GaugeTracker: AI-Powered Cost-Effective Analog Gauge Monitoring System

Beitong Tian[1], Mingyuan Wu[1], Ruixiao Zhang[1], Haozhen Zheng[1], Bo Chen[1], Yaohui Wang[1], Shiv Trivedi[1], Shanbo Zhang[1], Robert Bruce Kaufman[2], Leah Espenhahn[2], Gianni Pezzarossi[2], Mauro Sardela[3], John Dallesasse[2], Klara Nahrstedt[1]

[1]*Coordinated Science Laboratory,* [2]*Holonyak Micro & Nanotechnology Laboratory,* [3]*Materials Research Laboratory*
University of Illinois at Urbana-Champaign, Champaign, USA
{*beitong2,mw34,ruixiao,haozhen3,boc2,yaohuiw2,shivvt2,shanboz2,rbkaufm2,leahe2,gpezza2,sardela,jdallesa,klara*}*@illinois.edu*

***Abstract*—Automating analog gauge readings is essential for providing stakeholders with timely alerts about abnormalities in physical properties measured by gauges, such as pressure, and for offering detailed historical data to improve understanding of the work environment. However, existing systems face challenges in balancing accuracy, continuity, reading latency, network bandwidth usage, and cost. In this study, we introduce *GaugeTracker*, an end-to-end system to address these challenges. Our proposed method, based on template matching for gauge reading, precisely determines the current angle of the gauge pointer, significantly outperforming state-of-the-art baselines with an average error of 1.81 degrees. By leveraging the versatility of large vision-language models, we develop a pipeline for automatically generating accurate and realistic gauge templates for each specific gauge at various readings on the server. Deployed on the world's most affordable IoT camera, which is mounted in front of a gauge using our customized camera holder, our prototype system can read the gauge 7 times per second by processing entirely on the device. This delivers continuous and accurate gauge readings across diverse environmental conditions. Furthermore, with a cost of merely $10 per gauge, our system offers a highly cost-effective solution for real-time analog gauge monitoring.**

***Index Terms*—Computer Vision; Internet of Things; Analog Gauge Transcription; Template Matching; Large Vision-language Model; Synthetic Data; Real-Time Data Processing;**

I. Introduction

Analog gauges, depicted in Fig. 1, are widely used in settings such as factories and research laboratories to measure critical parameters like pressure. Traditionally, reading these gauges involves manual, periodic checks by specialized personnel, a method that is labor-intensive, error-prone, and unable to provide real time data. While replacing analog gauges with digital ones could solve these issues, such an upgrade is often prohibitively expensive and not always practical. A popular alternative for automating gauge readings is the use of cameras to capture images of the gauges and computer vision algorithms to interpret the readings from these images.

Various camera-based systems for transcribing analog gauge readings have been developed, each exhibiting unique characteristics [1]–[7]. These systems can be classified based on several criteria: accuracy, continuity (whether the reader is always on), gauge reading rate, and resource usage, including network bandwidth and the monetary cost of purchasing the device and service.

This research was funded by the NSF (award number 2126246). The opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the view of the NSF.

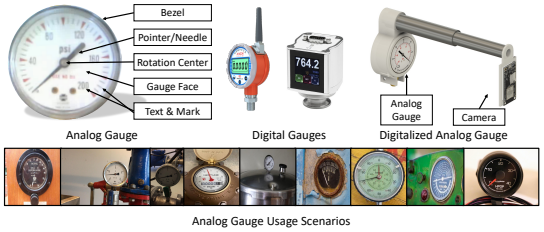


Fig. 1: Left: Analog gauge components and their names as used in this paper. **Middle:** Digital gauge examples. **Right:** Digitalized analog gauge with cameras. **Bottom:** Analog gauge usage in diverse environments.

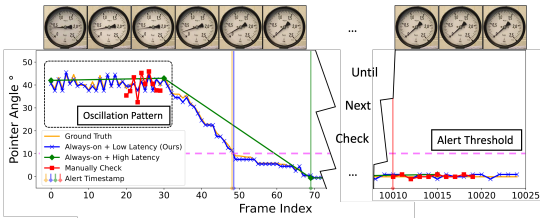


Fig. 2: Comparison of various analog gauge monitoring systems: This figure shows the actual performance of our system, represented by the blue line, achieving a mean absolute error in angle readings of 0.66°. The performance of competing systems is simulated based on their respective characteristics.

Figure 2 provides a comparative illustration of various systems utilizing a real gauge video recording from [5] to demonstrate the differences between these systems. This video showcases an initial oscillation of the gauge pointer, followed by a steady change to 0, which indicates a pressure drop in the machine. To detect the pressure drop in this video, our objective is to identify when the gauge pointer falls below a set threshold, indicated by an angle of less than 10 degrees. Different colors represent the readings from various systems, each corresponding to different timestamps indicated by frame indices. The orange line denotes the ground truth for the pointer angle reading, enabling the trigger of an alert around the 48th frame index.

