

Leveraging vision language models for digit recognition on digital measuring devices: a comparative analysis

Adindra Vickar Ega^{1,2}, Dadang Gunawan¹, Abdul Halim¹, Rudi Anggoro Samodro²

¹Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia

²Research Center for Equipment Manufacture Technology, National Research and Innovation Agency, South Tangerang, Indonesia

Article Info

Article history:

Received Nov 1, 2025

Revised Aug 11, 2026

Accepted Aug 20, 2026

Keywords:

Digit recognition

Digital measuring device

Google Gemini

Optical character recognition

Vision language models

ABSTRACT

Large language models (LLMs) with multimodal input capability have been demonstrated in various applications. Vision language models (VLMs), which were developed from LLMs, can process and learn both visual and textual data simultaneously, enabling them to generate descriptive text from images. This study explores the use of several VLM models for the seven-segment digit recognition of a digital measuring device compared with other baseline optical character recognition (OCR) methods, such as Tesseract OCR, EasyOCR, PaddleOCR, KerasOCR, and YOLO11, with or without the integration of you only look once 11 (YOLO11) for LCD screen cropping. The results showed that the Gemini model outperformed conventional OCR methods and other VLM models with an average per-digit-recognition (PDR) accuracy of 98.10% ($\sigma=0.36\%$) and full sequence reading (FSR) accuracy of 93.85% ($\sigma=1.29\%$). The performance evaluation across the distance variation also demonstrates high accuracy for the VLM result with stable results. Simultaneously, YOLO11 exhibits degradation at longer distances due to difficulty in detecting small decimal digits, whereas the traditional OCR method yields inconsistent results. This study showed promising results for the further development, improvement, and deployment of VLM models in the automatic seven-segment digit recognition task on the edge device.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Dadang Gunawan

Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia

Kampus UI Depok, Depok, West Java 16424, Indonesia

Email: guna@eng.ui.ac.id

1. INTRODUCTION

The recent development of large language models (LLMs), such as OpenAI's ChatGPT [1], [2], Meta's LLaMa [3], [4], Microsoft's Copilot [5], [6], and Google's Gemini, has made significant progress from the unimodal input method with text or speech to multimodal input with various inputs of text, images, audio, video, and PDFs [7], [8].

Launched on December 6, 2023, Gemini, the latest multimodal AI tool powered by Google DeepMind, is capable of generating output content based on the input prompt [9], [10]. Similar to ChatGPT-4 or the recent ChatGPT-5, Google Gemini can generate text and images, solve complicated mathematical equations, or even generate various programming language codes for developers and programmers, even though beginner programmers struggle to understand LLM-generated code [11], [12]. Unlike ChatGPT, Gemini can acquire and process information from the Google search database, ensuring responses to the latest ongoing knowledge.

Despite the contribution of Gemini AI in a wide application area, mostly in the healthcare industry [13]-[15] and education [16]-[18], the LLM-based approach was also used as an advance in handwritten

character recognition. Given a handwritten page image, the study found that the LLM was more accurate and reacted faster than the optical character recognition (OCR) [19], [20]. Vision language models (VLMs) were developed from LLMs to process and learn both visual and textual data simultaneously. For example, VLMs can generate descriptive text from images [21], [22]. A recent study reported that Google Gemini, along with other VLMs, such as OpenAI GPT-4o, Google PaliGemma, and Meta Llama, were used to predict license plate numbers in vehicle plate recognition applications under complex situations such as low, unclear vehicle plates, low-quality cameras, and close characters. The results showed that the proposed VLMs outperformed the other OCR baseline methods in terms of plate-level recognition accuracy, with a character recognition accuracy of 97.66% from 226 plate images out of 258 images [23].

To the best of the author's knowledge, no prior research has explored the use of VLMs for digit recognition tasks, especially for the recognition of seven-segment digit characters from a digit measuring device liquid crystal display (LCD) display. Thus, this is one of the first exploratory studies to apply VLMs to seven-segment digit recognition. The use of the VLM model to recognize seven-segment digits from images of digital measuring devices was evaluated. VLM models, such as Gemini 2.5 flash, are LLMs at their core, but they are also enhanced as VLMs to handle multimodal inputs, such as text, images, audio, and video. We also compared the proposed method with other state-of-the-art OCR methods, such as Tesseract-OCR [24], EasyOCR [25], KerasOCR [26], PaddleOCR [27], and YOLO11 [28], to benchmark the performance in the seven-segment digit recognition task.

The remainder of this paper is organized as follows: in the methods section, we describe the dataset used for the seven-segment digit recognition tasks and the evaluation metrics used for the comparison analysis. The results and discussion section presents the experimental results and findings. Finally, the conclusion section summarizes this study, followed by suggestions for future work to overcome the study's limitations.

2. METHOD

2.1. Dataset

Two dataset versions were used in this study, the DMD dataset LCD screen detection dataset and the DMD BEEI project dataset. The first dataset was used for YOLO11 training to detect and crop the LCD screen of the DMD as input images for further digit recognition. The first dataset consists of 8514 DMD images with a single class bounding box annotation, which is LCD-screen. The distribution of the first dataset consists of 7503 train dataset images (88%), 662 valid dataset images (8%), and 349 test dataset images (4%). The training dataset was augmented with various image preprocessing techniques, including rotation (between -10° and $+10^\circ$), blur (up to 1.5 pixel), and noise (up to 0.5% of pixels). Figure 1 shows samples of the first dataset.



Figure 1. Sample of the DMD dataset LCD screen detection

The DMD BEEI project dataset, as the second dataset, was used for training various digits recognition to detect and recognize digits of the DMM directly from the given DMM images with multi-class bounding box annotation, which are decimal point digits (".") and the digit number itself ("0"–"9"). It has a total of 692 images of various digital measuring instruments, with 484 images (70%) for the training dataset, 104 images (15%) for the validation dataset, and 104 images (15%) for the testing dataset. All images in the training, valid, and test folders were pre-augmented with rotation (by -10° , $+10^\circ$, -15° , and $+15^\circ$), Gaussian blur (with kernel

size 3×3 and 5×5 variation), and gaussian noise, which adds random noise with a standard deviation of 15. Figure 2 shows samples of the second dataset. The second dataset's test folder was also used to benchmark the digit recognition performance of several methods used in this study. Both datasets were annotated using the Robflow platform.



Figure 2. DMD BEEI project dataset samples

Several different types of digital measuring instruments, such as digital pressure gauge additel 681, digital multimeter (DMM) DT-830B, DMM fluke 116, DMM fluke 289, fluke 355 clamp meter, and fluke 754 documenting process calibrator, were used in both datasets to increase the diversity of both datasets and enhance the generality of the recognition performance for the various trained models or algorithms. Figure 3 illustrates various digit representations of several types of digital measuring instruments. Structure of the digit was based not only on seven-segment digit representation but also on bitmap or pixel-based LCD digit representation.



