gear shift commands to ADS EVO to reach the recommended gear.

Using this method, acceleration to the highest gear (7th) and subsequent deceleration were performed. During acceleration, gear shift requests were displayed; during deceleration, the test vehicle automatically performed downshifting without displaying gear shift requests. For the shift from 1st to 2nd gear, gear shift requests were not displayed, so manual shifting was performed to avoid engine over-revving. Figure 14 shows vehicle speed, current gear, recommended gear, and the gear shift command signals issued by STARS for the upshift section from 2nd to 7th gear. When a “M2▶M4” gear shift request was displayed during 2nd gear driving, two consecutive shifts were performed to reach the recommended 4th gear. Subsequent recommended gears were displayed for each upshift, with correct shift operations executed accordingly.

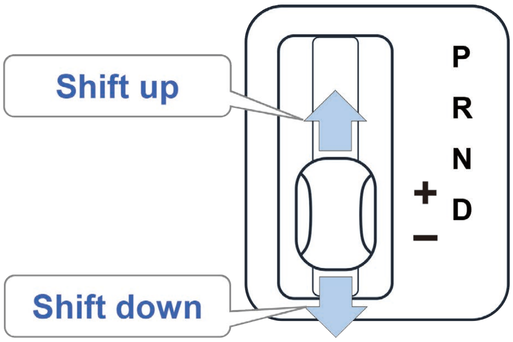


Figure 12 Operational method of GSI shift in the test vehicle.

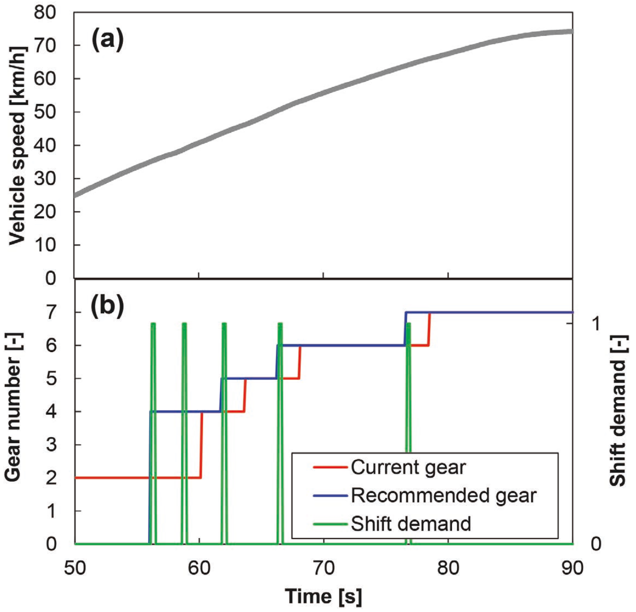


Figure 14 (a) Vehicle speed profile and (b) transition of current gear, recommended gear and shift demand.

The time from recommended gear recognition by the Machine Vision System to gear shift command issuance by STARS was within 100 ms, confirming satisfactory system responsiveness. Future work will use this system for regulatory cycle driving in accordance with GSI displays.

4-2 Adaptive Cruise Control Evaluation Application

Next, an application example of the Machine Vision System for evaluation of Advanced Driver Assistance Systems (ADAS) is presented. ADAS evaluations are typically conducted in real-world environments such as test courses, but laboratory-based evaluation methods have recently been studied[14]. This section reports results of laboratory evaluation of Adaptive Cruise Control (ACC), a key ADAS function, using ADS EVO and the Machine Vision System.