The red line illustrates the conventional manual checking method—whether by an operator, mobile phone [1], or inspection robot [8]—highlighting its limitation due to the absence of continuous monitoring. The initial manual check appears normal as shown in the left part of Fig. 2, but the lack of ongoing reading delays the detection of pressure loss until the subsequent check as shown in the right part of Fig. 2, thus failing to issue a timely alert.

Contrastingly, the system depicted by the green line pro-

vides persistent measurement. However, due to extensive computational demands [6] or bandwidth constraints, the time required to process each frame locally or to offload the computational tasks by transmitting high-resolution images to the server results in a reduced gauge reading rate. Despite its ability to detect anomalies earlier, notably around the 68th frame index, the system's low sampling rate limits its ability to identify oscillation patterns. Such patterns could act as early indicators of equipment malfunction.

Our system's performance is depicted by the blue line. It facilitates continuous and rapid monitoring of the gauge pointer's angle. This system provides detailed information and timely alerts to users, aiming to exceed the limitations of existing methods through persistent, real-time, on-device tracking.

Unlike previous methodologies that depend on complex, computationally intensive computer vision algorithms or deep learning approaches, which necessitate extensive datasets for training and present deployment challenges on resource-constrained hardware, our work adopts the concept of template matching for analog gauge reading. This technique compares newly captured images against a set of predefined templates representing specific gauge readings. Generating these templates is non-trivial, as various steps necessary for template creation—such as segmenting the pointer and removing it from the background—are currently performed manually on a case-by-case basis. To solve this problem, we design a pipeline to automate the template generation process using advanced large vision-language models for their zero-shot generalization capabilities on various computer vision tasks like grounding, segmentation, and object removal. We validate our method against the latest benchmarks, delivering state-of-the-art results. We implement our method on a low-cost IoT camera (costing less than $10) and test the prototype in various environments to confirm its effectiveness and practicality.

In summary, our work has the following contributions:

(**1**) Unlike previous approaches that focus merely on the accuracy of analog gauge reading, we jointly optimize metrics of accuracy, continuity, latency, cost, scalability, and resource usage to address the multifaceted demands of real-world applications.

(**2**) We present a template-matching method for analog gauge reading systems with limited resources, which demonstrates competitive accuracy. Complementing this, our fully automated template generation pipeline harnesses advanced large vision-language models, marking an innovative application in the field.

(**3**) We comprehensively test our method and implement a real system that showcases the cost-efficiency and practicality of our method for large-scale analog gauge monitoring.

II. Related Work

A. Analog gauge transcription

Over the past decades, automatic analog gauge monitoring has been approached through two main methodologies: traditional computer vision [3]–[5], [7]–[21] and advanced deep learning techniques [1], [2], [6], [21]–[32]. Traditional computer vision techniques, as per [3], utilize Hough Transform for gauge and pointer detection, refined over time to mitigate issues like lighting [9], camera angles [12], glare [8], and shadows [10]. Despite advancements, limitations in adaptability and the necessity for high-end camera and image processing hardware curtail their widespread application.

Deep learning methods, divided into key point detection [1], [2] and classification approaches [25], offer improvements in angle determination and position classification. However, the lack of extensive training datasets and the demand for high computational resources pose significant challenges, with attempts to overcome these through synthetic data [1], [2] and data augmentation [25].

Our approach revisits the concept of template matching, combining it with advanced vision and vision-language models to streamline template generation and address the scalability and hardware limitations of previous methods.

B. Large Vision-Language Model in Real-World Applications

Vision-language foundation models [33], [34], pre-trained on extensive web-scale datasets, have demonstrated remarkable zero-shot generalization capabilities for downstream vision-language tasks. These include applications such as promptable image segmentation across new data distributions, where they outperform traditional methods. This advantage is particularly significant in scenarios where domain-specific data or human-annotated ground truth is scarce. Prior research has highlighted the strong generalization potential of these models within various fields, including geographical analysis (e.g., watershed segmentation [35]) and the medical sector (e.g., computer-aided diagnosis [36]). Our work is the first to employ vision-language foundation models for the task of transcribing analog gauges. Extensive experiments demonstrate that, the analog gauges domain, characterized by limited ground truth availability, is benefited from existing foundation models with exceptional generalization capabilities as well.

III. System Design

In this section, we introduce our analog gauge monitoring system, detailed in Fig. 3. The system consists of a central server and IoT cameras. The server manages template generation and hosts the user interface, while IoT cameras, positioned on each gauge, use these templates for gauge reading. Communication between cameras and the server occurs over wireless networks like Wi-Fi networks.

Each deployment comprises initialization and tracking stages. Initially, the camera is set up on the gauge by the user and controlled via the user interface to capture a high-resolution image of the gauge's front face for template generation. The generation process happens offline on the server and takes about 10-20 seconds to finish. Then, templates generated for the specific gauge are transmitted to the camera, where they are stored for online real-time tracking. The template matcher, running locally on the camera, continuously analyzes images and sends angle readings back to the server for user alerts and visualization.