Figure 3. Various digit structure representations from several types of digital instruments in both datasets

2.2. Experimental environment and procedure steps

The experiment was conducted in a laptop with the following specifications: AMD Ryzen 7 7445HS 3.2GHz, 16 GB DDR5 RAM, 4 GB NVIDIA GeForce RTX 3050 Laptop GPU, Windows 11 home single language 64-bit, CUDA version 12.8, Python version 3.12.10, Python version 3.10.11, and visual studio code integrated development environment (IDE) version 1.113.0. Figure 4 shows the flowchart of the experiment step conducted in this study. In the first step, an OCR algorithm was chosen from several methods used for benchmarking. Afterward, in the experiment stage, whether or not the experiment needs to crop the LCD screen region of interest (ROI) to evaluate the effect of YOLO cropping on the digit recognition results needs to be

decided. To obtain the cropped LCD screen of the digital measuring instruments, the trained YOLO11 model with the first dataset (DMD dataset LCD screen detection) was used to detect the LCD screen and crop the results for further experiments. If the LCD screen does not need to be cropped, then the loaded test image will be preprocessed first or will be directly read the reading value by the chosen algorithm and save the result in an Excel file, except for the VLM model, which needs an application programming interface (API) key from the API key provider along with the provided images and input text prompt to perform the reading. Figure 5 illustrates the digit recognition pipeline with YOLO11 for LCD screen cropping and VLM inference.

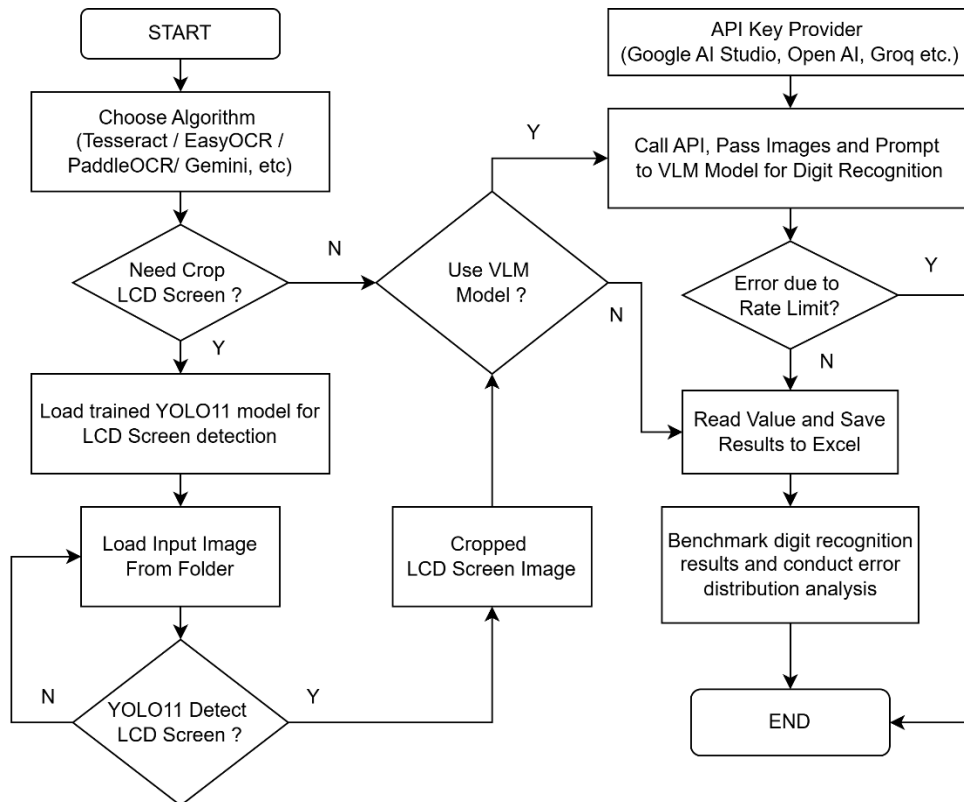


Figure 4. Flowchart of the study

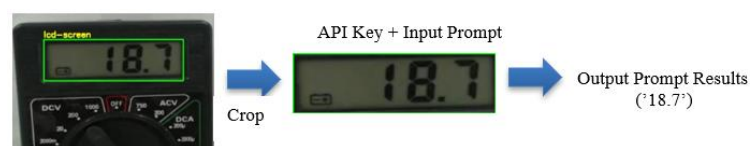


Figure 5. Digit recognition pipeline process with YOLO LCD screen cropping and VLM model inference

Several pre-trained models were used for the VLM, including Google Gemini-2.5-flash, Open AI GPT-4o, Google Paligemma-3b-mix-224, Llama-4-scout, and Qwen3.5-2b. An ablation experiment on a certain VLM model was then performed with a temperature setting variation from 0.0 to 2.0 to observe the effect of temperature setting on the reading result. During the usage of the VLM model with API calling, the usage of the rate limit quota was frequently checked to prevent failed reading due to the rate error limit. The rate limits were measured using three parameters: request per minute (RPM), tokens per minute (input) (TPM), and requests per day (RPD). The parameter limit depends on the type of user account, whether it is a free or an upgraded account with a provided billing account. As an example, based on the official website of Google AI Studio, a paid-tier 1 account version will have a maximum quota of 1000 RPM, 1,000,000 TPM, and 10,000 RPD. This study uses the paid-tier 1 version to resolve the limitations of the free-tier version of Gemini with 5 RPM, 250,000 TPM, and 20 RPD limit. Exceeding any of the three parameters results in an error due to the rate limit error. For example, in the case of Gemini API, if the RPM limitation was 20 in the free tier scheme, 21 RPM would trigger Gemini error, although the RPD usage did not exceed 250 during the day. The RPD

quotas will reset at midnight pacific time. Therefore, if Gemini Error occurred due to rate limit error during the experiment, the experiment was stopped and continued after the quotas were reset. If no Gemini error occurred, the obtained Gemini results were saved in an Excel file and analyzed further for benchmark analysis with the other OCR algorithm results.

An error distribution analysis was also conducted to investigate the type of error and how frequently it occurs. Several typical error types in OCR are deletion, insertion, substitution, wrong length, decimal shift, missing decimal, leading zero, no detection, and garbage output. However, several types of digital instruments are pre-configured to always have a reading value with a leading zero and fixed resolution. Thus, a leading zero cannot always be classified as an error if the ground truth (GT) or true label also has zero leading value.

2.3. Evaluation metrics

In this study, the performance of the YOLO11 model for LCD screen detection was evaluated using typical object detection evaluation metrics, which are precision (P), recall (R), average precision (AP), mean AP at intersection over union (IoU) threshold of 0.5 (mAP⁵⁰) and at mAP at IoU threshold of 0.5–0.95 with the interval steps of 0.05 (mAP⁵⁰⁻⁹⁵). All the typical object detection evaluation metrics are expressed in (1)–(5) [29], [30]:

$$P = \frac{TP}{TP+FP} \quad (1)$$

$$R = \frac{TP}{TP+FN} \quad (2)$$

$$AP = \int_0^1 P(R) dR \quad (3)$$

$$mAP^{50} = \frac{1}{c} \sum_{i=1}^c AP_c^{0.5} \quad (4)$$

$$mAP^{50-95} = \frac{1}{10c} \sum_{i=0}^9 \sum_{j=1}^c AP_c^{0.5+0.05i} \quad (5)$$

Where true positives denote (TP) (correctly predicted boxes), false positives denote (FP) (incorrectly predicted boxes), false negatives denote (FN) (missed ground-truth positive samples), and c denotes the number of classes. Several metrics will be used to evaluate digit recognition performance. The first is the accuracy of per-digit recognition (PDR) to evaluate PDR correctness, as expressed in (6) and (7):

$$PDR (\%) = 100 - DER \quad (6)$$

$$DER (\%) = \frac{D_{error}}{D_{total}} \quad (7)$$

Where DER is the digit error rate (%), D_{error} for the total number of digits incorrectly identified, and D_{total} for the total number of digits present in the sample image.