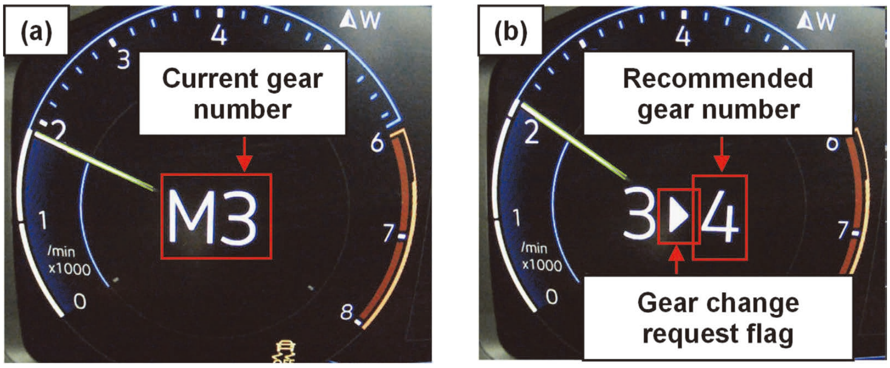


Figure 13 Meter display of (a) current gear and (b) gear number recommendation.

ACC not only maintains constant vehicle speed but also uses sensors and cameras to recognize and follow a preceding vehicle. With full-speed-range following, no operation is needed from test initiation to stop. However, before starting ACC driving, settings such as ACC enable/disable state, maximum allowable speed (set speed), and following distance (set distance) must be configured. When stopping behind or restarting after the preceding vehicle, specific operations on the brake pedal or steering buttons are required, based on instrument panel displays. Without the Machine Vision System, these operations required tester intervention. Here, a case is introduced where pre-driving ACC settings and ACC-based driving were automated using ADS EVO and the Machine Vision System.

Figure 15 shows the instrument panel display of the test vehicle. Pressing the ACC enable button illuminates the ACC state; set speed and set distance are adjusted using buttons, with the value and arrow position changing accordingly. The test vehicle allows set distance adjustment in five steps; “Short,” “Middle,” and “Long” refer to the shortest, middle, and longest settings, respectively.

Figure 16 shows the system configuration used for testing. The Machine Vision System reads ACC state and set distance via pattern matching and set speed via numerical OCR, transmitting results to STARS. STARS continuously recognizes current ACC state, set speed, and set distance, sending button operation commands to ADS EVO to match test conditions set by the tester. The test vehicle uses radar sensors for external recognition, and a radar target simulator was used to simulate preceding vehicle reflections. STARS calculates the relative relationship between the test vehicle and preceding vehicle from the specified profile and test vehicle speed, sending this to the radar target simulator. These functions allow combination of Machine Vision System and ADS EVO operation sections with ACC-based driving, enabling the test vehicle to follow the preceding vehicle according to tester-specified conditions[15].

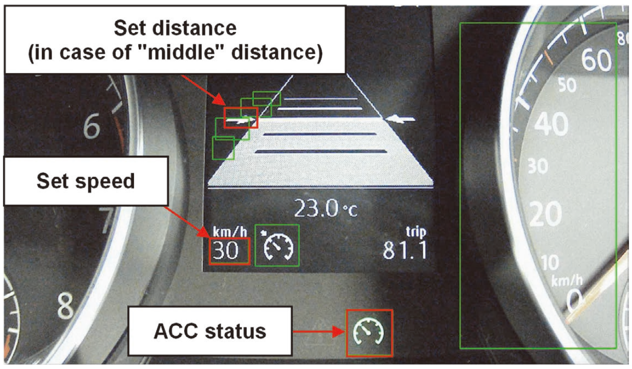


Figure 15 Meter display of ACC status, set speed and set distance.