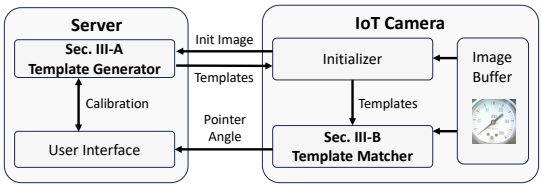


Fig. 3: System Overview

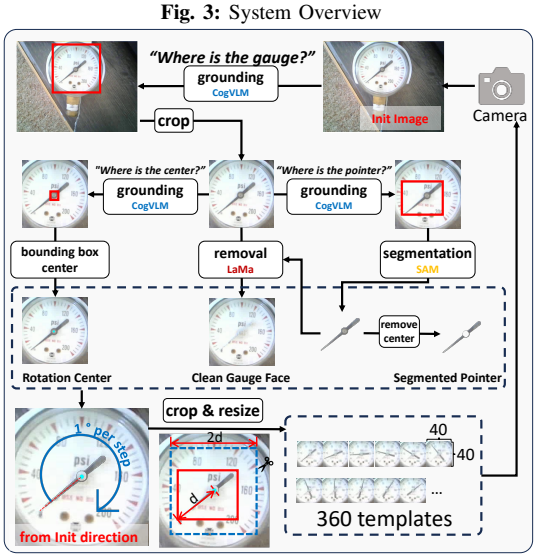


Fig. 4: Template Generation Pipeline

Our system measures the gauge pointer's angle. This angle is then utilized in conjunction with a user interface for various applications. Before we delve into the methodology, it's important to define our coordinate system for angle measurements, as illustrated in the left image of Fig. 6. Our system initializes at 0 degrees, which aligns with the gauge's minimum value, and angles increase clockwise up to 359 degrees.

A. Template Generation

The primary challenge of the template-matching method is acquiring gauge images with varied readings without manually adjusting the gauge, and we need to do this for each gauge. To achieve this, we can synthesize these images by rotating the pointer around the rotation center on a clean gauge face. However, obtaining these components is challenging. Previous attempts [25] used image processing software, but these methods required manual intervention, becoming impractical and non-scalable due to diverse gauge appearances. To automate this process, algorithms that can locate the rotation center, segment the pointer, and cleanly remove the pointer from the gauge face are required. Traditional computer vision algorithms struggle to perform these tasks effectively. Training specific deep learning models for these purposes demands the collection and labeling of large datasets, which is both complex and costly. We innovatively utilize the generalizability and zero-shot capabilities of large vision and large vision-language models to automatically extract these components without the need to train our own models. This method demonstrates excellent performance, as shown in Fig. 4.

Large Vision Language Model Overview: Existing individual foundation vision language models cannot localize the gauge, detect and segment the pointer, and cleanly remove the pointer. We build a novel multi-step vision language model pipeline that harnesses distinct capabilities from two state-of-the-art large models, CogVLM [33] and SAM [34], and an inpainting model LaMa [37]. CogVLM excels in grounding, identifying, and localizing elements within images. In our pipeline, we use this feature to detect and mark the gauge, the gauge pointer, and its rotation center with bounding boxes. SAM offers zero-shot segmentation, allowing for precise object segmentation without extensive manual labeling polygons. We apply this with CogVLM's bounding boxes to segment the pointer accurately. Lastly, we apply LaMa which can remove specific image elements cleanly. We utilize this to erase the pointer from the gauge, revealing a clean gauge face for template generation.

Generation Process: Given an initialization image, we initiate our process by querying CogVLM with "Where is the gauge?" to locate and crop the gauge's bounding box. Based on this cropped gauge image, we proceed with different processes to extract the following components:

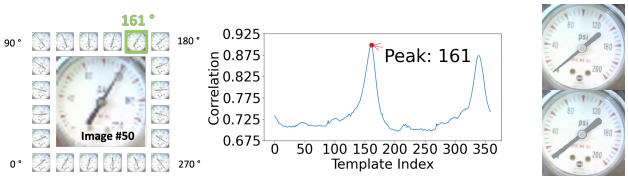
Rotation Center: We query CogVLM with "Where is the rotation center of the gauge?" on the gauge image. The center of the resulting bounding box is set as the rotation center.

Segmented Pointer: Initially, we query CogVLM with the prompt "Where is the pointer of the gauge?" to identify the pointer's bounding box in the gauge image. This bounding box serves as the box prompt, and together with the rotation center, which is used as the point prompt, for the SAM model to segment the pointer. To enhance accuracy, we mask a circular area at the pointer's center, with a radius half the width of the bounding box we used to calculate the rotation center. Furthermore, we color the segmented pointer to match the average color of the pointer.

Clean Gauge Face: We expand the size of the pointer mask through dilation and then process the gauge image using this expanded mask via an inpainting pipeline to remove the pointer, yielding a clean gauge face.

After getting all three components, we rotate the **segmented pointer** around the **rotation center** on the **clean gauge face**, involving 359 clockwise rotations in 1° increments. This process creates 360 templates indexed from 0 to 359 for various gauge readings. To process these templates on resource-constraint devices, we crop and resize them to reduce the image size before transmitting them to the IoT camera. As depicted in Fig. 4, we perform a center crop to isolate a square centered on the rotation center, with a width of $2 \times d$, twice the length (d) of the pointer's longest segment, which is the maximum distance from the rotation center to the corners of the pointer's bounding box. After cropping, we downscale the template to a manageable size with interpolation methods.