Furthermore, another metric was introduced with full sequence reading or (FSR) (%) to evaluate the complete digit sequence reading accuracy, as expressed in (8). This metric focuses on complete and exact digital measuring device recognition reading, where even a single digit or misplaced decimal point is classified as incorrect sample reading.

$$FSR (\%) = \left(\frac{S_{correct}}{S_{total}} \right) \times 100 \quad (8)$$

Finally, to evaluate the inference speed for the recognition of the digit, additional metrics of frames per second (FPS) in (9) were used with test hardware on a laptop graphics processing unit (GPU).

$$FPS = \frac{1000}{T_{total}} \quad (9)$$

Where T_{total} denotes the total time per image (in ms) for processing and inferencing.

Error distribution analysis was performed by calculating the number of errors by subtracted S_{total} and $S_{correct}$. The error types were then classified, and the percentage contribution of the error was calculated.

3. RESULTS AND DISCUSSION

Table 1 describes the training results of YOLO11n (nano) for detecting LCD screens of various digital measuring devices. After training for 317 epochs, the trained YOLO11n model obtained 99.8% P, 100% R,

99.4% mean AP with an IoU threshold value of 0.5 (mAP^{50}), and 99.0% mAP with varying IoU threshold from 0.5 to 0.95. Figure 6 shows inference result samples of the trained YOLO11n model on the DMD BEEI project dataset's test dataset. All 104 LCD images were detected and cropped with 100% accuracy.

Table 1. YOLO11n training results for DMD LCD screen detection

Class	P%	R%	$mAP^{50\%}$	$mAP^{50-95\%}$
LCD-screen	99.8	100	99.4	99.0

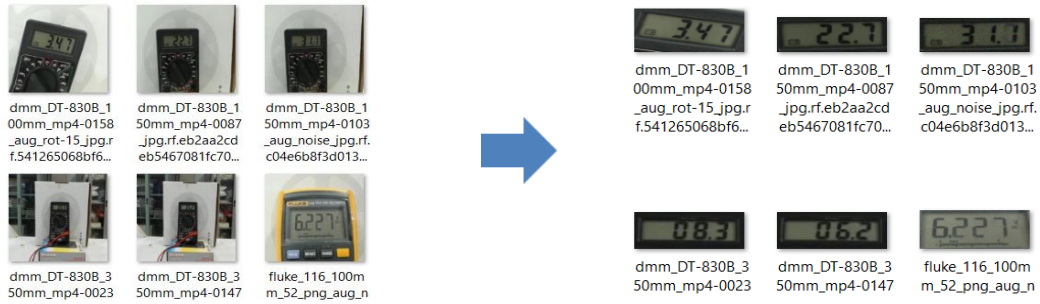


Figure 6. Sample results of YOLO detection and LCD screen cropping on different test dataset images

Table 2 describes the results of the ablation experiment to investigate the effect of custom training and implementation of YOLO LCD cropping on the digit recognition performance in all compared methods and algorithms. Custom training implementation could enhance the digit recognition performance of the current OCR baseline method (Tesseract OCR, EasyOCR, PaddleOCR, and Keras OCR) with increased PDR and FSR percentage. Furthermore, the implementation of YOLO LCD cropping was significant for the OCR baseline method. The result of Keras OCR without YOLO LCD cropping yielded 19.23% FSR, whereas the implementation of YOLO LCD cropping improved the FSR to 60.58%.

Table 2. Effect of the ablation experiment on custom training and YOLO cropping

Method/algorithm	Custo m train	+YOLO LCD cropping	D_{error}	D_{total}	PDR (%)	$S_{correct}$	S_{total}	FSR (%)	T_{total} (ms)	FPS
Tesseract OCR (default 'eng' model)	✗	☑	493	569	13.36	0	104	0.00	352	2.8
Tesseract OCR (beei_final)	☑	☑	245	569	56.94	30	104	28.85	162	6.2
EasyOCR (default 'en/eng' model)	✗	☑	267	569	53.08	16	104	15.38	59	16.9
EasyOCR (Custom model)	☑	☑	181	569	68.19	49	104	47.12	65	15.5
PaddleOCR (PP-OCRv5_server_rec)	✗	☑	126	569	77.86	42	104	40.38	41	24.6
Keras OCR (custom model)	☑	✗	282	569	50.44	20	104	19.23	134	7.4
Keras OCR (custom model)	☑	☑	105	569	81.55	63	104	60.58	87	11.5
YOLO11l (large) (custom model)	☑	✗	46	569	91.92	67	104	64.42	42	23.8
YOLO11l (large) (custom model)	☑	☑	127	569	77.68	52	104	50.00	42	23.5
Gemini-2.5-Flash (temp 0.0)	✗	✗	13	569	97.72	95	104	91.35	3507	0.3
Gemini-2.5-Flash (temp 0.0)	✗	☑	57	569	89.98	73	104	70.19	7197	0.1
Gemini-2.5-Flash (temp 0.4)	✗	✗	10	569	98.24	97	104	93.27	4786	0.2
Gemini-2.5-Flash (temp 0.4)	✗	☑	45	569	92.09	78	104	75.00	5572	0.2
Gemini-2.5-Flash (temp 1.0)	✗	☑	57	569	89.98	74	104	71.15	5435	0.2
Gemini-2.5-Flash (temp 2.0)	✗	☑	50	569	91.21	74	104	71.15	4798	0.2
GPT-4o (temp 0.4)	✗	✗	33	569	94.20	84	104	80.77	1752	0.6
GPT-4o (temp 0.4)	✗	☑	30	569	94.73	86	104	82.69	1610	0.6
Paligemma-3b-mix-224	✗	✗	73	569	87.17	61	104	58.65	72038	0.01
Paligemma-3b-mix-224	✗	☑	74	569	86.99	62	104	59.62	68944	0.01
Llama-4-scout (temp 0.4)	✗	✗	26	569	95.43	87	104	83.65	5802	0.2
Llama-4-scout (temp 0.4)	✗	☑	49	569	91.39	77	104	74.04	5631	0.2
Qwen3.5-2b*	✗	✗	28	569	95.08	90	104	86.54	11507	0.1
Qwen3.5-2b*	✗	☑	23	569	95.96	86	104	82.69	2600	0.4

*Note: use different prompt (since when using the default prompt, the decimal point was not read in all 104 samples, producing 0% FSR). The specific prompt used for Qwen3.5-2b is "Read the digital meter display exactly as shown. Include the decimal point if visible. For example, if you see "02.7" output "02.7," if you see "07.4" output "07.4," if you see "101.5" output "101.5." Ignore the battery icon and any other symbols on the display. Return ONLY the number with decimal point, no other text, and no units."