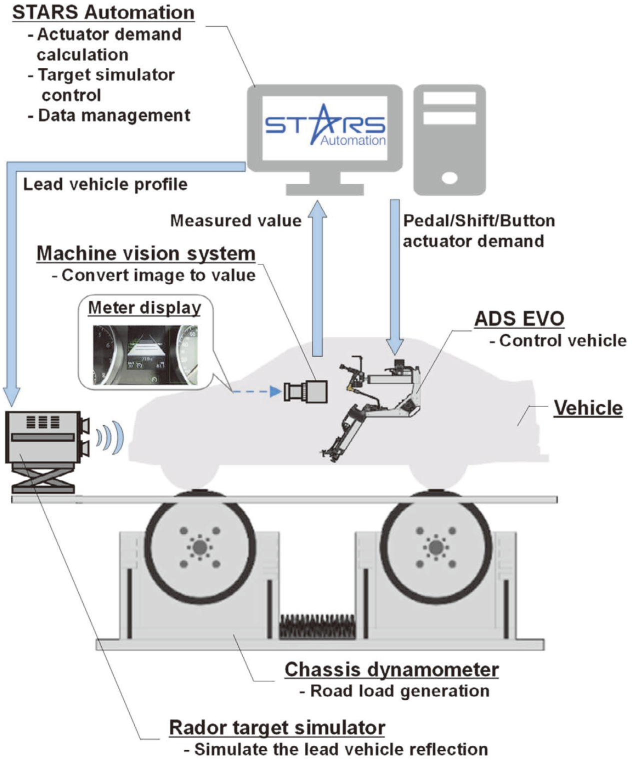


Figure 16 Test system diagram.

Using this system, speed and following distance were measured for different set distances, with the preceding vehicle profile set to WLTC. Figure 17 shows time-series changes in speed and following distance for the urban mode (Low phase) with frequent stop-and-go and

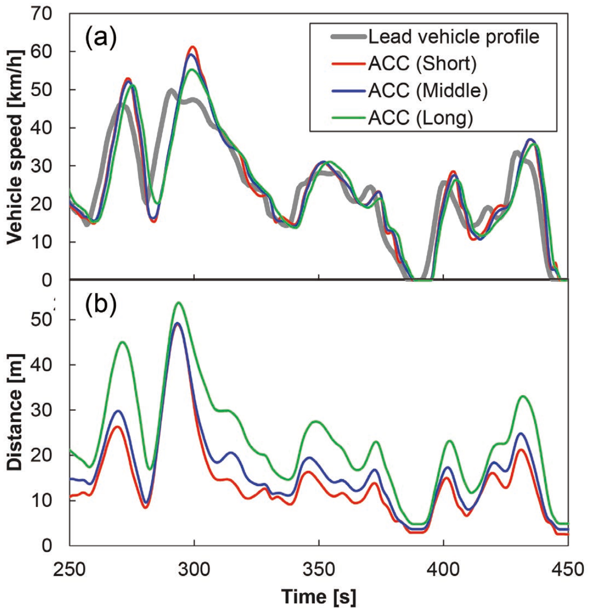


Figure 17 (a) Vehicle speed and (b) inter-vehicle distance at different set distances.

high ADS EVO operation frequency. In all set distances, the test vehicle followed the preceding vehicle's speed, and following distance decreased in order "Long," "Middle," "Short" during both driving and stopping.

The consistency between set and actual following distances confirms that set distances adjusted via the Machine Vision System were correctly reflected in the vehicle, validating the system.

5. Consideration of Efficiency

The system combining ADS EVO and the Machine Vision System described in Section 4 can potentially reduce tester workload and enable labor savings. However, preparation time for these devices must be considered when assessing the overall effect. This section provides a simple estimate of time savings when using ADS EVO and the Machine Vision System compared to manual operation. The estimate assumes a 30-minute test (WLTC equivalent), a 30-minute interval for data confirmation and adjustments, and seven tests conducted during an 8-hour workday.

Table 4 shows the time required for system setup and cleanup, totaling 63 minutes. The system enables 210 minutes of test driving without manual operation, yielding a net time saving of 147 minutes. This can contribute to reduced tester workload and improved productivity by reallocating time to other tasks.

6. Conclusion

This report presents a Machine Vision System that converts vehicle instrument panel displays into numerical data via image analysis technology and integrates with proprietary test systems, covering the following points:

- Developed three reading algorithms and a UI for configuring them to accommodate different instrument panel displays for each vehicle.
- Built a machine learning model specialized for numerical recognition to enable fast processing even with increased reading targets.
- Achieved high reading accuracy close to targets for all three algorithms and confirmed processing within the targeted time.
- Demonstrated that instrument panel displays related to GSI and ACC functions can trigger integrated testing with ADS EVO.
- Confirmed, through work-time estimation including preparation and cleanup, that system operation contributes to overall test efficiency.