Calibration: Using our user interface, as depicted in Fig. 6, we establish the correlation between the index of each template and the actual angle within our coordinate system. The interface calculates Δ, the angular difference between the pointer's initial direction in the initialization image and its alignment with the gauge's minimum reading. Consequently,



(a) (b) (c)

Fig. 5: Template Matching Examples. **(a)** Outer circle: generated templates. Inner image: the new input image captured by the camera. **(b)** Pearson correlations between the new input image and all templates. **(c)** Top: Small pointer template - template example with the original segmented pointer. Bottom: Large pointer template - template with the enlarged pointer.

the angle for each template T_i, where $i \in \mathbb{Z}$ is the index of the template within the range $\{0, \ldots, 359\}$ is determined by:

$$angle_{T_i} = (\Delta + i) \bmod 360 \tag{1}$$

For instance, if Δ is 191°, the angle for the 10th template is computed as $(191 + 10) \bmod 360 = 201°$.

B. Template Matching

Given a new image, we calculate its correlation with each template generated in the previous step. The image, denoted as I, and each template T_i, are RGB images with the dimensions $I, T_i \in \mathbb{R}^{N \times N \times 3}$. The template matching process involves calculating the Pearson correlation between the new image and each template to find the best-matched template. The index of this template is then used to calculate the pointer angle. A toy examples is shown in Fig. 5a&5b.

The Pearson correlation between the image I and a template T_i is given by (each image is treated as a 1D vector):

$$\text{corr}(I, T_i) = \frac{\sum \left((I - \bar{I}) \cdot (T_i - \bar{T_i}) \right)}{\sqrt{\sum (I - \bar{I})^2 \cdot \sum (T_i - \bar{T_i})^2}}$$

, where the $\bar{I}$ and $\bar{T_i}$ are averaged values of I and T_i.

To find the best match, the index of the template with the highest correlation is identified as:

$$i_{\text{best}} = \arg\max_i \text{corr}(I, T_i)$$

and the current pointer angle can be calculated with Eq. 1.

C. Coarse-to-fine Template Matching

The template matching algorithms we previously discussed necessitate 360 comparisons. Although Pearson correlation is computationally efficient, executing it 360 times per frame can be burdensome on resource-limited hardware. To enhance efficiency, we introduce a coarse-to-fine method that initially determines a broad range where the pointer is likely situated, followed by a detailed search within this narrowed scope.

To ascertain the likely range of the pointer, we create a new set of templates named large pointer templates featuring enlarged pointers, depicted in Fig. 5c. The process of enlargement involves rotating the segmented pointer by n degrees around its center, coloring the expanded area with the pointer's average color. Large pointer templates are generated by rotating this enlarged pointer n degrees for 360 mod n times. We opt for $n = 5$ degrees to strike a balance between coverage and precision.

Employing the large pointer templates, we first identify the approximate range of the pointer. We then refine our search within this 5-degree window using the small pointer templates. This method narrows the search to 77 iterations $(360/5 + 5)$, marking a significant acceleration compared to the original approach.

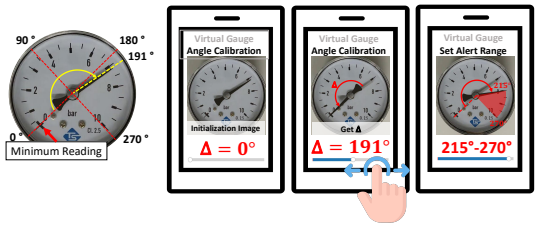


Fig. 6: Left: Analog gauge pointer angle reading coordinate system. **Right:** User interfaces. After the template generation phase, a virtual gauge will be shown on the user interface, e.g., the user's mobile phone. The user can slide the slider to calibrate the angle (middle two images). The user can also use the interface to set the alert range. For example, the user sets the alert range to be 8 to 10 bars, where the corresponding angle range is from 215° to 270°.

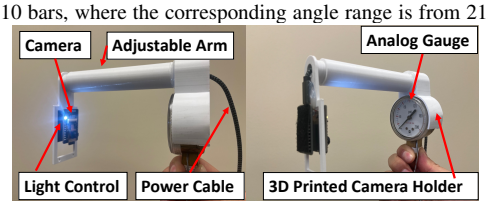


Fig. 7: Analog Gauge Reading System Prototype

D. Post Processing

Using the interface depicted in Fig. 6, users can set multiple alert thresholds. By adjusting the pointer on the virtual gauge - a digital twin of the physical gauge, they observe real-time readings while the backend monitors corresponding pointer angles. Alerts can be configured for readings that exceed, fall below, or fall within specific ranges. The backend logs these angles, updating the alert system's rules and values. Our system's continuous monitoring allows for the analysis of time series data with anomaly detection algorithms, identifying unusual patterns.