YOLO11l (large) and most of the VLM models generate fewer digit error (D_{error}), higher PDR and FSR values than the baseline OCR method. However, the effect of YOLO LCD cropping results in lower recognition performance with lower PDR and FSR in the YOLO11l, Gemini-2.5-flash, Llama-4-scout, and Qwen 3.5-2b models. For YOLO11l, this was mainly because the YOLO11l model was trained on the uncropped LCD screen dataset. Retraining YOLO11l on a cropped LCD screen dataset significantly improves the recognition performance. Meanwhile, the ablation experiment on the Gemini-2.5-flash model with temperature setting variation shows that the Gemini-2.5-flash with a temperature setting of 0.4 generates the best recognition performance. Compared with the higher temperature setting of 1.0 and maximum value of 2.0, the lower temperature setting of the Gemini model would generate better recognition performance in the case of digit recognition on digital measuring instruments. In terms of inference speed, PaddleOCR and YOLO11l yield the highest FPS compared with other OCR methods. However, Gemini-2.5-flash generates the lowest FPS but yields the best recognition performance. This was mainly because of its reliance on cloud-based API inference, which requires additional latency for image transmission and server-side processing, which significantly increases end-to-end inference time compared to locally deployed models.

A bootstrap sampling method was used to statistically evaluate the performance of each model. Figure 7 illustrates the bootstrap simulation results of 2000 iterations, showing the FSR value distribution for each model. The red dashed line denotes the actual FSR, whereas the green dotted lines represent the lower and upper bounds of the 95% confidence interval (CI). The results show that most of the VLM models achieve a higher estimated FSR value range and exhibit greater reliability with the typical bell-shaped curves compared to traditional OCR engines. Figure 8 describes the per-class character or digit recognition performance accuracy across all the models compared.

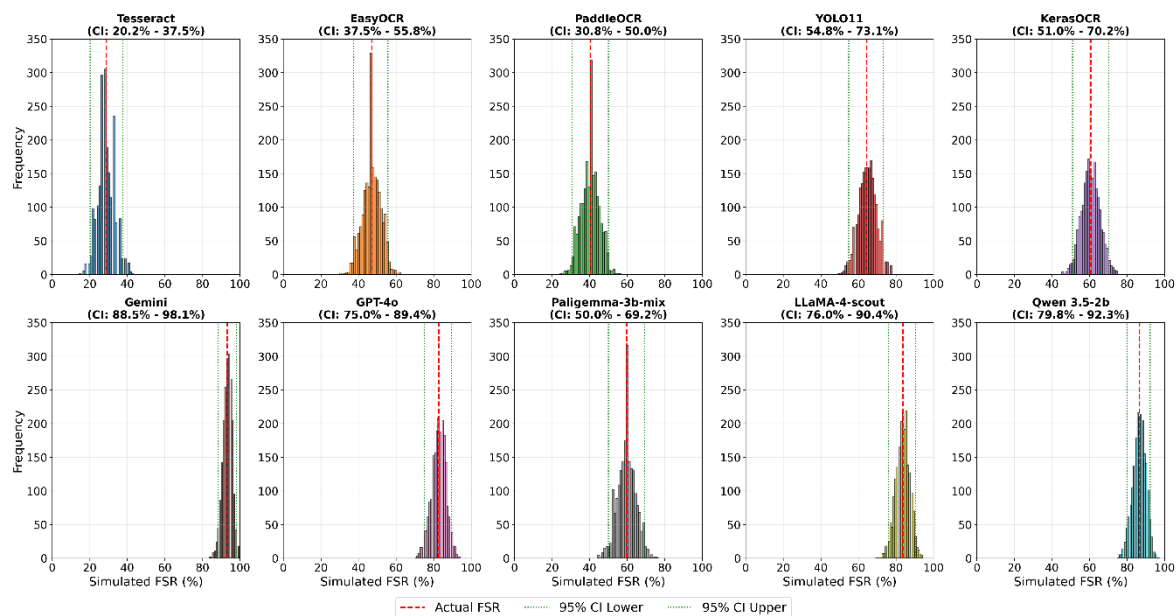


Figure 7. Distribution of 2000 bootstrap simulations for the FSR value in each model compared

Figure 9 presents a side-by-side comparison of the sample of correct and incorrect digit recognition for all the best versions of the model compared. The correct sample shows that the model prediction result has the same value with its true label or GT value. An incorrect example has different values between the prediction and GT values. A single incorrectly read sample image could have more than one type of error. For example, the incorrect PaddleOCR result in the Fluke 116 sample in Figure 9 has two error types, namely, wrong length and deletion. From the different prediction value (“2358”) and its GT value (“23.58”), it can be inferred that undetected decimal point (“.”) was the main cause of incorrect reading. This decimal missing type error is also common in YOLO11 results, as shown in Figure 9, in which the GT value was “20.53”, while the YOLO11 prediction was “2053” due to the undetected decimal point (“.”). Further modification of the YOLO11 architecture with an extra small detection layer in the detection head could resolve the issue in detecting extra small objects such as decimal point (“.”). Meanwhile, the implementation of YOLO cropping and training with a more augmented dataset for decimal point could resolve the undetected decimal point in the typical OCR baseline method (Tesseract OCR, EasyOCR, and PaddleOCR).

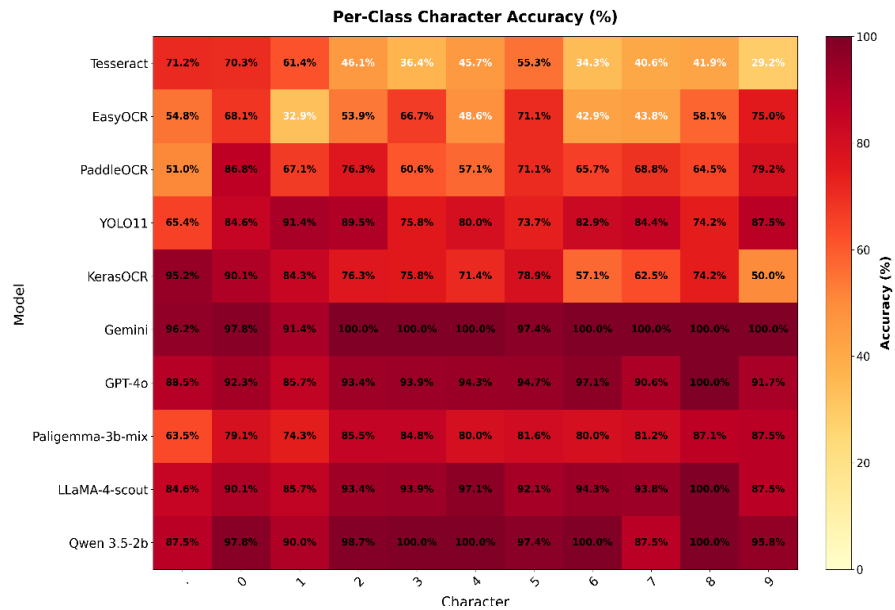


Figure 8. Comparison of per-class character or digit recognition accuracy across all models



Figure 9. Side-by-side comparison of correct and incorrect recognition sample results from all models

Table 3 summarizes the analysis of the error distribution for all the models compared. Overall, Gemini achieves the best performance with the lowest total error (7), followed by Qwen 3.5-2b (14), Llama-4-scout (17), GPT-4o (18), and YOLO11 (37). The error percentage of each error type was calculated based on its contribution to the 104 test images. YOLO11 has very low substitution errors, highlighting its strong digit classification capability. However, it suffers from higher deletion and decimal-related errors due to missing or undetected decimal points. Most VLM models also have very low substitution errors, but they suffer from decimal shift errors. Traditional OCR methods show higher error rates and are more affected by structural issues, especially in handling digit sequences and decimal digit placement.