These results indicate that the Machine Vision System is not merely a recording device, but an effective element for improving the efficiency of automotive testing. Future plans include productization and market introduction, with ongoing improvement of functions and performance based on market feedback. Specifically, efforts will focus on enhancing robustness for images with reduced visibility due to blur or distortion, developing outlier removal and correction functions, expanding algorithms, and strengthening configuration support features.

* Editorial note: This content is based on HORIBA's investigation at the year of publication unless otherwise stated.

Table 4 Classification of pre-test and post-test procedures for automated driving tests.

Main category	Subcategory	Time
Pre-test procedure	ADS EVO installation	8 min
	Camera installation	5 min
	Test run for software configuration	5 min
	Software configuration	5 min
	Vehicle characteristic learning	25 min
Post-test procedure	Disassemble / Removal of test equipment	15 min
Total time of pre-test and Post-test procedure	—	63 min

References

[1] 経済産業省・国土交通省, モビリティDX戦略, 2024.（in Japanese）

[2] 駒田峰之ほか, HORIBAの最新自動車実験棟 “E-LAB”, HORIBA Readout, Vol. 46, pp. 47–50, 2016.（in Japanese）

[3] 堀場製作所, ADS EVO製品紹介ページ, https://www.horiba.com/int/automotive/products/detail/action/show/Product/ads-evo-1994/, accessed 2026-03-30.（in Japanese）

[4] 鷲見和彦ほか, マシンビジョン技術の実利用, 電気学会論文誌C（電子・情報・システム部門誌）, Vol. 124, No. 3, pp. 598–605, 2004.（in Japanese）

[5] S.Kalkar, et al., Machine Learning Based Instrument Cluster Inspection Using Camera, *SAE Technical Paper* 2022-28-0076, 2022

[6] Tesseract OCR, https://tesseract-ocr.github.io/, accessed 2026-03-30.

[7] Hough Line Transform, https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html, accessed 2026-03-30.

[8] Template Matching, https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html , accessed 2026-03-30.

[9] A.T.Welford, Introduction: An Historical Background Sketch, Reaction Times, pp.1-23, ACADEMIC PRESS, 1980

[10] 小松陽二ほか,「供試体表示認識装置，供試体試験システム，供試体表示設定装置，供試体表示認識方法，及び，供試体表示認識プログラム」, WO2024/204704, Oct. 3, 2024.（in Japanese）

[11] B.Sunghwan, et al., Performance Characterization and Parallelization of Tesseract Optical Character Recognition on Multicore Architectures, Proceedings of IEEE High Performance Extreme Computing Conference（HPEC）, 2016

[12] PyTorch, https://pytorch.org/, accessed 2026-03-30.

[13] United Nations Economic Commission for Europe（UNECE）, UN Regulation No. 154: Uniform provisions concerning the approval of light duty passenger and commercial vehicles with regards to criteria emissions, emissions of carbon dioxide and fuel consumption and/or the measurement of electric energy consumption and electric range（WLTP）, Supplement 1 to the 02 series of amendments, Annex B8, Paragraph 1.5, 2024

[14] 香月影幸ほか, 自動運転に向けた実車両を使った検証（VIL）を実現する際の課題と対策, 自動車技術会学術講演会予稿集（春）, 20235045, 2023.（in Japanese）

[15] 小松陽二ほか,「供試体試験システム，供試体試験方法及び供試体試験プログラム」, WO2024/070865, Apr. 4, 2024.（in Japanese）

KOMATSU Yoji

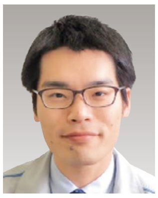
E-Mobility Dept.
Energy & Environment System Center
Energy & Environment Division
HORIBA, Ltd.

Rohit Shiv PRASAD

E-Mobility Dept.
Energy System Center
Energy & Environment Division
HORIBA, Ltd.

YOKOTA Masashi

E-Mobility Dept.
Energy System Center
Energy & Environment Division
HORIBA, Ltd.



MICHIKITA Toshiyuki

E-Mobility Dept.
Energy System Center
Energy & Environment Division
HORIBA, Ltd.