IV. Implementation

A. Hardware Implementation

Our testbed as shown in Fig. 7 evaluates our method under real-world conditions, featuring a 3D-printed camera holder designed for non-intrusive gauge monitoring. This holder includes an adjustable arm for optimal camera positioning and a flashlight to enable operation in dark environments. The system is powered through a standard power outlet which is abundant in the indoor work environment. With modifications, the analog gauge's value is adjustable via a manual handle for the testing purpose. The total cost, excluding the gauge, is under $10, supporting large-scale deployment.

B. Software Implementation

We deploy and serve SAM, CogVLM, and LaMa models on a server equipped with one RTX4090 GPU (24GB RAM). We use the MQTT protocol for data exchange between camera modules and the server, including uploading initialization images and transmitting templates and angle readings. We implement our method on the AI-Thinker ESP32 CAM, an off-the-shelf IoT camera with a 240 MHz CPU, 4 MB of PSRAM,

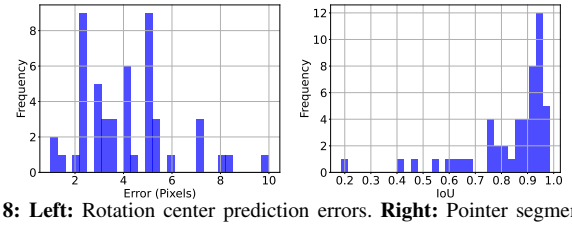


Fig. 8: Left: Rotation center prediction errors. **Right:** Pointer segmentation IoU results.

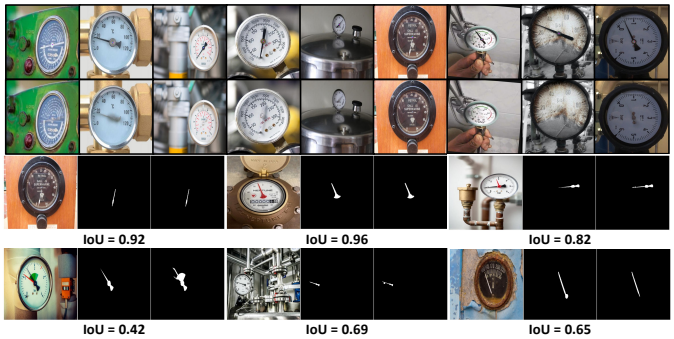


Fig. 9: Top Row: Examples of rotation center detection results; for optimal viewing, please use digital zoom features. The cyan dot indicates the predicted center and the red dot represents the manually labeled center. **Second Row:** Examples of pointer removal results. **Bottom Rows:** Examples of pointer segmentation results. Within each image group, the sequence from left to right is as follows: raw image, manually labeled mask, and predicted mask.

and 2 MB of flash memory (setup used in our application). To overcome storage and computational limitations, we reduce template resolution to 40x40 pixels, storing them in JPEG format to minimize space. The templates occupy approximately 400 KB of storage in total. JPEG images are converted to raw RGB 888 format for processing, taking up about 2 MB of PSRAM. Converting a JPEG image (e.g., 640x480) to RGB 888 format, cropping, and resizing it to 40x40 pixels on a microcontroller takes about 1.5 seconds. To optimize this process, we use the windowing and scaling features of the OV2640 image sensor to directly output the region of interest at the desired size, eliminating the need for separate cropping and resizing steps and saving time.

V. Evaluation

In this section, we conduct a comprehensive evaluation of our system, focusing on the generalizability of our template generation pipeline, the accuracy of our gauge reading algorithm, and the performance characteristics of our prototype in real-world application scenarios.

A. Template generation results

The most critical step in our methodology involves the precise construction of templates, necessitating accurate detection of the pointer's rotation center, segmentation of the pointer, and pointer removal from the background. To assess the robustness of our template generation pipeline across a variety of analog gauges, we compile a dataset of 50 images. These images are selected from the literature [1], [5], [38] and the Internet, chosen specifically for their complex backgrounds and varied appearances. We manually label the center of each gauge and manually segment each pointer. Examples of dataset samples are depicted in Fig. 9.

TABLE I: Comparison of methods on analog reading. Results are reported as the mean absolute angle reading error, with units in degrees. **CTF** is short for coarse-to-fine.

Dataset	Ours		Howells et al. [1]	Leon et al. [2] (Large)
	Normal	CTF		
meter_a	**0.55**	0.78	9.24	2.50
meter_b	**1.19**	1.07	26.06	8.29
meter_c	1.29	**1.10**	1.94	1.97
meter_d	**0.95**	1.02	11.70	7.36
meter_e	3.18	**3.04**	3.70	4.83
meter_f	**3.71**	4.06	4.31	4.69
Average ↓	**1.81**	1.85	9.49	4.94

For the task of gauge center detection, we calculate the Euclidean distance in pixels between the coordinates of the predicted and labeled centers. In the task of gauge pointer segmentation, we measure the Intersection over Union (IoU) between the predicted and the ground truth masks. Results and failure cases are presented in Figs. 8 and 9, respectively. These results indicate the effectiveness and generalizability of our methods, with averaged center coordinate prediction errors of less than 4 pixels and an average pointer mask IoU of 0.84. After reviewing samples with notable errors in predicting the rotation center, we observe that these samples are of higher resolution. Consequently, the error, measured in pixels, appears larger. However, visually, the predicted center is very close to the actual ground truth.