Table 3. Summary results of the error distribution analysis

Model	Total	Error	Del (%)	Ins (%)	Subs (%)	WL (%)	DS (%)	DM (%)	LZ (%)	ND (%)	GO (%)
Tesseract	104	74	25.00	33.65	62.50	33.65	25.00	3.85	6.73	0.00	0.00
EasyOCR	104	46	41.35	3.85	21.15	42.31	13.46	23.08	20.19	0.00	0.00
PaddleOCR	104	59	53.85	0.96	7.69	53.85	0.96	45.19	2.88	0.00	0.00
KerasOCR	104	41	12.50	8.65	34.62	11.54	4.81	0.00	0.00	0.00	0.00
YOLO11	104	37	33.65	8.65	1.92	28.85	7.69	26.92	1.92	0.00	0.00
Gemini-2.5-flash	104	7	6.73	2.88	0.00	3.85	2.88	0.96	0.00	0.00	0.00
GPT-4o	104	18	15.38	6.73	2.88	9.62	8.65	1.92	1.92	0.00	0.96
Paligemma-3b	104	42	29.81	9.62	13.46	27.88	12.50	20.19	11.54	0.00	3.85
LLaMA-4-scout	104	17	14.42	6.73	1.92	8.65	9.62	4.81	3.85	0.00	0.96
Qwen 3.5-2b	104	14	12.50	11.54	1.92	0.96	12.50	0.00	1.92	0.00	0.00

*Note: del (deletion), ins (insertion), subs (substitution), wrong length (WL), decimal shift (DS), decimal missing (DM), leading zero (LZ), no detection (ND), and garbage output (GO).

Table 4 presents the results of the ablation experiment on the effect of Gemini prompt variations on recognition performance. In this case, the variation of Gemini prompt consists of nine (9) categories: original prompt, simple direct, format specific, context aware, strict numeric, with examples, chain of thought, edge cases, and minimal prompt. Based on Table 2, YOLO cropping was not used because it yields better recognition results than the cropped LCD version. The same temperature setting of 0.4 was applied in all of the experiments. The original prompt yielded the highest PDR with 98.42% and FSR with 95.19%. The second-best prompt variation was simple direct prompt with the same PDR accuracy of 98.42% and a slightly lower FSR of 93.27%. Meanwhile, the context-aware prompt generated the worst recognition performance (86.54%), which was less effective with the minimal prompt.

Table 4. Ablation experiment results on the effect of Gemini prompt engineering variation

Gemini prompt variation	Gemini prompt text	+YOLO LCD crop	PDR (%)	FSR (%)
Original prompt	Read the digital meter as it is without converting it to numbers (e.g., 02.7 or 07.4). Ignore the battery icon on the left. Return only the digit number without the addition of units. There is only one decimal point on the digital meter display, but there are no decimal points on the left most of the digital meter display.	✗	98.42	95.19
Simple direct	Extract only the numerical reading from this digital meter display. Return just the number (e.g., 02.7, 07.4, 10.5, 22.9). Ignore all text, labels, and icons.	✗	97.01	90.38
Format specific	Read the digital meter display and return ONLY the numeric value in this exact format: if it shows '00030' return '00030', if it shows '0.92' return '0.92', if it shows '10.5' return '10.5'. Preserve leading zeros and decimal points exactly as displayed.	✗	98.42	93.27
Context aware	This is a digital measurement device. Extract the main numeric reading from the display. Look for the largest or most prominent number showing the measurement value. For multimeters, ignore the range/setting numbers. Return only the numeric value without units.	✗	97.01	86.54
Strict numeric	From this image, identify and return ONLY the numeric value representing the measurement reading. Exclude: units (V, A, and psi), mode indicators (DC, AC), brand names, and any text. Return just the number.	✗	97.72	90.38
With examples	Extract the digital meter reading. Examples: '00030'→'00030', '0.92'→'0.92', '10.5'→'10.5', '22.9'→'22.9', '5.54'→'5.54'. Return only the numeric value as shown.	✗	97.72	90.38
Chain of thought	Step 1: identify the digital display area. Step 2: look for the main measurement value (largest digits). Step 3: extract the number exactly as shown, including decimal points and leading zeros. Step 4: return only that number. No additional text.	✗	96.84	90.38
Edge cases	Extract the digital meter reading. For displays with leading zeros (e.g., '00030'), preserve them. For decimal values ('0.92', '10.49'), keep the decimal point. For values without decimal (e.g., '22.9' actually has decimal), include it. Return only the numeric string.	✗	96.13	91.35
Minimal	What number is displayed? return only the number.	✗	97.19	90.38

To demonstrate the reliability of the given input prompt to the VLM model, a five-time repeated experiment of the Gemini-2.5-flash model (temperature setting of 0.4) with the best original prompt was performed, and the results are presented in Table 5. The results show that the repeated reading with Gemini-2.5 flash with the same original prompt has an average PDR of 98.10% with a standard deviation of 0.36% and an average FSR of 93.85% with a standard deviation of 1.29%. Thus, it demonstrates low variability, indicating stable performance for the Gemini-2.5-flash model inference process.

Table 5. Repeatability test result of Gemini with the same original prompt from five repeated experiments

Gemini prompt experiment	+ YOLO LCD cropping	D_{error}	D_{total}	PDR (%)	$S_{correct}$	S_{total}	FSR (%)	T_{total} (ms)	FPS
Original prompt (1 st experiment)	✗	10	569	98.24	97	104	93.27	4786.09	0.21
Original prompt (2 nd experiment)	✗	9	569	98.42	99	104	95.19	5709.76	0.18
Original prompt (3 rd experiment)	✗	9	569	98.42	99	104	95.19	6211.87	0.16
Original prompt (4 th experiment)	✗	13	569	97.72	97	104	93.27	3941.87	0.25
Original prompt (5 th experiment)	✗	13	569	97.72	96	104	92.31	4106.46	0.24
Average		10.8	569	98.10	97.6	104	93.85	4951.21	0.21
Standard deviation		2.0	0.0	0.36	1.3	0.0	1.29	990.49	0.04

Figure 10 illustrates the sample of a cropped LCD screen image that has been preprocessed with various techniques: grayscale-standard, grayscale-weighted, contrast-gamma correction, and denoise-gaussian blur to investigate the effect of image preprocessing technique on the recognition result.

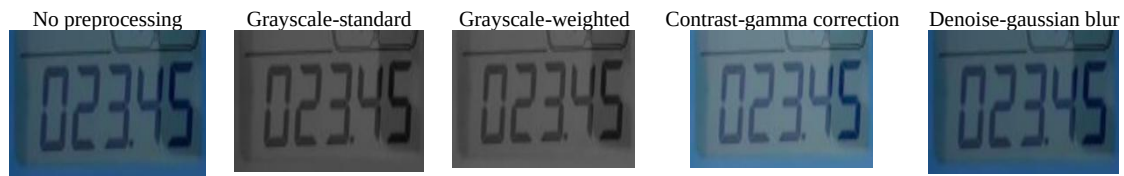


Figure 10. Comparison of various image preprocessing techniques implemented on cropped LCD screen

The results in Table 6 show that the implementation of most image preprocessing techniques, except for contrast-gamma correction, could enhance the FSR value by increasing correct prediction and reducing wrong prediction. The grayscale-standard technique demonstrates the best result with the highest FSR value improvement of 4.81% from the baseline without preprocessed image version. These findings highlight that simple intensity normalization techniques are more effective than aggressive enhancement methods in the case of seven-segment digital display.

Table 6. Result of image preprocessing variation on Gemini result with cropped LCD screen

Image preprocessing method	FSR (%)	Correct predictions	Total with GT	Wrong predictions	Min time (s)	Max time (s)	Average time (s)
Baseline (no preprocessing)	77.88	81	104	23	3.186	10.498	5.423
Contrast-gamma correction	77.88	81	104	23	2.807	27.984	5.669
Denoise-gaussian blur	78.85	82	104	22	3.613	20.384	6.13
Grayscale-weighted	81.73	85	104	19	3.908	22.263	6.523
Grayscale-standard	82.69	86	104	18	3.56	11.609	5.783

Comparison results of model performance trends across distance variation from the test dataset images used in this study are described in Figure 11. The comparison was conducted using bootstrapping method with resampled 8 samples of test images for 2000 iterations on each distance variance. The solid line represents the actual average accuracy FSR of the 8 original images. The shaded band area around the solid line represents the 95% CI from the bootstrapping. The result shows that Tesseract OCR, EasyOCR, and PaddleOCR generate unstable recognition performance with varying distance. Keras-OCR resulting in relatively stable accuracy across most of the distance variation. YOLO11 demonstrated strong performance at shorter distances (100-200 mm) but experienced sharp drop performance above 200 mm. This drop is primarily attributed to the model's

difficulty in reliably detecting the decimal point digit (“.”) in distant, low-resolution images. Notably, most of the VLM models consistently achieve the highest accuracy at most distance variations, except for Paligemma-3b-mix. Among the VLMs, Gemini stands out with near-perfect accuracy (approximately 100%) across all distance variations. GPT-4o, Qwen 3.5-2b, and Llama-4-scout also demonstrated highly robust performance, maintaining accuracy performance above 80% across the entire distance range. Overall, this finding indicates that, for this specific dataset in this study, modern VLMs model offer a more reliable solution for seven-segment digit recognition across varying camera distances, effectively mitigating the resolution degradation issues that severely hinder traditional OCR systems and object-detection-based models like YOLO11.

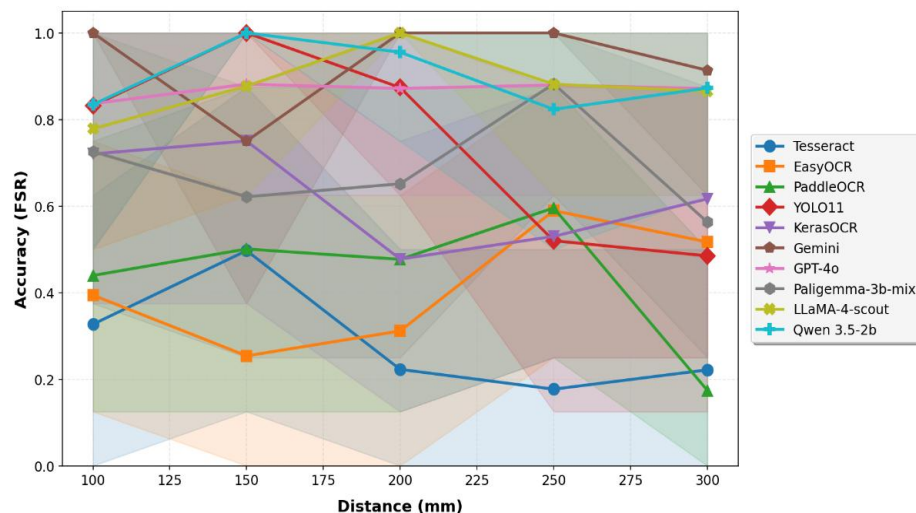


Figure 11. Comparison of model performance trends across distances using bootstrapping of 8 samples for each distance for 2000 iterations with 95% CI

A hardware experiment was set up to explore local deployment for latency reduction and real-time embedded system testing, as illustrated in Figure 12. A Raspberry Pi 5 was used as the edge device representative, which is integrated with a 7-inch LCD touchscreen and a Logitech C922 pro stream webcam with 3 MP, a maximum resolution of 1080 pixels for 30 FPS or 720 pixels for 60 FPS, and autofocus feature. A portable electronic scale was used as the digital measuring device for real-world testing in a household environment. The Gemini pipeline could read the unloaded portable electronic scale with a reading value of “0.000”. The terminal output of the Raspberry Pi 5 showed that the total inference time was 648.5 milliseconds or an inference speed of 1.5 FPS. This result explains why local GPU acceleration does not contribute to inference speed because the Gemini model operates as a cloud-based service.

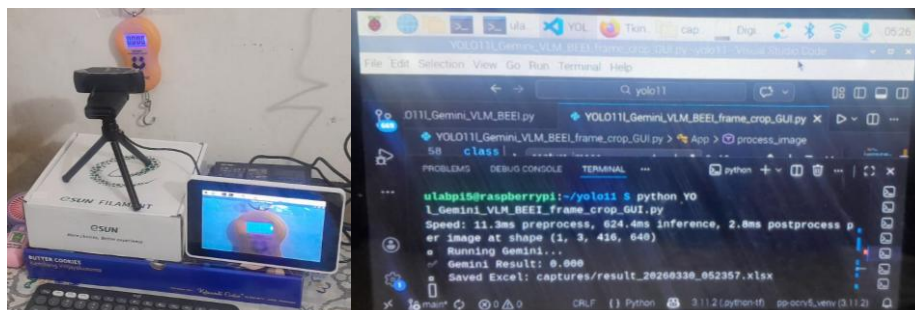


Figure 12. Hardware deployment on edge devices with real-world household environment testing

Although the inference speed was faster than the batch processing results in Table 2, it remains insufficient for real-time industrial monitoring applications. Thus, future research should focus on the local deployment of a small-scale, quantized VLM model directly onto the edge device to fundamentally address the

latency and eliminate cloud network-based dependency. By deploying the lightweight VLM models entirely offline on the laptop, PC, or Raspberry Pi 5’s NPU/GPU, the API calls can be bypassed entirely. In the near future, this local deployment strategy will drastically reduce inference time to millisecond ranges, enabling real-time embedded inference independent of cloud infrastructure.