We identify three types of imperfect pointer segmentation: The first is a fatal failure, illustrated in the bottom left example of Fig. 9, where the pointer blends in with the background, leading to erroneous segmentation. Such cases are infrequent, as gauges typically feature pointers in high contrast to the gauge plate. Even in the event of a fatal failure, users can easily correct the segmentation via the user interface. The second type involves the pointer tip overlapping with gauge text or markers as shown in the bottom middle example of Fig. 9, resulting in partial segmentation of these overlapped objects. Our experimental findings suggest that this type of error does not greatly affect the accuracy of gauge pointer angle readings. The third type occurs when the rotation center is not segmented correctly as illustrated in the last example in Fig. 9, an error deemed inconsequential since our pipeline ultimately excludes the rotation center from the segmented pointer. Overall, our automated pipeline for rotation center detection and pointer segmentation proves to be adaptable to gauges of diverse appearances.

Pointer removal is a comparatively simpler task, demonstrating good average performance across all samples; minor imperfections in the cleaned gauge face do not affect prediction accuracy. Qualitative results are showcased in the second row of Fig. 9.

B. End-to-end reading results

We evaluate our method using the dataset from [1], which includes 450 frames for each of six distinct gauges, as shown in Fig. 10. Following [2]'s suggestion, we correct some inaccuracies in the provided ground truths.

We compare our normal and coarse-to-fine methods (Sec. III-B and Sec. III-C, respectively) against two advanced, deep learning-based approaches [1] and [2] on the gauge reading

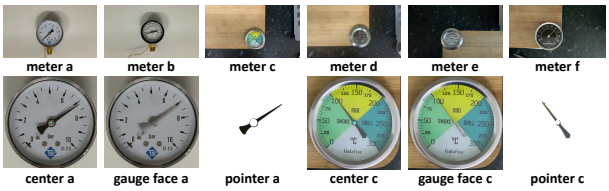


Fig. 10: Top: Examples of images of each analog gauge in the analog gauge reading dataset. **Bottom**: Generated rotation centers, clean gauge faces, and segmented pointers of meters a and c.

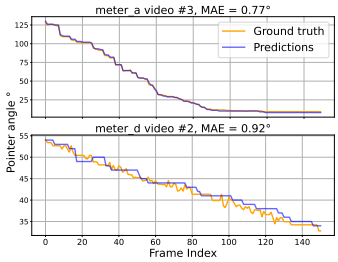


Fig. 11: Angle reading result examples of our **normal** method.

task. Our method requires the camera to be fixed to the gauge, whereas in the meter_f videos, the gauge moves. To adapt our system for the meter_f datasets, we developed a script that aligns the rotation center in each frame, thereby stabilizing the video. The results, shown in Table I as the mean absolute error (MAE) in angle readings (degrees), demonstrate that our methods greatly outperform the current state-of-the-art. Our coarse-to-fine method achieves a significant speed improvement with a minimal accuracy trade-off, making it the preferred choice for real-world applications. Fig. 11 illustrates two examples where our predictions closely match the ground truth, contrasting with the significant discrepancies observed with the method from [2], sometimes exceeding 100 degrees. Our comparison, based on visual inspection against their supplementary materials, highlights our method's accuracy and reliability. Despite requiring an angle calibration step by users, this process is quick (under 5 seconds) and easy with our interface, yielding a mean calibration error of 1.33°.

C. Real-world deployment results

We implement our system on a low-cost camera, as detailed in Sec. IV-A, to evaluate our method's performance in real-world scenarios from the following aspects:

Bandwidth: All image processing tasks are performed directly on the IoT camera, which then sends only the angle results to the server, significantly reducing bandwidth usage. In contrast, methods that require the camera to offload tasks to the server by sending images at high resolutions greatly increase bandwidth needs. This is particularly true for continuous tracking at rates such as 5 FPS and when multiple cameras are installed in proximity.

Process Latency: Our testbed achieves a maximum reading rate of 7.5 readings per second. Further latency reductions are possible since the program isn't fully optimized. Our method is flexible, with the potential for performance enhancements via advanced hardware and camera technologies. In contrast, the neural network approach [6] and Lai et al.'s traditional

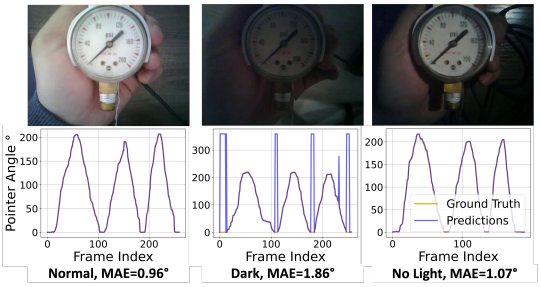


Fig. 12: Gauge reading results in different environment conditions.[1]

algorithm [39] record processing times over 1.5 seconds per frame on the more powerful, expensive RK3399 hardware.

Deployment time: Our system installs quickly on any gauge with minimal setup. Users attach the camera holder using adhesive and power on the camera. Initialization takes only one minute: 20 seconds for server template generation and 40 seconds for template transmission via Wi-Fi. After setup, angle readings are streamed to the user interface instantly.

Overall Cost: Each setup is less than **$10**. The server used to make templates can be shared in large organizations, spreading out the cost.

Robustness to environment changes: We evaluate the resilience of our methodology under various environmental conditions: normal lighting, low light, and complete darkness. In situations of no light, we use a handheld iPhone flashlight to simulate an external light source, illuminating the gauge from the side[2]. In each setting, we adjusted the gauge's pointer from its minimum to maximum values and back, repeating this process three times at a consistent pace. Videos were captured at 5 FPS. Templates are generated under normal lighting conditions and reused in two other conditions. We label each video frame by frame to get the angle ground truth. Fig. 12 illustrates sample images in each environment alongside the angle reading results. Our methodology demonstrated robust performance in all tested conditions. Notably, in low light, the method inaccurately estimated the pointer's angle due to its blurred appearance, suggesting the pointer had moved in the opposite direction. To mitigate this, filtering techniques could be employed, or an assumption could be made that the pointer does not shift beyond ± 160 degrees between two consecutive frames, allowing for the exclusion of erroneous readings.

VI. Conclusion

In this work, we introduce GaugeTracker, a novel system for reading analog gauges that combines innovative template-matching algorithms with a large vision-language model-aided template generation pipeline. Our system operates entirely on the world's most affordable IoT camera, offering accurate, real-time, continuous gauge readings and demonstrating its scalability and practicality.

[1]Spikes in the result of dark environment occur with predictions around the 0° to 359° transition, appearing as large errors due to our coordinate system, yet actual errors are only 1-2 degrees. These are considered in our MAE calculation.

[2]We avoid using the camera's LED because it acts as a spotlight, causing significant reflection on the gauge. Future work will explore lighting adjustments, like filters or repositioning, through hardware modifications.

REFERENCES

[1] B. Howells, J. Charles, and R. Cipolla, "Real-time analogue gauge transcription on mobile phone," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2369–2377, 2021.

[2] J. Leon-Alcazar, Y. Alnumay, C. Zheng, H. Trigui, S. Patel, and B. Ghanem, "Learning to read analog gauges from synthetic data," *arXiv preprint arXiv:2308.14583*, 2023.

[3] M. K. Gellaboina, G. Swaminathan, and V. Venkoparao, "Analog dial gauge reader for handheld devices," in *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1147–1150, IEEE, 2013.

[4] J. Peixoto, J. Sousa, R. Carvalho, G. Santos, J. Mendes, R. Cardoso, and A. Reis, "Development of an analog gauge reading solution based on computer vision and deep learning for an iot application," in *Telecom*, vol. 3, pp. 564–580, MDPI, 2022.

[5] J. S. Lauridsen, J. A. Graasmé, M. Pedersen, D. G. Jensen, S. H. Andersen, and T. B. Moeslund, "Reading circular analogue gauges using digital image processing," in *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Visigrapp 2019)*, pp. 373–382, SCITEPRESS Digital Library, 2019.

[6] Y. Lin, Q. Zhong, and H. Sun, "A pointer type instrument intelligent reading system design based on convolutional neural networks," *Frontiers in Physics*, vol. 8, p. 618917, 2020.

[7] C.-H. Wang, K.-K. Huang, R.-I. Chang, and C.-K. Huang, "Scale-mark-based gauge reading for gauge sensors in real environments with light and perspective distortions," *Sensors*, vol. 22, no. 19, p. 7490, 2022.

[8] J. Wang, J. Huang, and R. Cheng, "Automatic reading system for analog instruments based on computer vision and inspection robot for power plant," in *2018 10th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 1–6, IEEE, 2018.

[9] C. Zheng, S. Wang, Y. Zhang, P. Zhang, and Y. Zhao, "A robust and automatic recognition system of analog instruments in power system by using computer vision," *Measurement*, vol. 92, pp. 413–420, 2016.

[10] Y. Ma and Q. Jiang, "A robust and high-precision automatic reading algorithm of pointer meters based on machine vision," *Measurement Science and Technology*, vol. 30, no. 1, p. 015401, 2018.

[11] M. Yi, Z. Yang, F. Guo, and J. Liu, "A clustering-based algorithm for automatic detection of automobile dashboard," in *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 3259–3264, IEEE, 2017.

[12] X. Mai, W. Li, Y. Huang, and Y. Yang, "An automatic meter reading method based on one-dimensional measuring curve mapping," in *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, pp. 69–73, IEEE, 2018.

[13] X. Li, P. Yin, C. Duan, and Y. Zhi, "Analog gauge reader based on image recognition," in *Journal of Physics: Conference Series*, vol. 1650, p. 032061, IOP Publishing, 2020.

[14] T. Selvathai, S. Ramesh, K. Radhakrishnan, *et al.*, "Automatic interpretation of analog dials in driver's instrumentation panel," in *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 411–415, IEEE, 2017.