4. CONCLUSION

In this study, the multimodal capabilities of VLM were explored to perform seven-segment digit recognition tasks. The comparison results show that the Gemini model was superior to other baseline OCR and VLM methods. Various ablation experimental results demonstrate that the digit recognition results with the VLM model depend on the command prompt input quality. The repeatability test of inference with the same prompt demonstrates that the Gemini VLM model reading with the same prompt and same condition has low variability and indicates stable performance with an average PDR value of 98.10% with a standard deviation of 0.36% and an average FSR value of 93.85% with a standard deviation of 1.29%. The results show that LCD screen cropping with YOLO11 could enhance the recognition results for most baseline OCR methods, such as Tesseract OCR, EasyOCR, PaddleOCR, and Keras OCR. However, the uncropped LCD screen yields higher recognition accuracy with higher PDR and FSR values for YOLO11 and most of the VLM methods. Simple intensity normalization techniques, such as the grayscale method, were found to be more effective than aggressive enhancement methods in image preprocessing. The result also shows that Gemini inference is stable with high FSR accuracy across variation in distance. YOLO11 experiences drop performance at higher distances due to difficulty in detecting objects with extra small decimal digits. Meanwhile, the other baseline OCR method shows unstable performance at most distances. The effect of local deployment of lightweight VLM model on edge devices to reduce latency in performing inference should be investigated further. In future research, fine-tuning the VLM model or developing a modified VLM model architecture specifically for seven-segment digit recognition should also be conducted.

ACKNOWLEDGMENTS

The authors express gratitude to the National Research and Innovation Agency (BRIN) for providing the facility used in this study. The authors also acknowledge the use of several AI tools (ChatGPT, Trinka) to support the refinement of language, clarity, and structure during the preparation of the manuscript.

FUNDING INFORMATION

The authors state that the publication of this research is funded by the 2025-2026 Seed Funding Publication Grant Faculty of Engineering Universitas Indonesia with contract number PKS-3699/UN2.F4.D/PPM.00.00/2025.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Adindra Vickar Ega	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Dadang Gunawan	✓				✓	✓				✓		✓	✓	✓
Abdul Halim	✓					✓				✓		✓		
Rudi Anggoro		✓			✓	✓				✓	✓	✓		
Samodro														

C : C onceptualization	I : I nterpretation	Vi : V isualization
M : M ethodology	R : R esources	Su : S upervision
So : S oftware	D : D ata Curation	P : P roject administration
Va : V alidation	O : O rganizing - O riginal Draft	Fu : F unding acquisition
Fo : F ormal analysis	E : E valuation - E diting	

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data supporting the findings of this study are openly available in Roboflow Universe at https://universe.roboflow.com/escaproject/dmd_dataset_lcd_screen_detection for the DMD LCD Screen Detection Dataset and at <https://universe.roboflow.com/backupescaproject/bee-project> for the DMD BEEI Project Dataset.

REFERENCES

- [1] E. Sarrion, "What Is ChatGPT?," in *Exploring the Power of ChatGPT*, Berkeley, CA: Apress, 2023, pp. 3–8, doi: 10.1007/978-1-4842-9529-8_1.
- [2] F. I. Malfatti, "ChatGPT, Education, and Understanding," *Social Epistemology*, vol. 39, no. 6, pp. 652–666, Nov. 2025, doi: 10.1080/02691728.2025.2449599.
- [3] G. Lotfian, K. Parekh, and P. P. Suthar, "Performance Review of Meta LLaMa 3.1 in Thoracic Imaging and Diagnostics," *iRADIOLOGY*, vol. 3, no. 4, pp. 279–288, Aug. 2025, doi: 10.1002/ird3.70013.
- [4] B. Acharya, S. Koirala, R. Chaudhari, K. Gautam, and B. Pokharel, "A Semantic Retrieval and Generation Framework for Alumni Intelligence Using LLaMA 3," *Journal of Engineering Issues and Solutions*, vol. 4, no. 1, pp. 422–435, Jul. 2025, doi: 10.3126/joeis.v4i1.81598.
- [5] J. Stratton, "An Introduction to Microsoft Copilot," In: *Copilot for Microsoft 365*, Inside Copilot, Apress, Berkeley, CA, 2024, pp. 19–35, doi: 10.1007/979-8-8688-0447-2_2.
- [6] D. Portelli *et al.*, "ChatGPT and Microsoft Copilot for Cochlear Implant Side Selection: A Preliminary Study," *Audiology Research*, vol. 15, no. 4, p. 100, Aug. 2025, doi: 10.3390/audiolres15040100.
- [7] M. Imran and N. Almusharraf, "Google Gemini as a next generation AI educational tool: a review of emerging educational technology," *Smart Learning Environments*, vol. 11, no. 1, p. 22, May 2024, doi: 10.1186/s40561-024-00310-z.
- [8] R. Islam and O. M. Moushi, "GPT-4o: The Cutting-Edge Advancement in Multimodal LLM," in *Lecture Notes in Networks and Systems*, vol. 1426, 2025, pp. 47–60, doi: 10.1007/978-3-031-92611-2_4.
- [9] P. Perera and M. Lankathilake, "Preparing to Revolutionize Education with the Multi-Model GenAI Tool Google Gemini? A Journey towards Effective Policy Making," *Journal of Advances in Education and Philosophy*, vol. 7, no. 08, pp. 246–253, Aug. 2023, doi: 10.36348/jaep.2023.v07i08.001.
- [10] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, "Unleashing the potential of prompt engineering for large language models," *Patterns*, vol. 6, no. 6, p. 101260, Jun. 2025, doi: 10.1016/j.patter.2025.101260.
- [11] M. K. Siam, H. Gu, and J. Q. Cheng, "Programming with AI: Evaluating ChatGPT, Gemini, AlphaCode, and GitHub Copilot for Programmers," in *ICCA 2024 - 3rd International Conference on Computing Advancements, 2024*, Oct. 2025, pp. 346–354, doi: 10.1145/3723178.3723224.
- [12] Y. Zi, L. Li, A. Guha, C. J. Anderson, and M. Q. Feldman, "I Would Have Written My Code Differently': Beginners Struggle to Understand LLM-Generated Code," in *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, Jun. 2025, pp. 1479–1488, doi: 10.1145/3696630.3731663.
- [13] M. Irshad, "Revolutionizing Healthcare Delivery: Evaluating the Impact of Google's Gemini AI as a Virtual Doctor in Medical Services," *Journal of Artificial Intelligence, Machine Learning and Data Science*, vol. 2, no. 4, pp. 1618–1625, Dec. 2024, doi: 10.51219/jaimld/mahnoor-mughal/362.
- [14] N. Taymour, S. M. Fouda, H. H. Abdelrahman, and M. G. Hassan, "Performance of the ChatGPT-3.5, ChatGPT-4, and Google Gemini large language models in responding to dental implantology inquiries," *Journal of Prosthetic Dentistry*, vol. 134, no. 6, pp. 2427–2434, Dec. 2025, doi: 10.1016/j.prosdent.2024.12.016.
- [15] M. Tatar, S. Farokhi, A. A. Foumani, E. Uzunlar, and O. M. Araz, "Application of Gemini in Public Health Amid the Artificial Intelligence Era," *IEEE Engineering Management Review*, vol. 54, no. 1, pp. 21–27, Feb. 2026, doi: 10.1109/EMR.2025.3577913.
- [16] P. D. A. Nguyen, "Gemini Google: a Potential Tool for English Learning," *Thu Dau Mot University Journal of Science*, vol. 6, no. 3, Aug. 2024, doi: 10.37550/tdmu.ejs/2024.03.586.
- [17] H. Y. Lee and Y. M. Huang, "Bridging STEM Education and Ubiquitous Learning: A Case Study on Developing a LINE Chatbot with Google's Gemini for Virtual Peer Collaboration," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14786, pp. 237–246, 2024, doi: 10.1007/978-3-031-65884-6_25.
- [18] T. Joseph, N. Sanghavi, S. Kanyal, K. Majumder, H. Seidu-Aroza, and A. Godavarthi, "Comparative Analysis of ChatGPT and Google Gemini in the Creation of Patient Education Materials for Acute Appendicitis, Cholecystitis, and Hydrocele," *Indian Journal of Surgery*, vol. 87, no. 1, pp. 117–122, Feb. 2024, doi: 10.1007/s12262-024-04112-y.
- [19] U. Gupta, A. Kumar, A. Gupta, G. Raj, and A. P. Agrawal, "Advances in Handwritten Character Recognition: A Comparison of OCR and Large Language Model-Based Approaches," in *Proceedings of International Conference on Emerging Technologies and Innovation for Sustainability, EmergIN 2024*, Dec. 2024, pp. 192–198, doi: 10.1109/EmergIN63207.2024.10961487.
- [20] T. Do, D. P. Tran, A. Vo, and D. Kim, "Reference-Based Post-OCR Processing with LLM for Precise Diacritic Text in Historical Document Recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 27, pp. 27951–27959, Apr. 2025, doi: 10.1609/aaai.v39i27.35012.
- [21] A. Sharshar, L. U. Khan, W. Ullah, and M. Guizani, "Vision-Language Models for Edge Networks: A Comprehensive Survey," *IEEE Internet of Things Journal*, vol. 12, no. 16, pp. 32701–32724, Aug. 2025, doi: 10.1109/JIOT.2025.3579032.
- [22] X. Li, C. Wen, Y. Hu, Z. Yuan, and X. X. Zhu, "Vision-Language Models in Remote Sensing: Current progress and future trends," *IEEE Geoscience and Remote Sensing Magazine*, vol. 12, no. 2, pp. 32–66, Jun. 2024, doi: 10.1109/MGRS.2024.3383473.
- [23] N. AlDahoul *et al.*, "Multitasking vision language models for vehicle plate recognition with VehiclePaliGemma," *Scientific Reports*, vol. 15, no. 1, p. 26189, Jul. 2025, doi: 10.1038/s41598-025-10774-9.
- [24] S. Badla, "Improving the Efficiency of Tesseract OCR Engine," San Jose State University, San Jose, CA, USA, 2014, doi: 10.31979/etd.5avd-kf2g.
- [25] R. N. Suharto, M. H. Z. A. Faroby, and Y. Setiawan, "Comparative Evaluation of YOLO-based Models for Indonesian License Plate Detection and Recognition with EASY OCR," *Procedia Computer Science*, vol. 269, pp. 943–952, 2025, doi: 10.1016/j.procs.2025.09.037.
- [26] M. Troller, "Practical OCR system based on state of art neural networks," *Czech Technical University in Prague*, pp. 1–22, 2017.
- [27] W. Zhan, A. Pang, W. Yang, W. Zhang, Y. Luo, and Q. Zhao, "Auto-PCI: an approach to automated reading of single-pointer meters with adaptive ranges based on YOLOv5s and PP-OCrv3," *Measurement Science and Technology*, vol. 36, no. 3, p. 035410, Mar. 2025, doi: 10.1088/1361-6501/adb76a.