[15] R. Sablatnig and W. G. Kropatsch, "Automatic reading of analog display instruments," in *Proceedings of 12th international conference on pattern recognition*, vol. 1, pp. 794–797, IEEE, 1994.

[16] H. Bao, Q. Tan, S. Liu, and J. Miao, "Computer vision measurement of pointer meter readings based on inverse perspective mapping," *Applied Sciences*, vol. 9, no. 18, p. 3729, 2019.

[17] Y.-S. Chen and J.-Y. Wang, "Computer vision-based approach for reading analog multimeter," *Applied Sciences*, vol. 8, no. 8, p. 1268, 2018.

[18] S. Chavan, X. Yu, and J. Saniie, "High precision analog gauge reader using optical flow and computer vision," in *2022 IEEE International Conference on Electro Information Technology (eIT)*, pp. 171–175, IEEE, 2022.

[19] J. Chi, L. Liu, J. Liu, Z. Jiang, G. Zhang, *et al.*, "Machine vision based automatic detection method of indicating values of a pointer gauge," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[20] J. Peixoto, J. Sousa, R. Carvalho, G. Santos, R. Cardoso, and A. Reis, "End-to-end solution for analog gauge monitoring using computer vision in an iot platform," *Sensors*, vol. 23, no. 24, p. 9858, 2023.

[21] S. Dumberger, R. Edlinger, and R. Froschauer, "Autonomous real-time gauge reading in an industrial environment," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 1281–1284, IEEE, 2020.

[22] Y. Liu, J. Liu, and Y. Ke, "A detection and recognition system of pointer meters in substations based on computer vision," *Measurement*, vol. 152, p. 107333, 2020.

[23] Z. Li, Y. Zhou, Q. Sheng, K. Chen, and J. Huang, "A high-robust automatic reading algorithm of pointer meters based on text detection," *Sensors*, vol. 20, no. 20, p. 5946, 2020.

[24] A. Alexeev, G. Kukharev, Y. Matveev, and A. Matveev, "A highly efficient neural network solution for automated detection of pointer meters with different analog scales operating in different conditions," *Mathematics*, vol. 8, no. 7, p. 1104, 2020.

[25] W. Cai, B. Ma, L. Zhang, and Y. Han, "A pointer meter recognition method based on virtual sample generation technology," *Measurement*, vol. 163, p. 107962, 2020.

[26] J. Huang, J. Wang, Y. Tan, D. Wu, and Y. Cao, "An automatic analog instrument reading system using computer vision and inspection robot," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6322–6335, 2020.

[27] H. Nagarajan, R. S. A. Podila, C. Vishal, D. Radha, J. Amudha, G. P. Murthy, and A. Rajendran, "Automated analogue dial reading in cockpits," in *2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pp. 1–6, IEEE, 2022.

[28] J. Barbosa, R. Graça, G. Santos, and M. J. M. Vasconcelos, "Automatic analogue gauge reading using smartphones for industrial scenarios," in *Proceedings of the 2023 8th International Conference on Machine Learning Technologies*, pp. 277–284, 2023.

[29] G. Salomon, R. Laroca, and D. Menotti, "Deep learning for image-based automatic dial meter reading: Dataset and baselines," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2020.

[30] V. Trairattanapa, S. Phimsiri, C. Utintu, R. Cherdchusakulcha, T. Tosawadi, E. Thamwiwatthana, S. Tungjitnob, P. Tangamonsiri, A. Takutruea, A. Keomeesuan, *et al.*, "Real-time multiple analog gauges reader for an autonomous robot application," in *2022 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, pp. 1–6, IEEE, 2022.

[31] J. Gallagher, "Turn analog dials into digital data with computer vision." https://blog.roboflow.com/read-analog-dials-computer-vision/, 2022. Accessed: 03-06-2024.

[32] Zalmotek, "Analog meter reading - arduino nicla vision." https://docs.edgeimpulse.com/experts/image-projects/analog-meter-reading-arduino-nicla-vision, 2023. Accessed: 03-06-2024.

[33] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song, J. Xu, B. Xu, J. Li, Y. Dong, M. Ding, and J. Tang, "Cogvlm: Visual expert for pretrained language models," 2023.

[34] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[35] H. Zheng, C. Zhang, K. Guan, Y. Deng, S. Wang, B. L. Rhoads, A. J. Margenot, S. Zhou, and S. Wang, "Segment any stream: Scalable water extent detection with the segment anything model," in *NeurIPS 2023 Computational Sustainability: Promises and Pitfalls from Theory to Deployment*, 2023.

[36] K. Zhang and D. Liu, "Customized segment anything model for medical image segmentation," *arXiv preprint arXiv:2304.13785*, 2023.

[37] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2149–2159, 2022.

[38] Gauges, "gauges object detection dataset." https://universe.roboflow.com/gauges-e7hmd/gauges_object_detection, aug 2023. visited on 2024-03-07.

[39] H. Lai, Q. Kang, L. Pan, and C. Cui, "A novel scale recognition method for pointer meters adapted to different types and shapes," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 374–379, IEEE, 2019.