- [28] M. Bakirci, P. Dmytrovych, I. Bayraktar, and O. Anatoliyovych, "Multi-Class Vehicle Detection and Classification with YOLO11 on UAV-Captured Aerial Imagery," in *2024 IEEE 7th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD)*, Oct. 2024, vol. 40, no. 6, pp. 191–196, doi: 10.1109/APUAVD64488.2024.10765862.
- [29] R. Khanam, T. Asghar, and M. Hussain, "Comparative Performance Evaluation of YOLOv5, YOLOv8, and YOLOv11 for Solar Panel Defect Detection," *Solar*, vol. 5, no. 1, p. 6, Feb. 2025, doi: 10.3390/solar5010006.
- [30] M. Tariq and K. Choi, "YOLO11-Driven Deep Learning Approach for Enhanced Detection and Visualization of Wrist Fractures in X-Ray Images," *Mathematics*, vol. 13, no. 9, p. 1419, Apr. 2025, doi: 10.3390/math13091419.

BIOGRAPHIES OF AUTHORS



Adindra Vickar Ega    received the B.Eng. degree in physics engineering and M.Eng. degree in electrical engineering from Universitas Gadjah Mada, Yogyakarta, Indonesia in 2011 and 2022, respectively. Currently, he is pursuing Ph.D. degree in electrical engineering with the Department of Electrical Engineering, Universitas Indonesia. He is a junior researcher at the Research Center for Equipment Manufacturing Technology, National Research and Innovation Agency of Republic Indonesia (BRIN). His research interests include metrology, measurement, embedded systems, internet of things, computer vision, image processing, and artificial intelligence. He can be contacted at email: adindra.vickar@ui.ac.id.



Dadang Gunawan    received the B.Eng. degree in electrical engineering from Universitas Indonesia in 1983, M.Eng. degree from Keio University, Japan in 1989, and Ph.D. degree from University of Tasmania, Australia, in 1995. He has been a Professor in the Department of Electrical Engineering, Universitas Indonesia, since 2004. He has published hundreds of academic articles as a first author or a co-author in proceedings and international journals. His research interests include wireless and signal-processing technology. He can be contacted at email: guna@eng.ui.ac.id.



Abdul Halim    received the B.Eng. and M.Eng. degree in electrical engineering from Keio University, Japan in 1995 and 1997, respectively. He received his Ph.D. degree from Tokyo Institute of Technology, Japan in 2000. He is currently an assistant professor and secretary at the Department of Electrical Engineering, Universitas Indonesia. He has published hundreds of academic articles as a first author or a co-author in proceedings and international journals. His research interests include optimal control systems, modeling and optimization, and machine learning. He can be contacted at email: a.halim@ui.ac.id.



Rudi Anggoro Samodro    received the B.Sc. in physics from Universitas Gadjah Mada, Yogyakarta, Indonesia, in 2005, M.Sc. and Ph.D. degree in science of measurement from University of Science and Technology South Korea in 2012 and 2021, respectively. Currently, he is a senior researcher and head of measurement system manufacturing technology research group at Research Center for Equipment Manufacturing Technology, National Research and Innovation Agency (BRIN). His research interests include metrology, measurement, simulation, and instrumentation. He can be contacted at email: rud001@brin.go.id.