

Open CV-Based Analogue Gauge Detection and Digitization: A Sturdy, Computationally Effective Framework for Industrial Retrofitting

Sadiya Patel

Department of Computer Science and Engineering, Anjuman College of Engineering and Technology, Nagpur, India

Prof. Manish K Assudani

Department of Computer Science and Engineering, Anjuman College of Engineering and Technology, Nagpur, India

Dr. Muzaffar Khan

Department of Electronics & Telecom Engineering, Anjuman College of Engineering and Technology, Nagpur, India

Abstract

Despite the rapid proliferation of Industry 4.0 and the Industrial Internet of Things (IIoT), analogue gauges remain deeply entrenched in critical industrial infrastructure due to their inherent robustness, passive operation, and zero electrical dependency. Their continued use, however, creates significant data silos, impeding real-time monitoring, predictive maintenance, and operational efficiency. This paper presents a comprehensive computer vision framework for the automated digitization of circular analogue gauges using the OpenCV library. The proposed pipeline employs the Hough Circle Transform (HCT) for gauge localization and the Probabilistic Hough Line Transform (PHLT) for pointer segmentation, augmented by adaptive image preprocessing and angular interpolation for value computation. Unlike data-hungry deep learning architectures requiring specialized GPU hardware, the proposed classical approach employs a semi-automated human-in-the-loop calibration strategy that ensures generalizability across diverse gauge types and configurations. Experimental evaluation demonstrates real-time performance at approximately 28 frames per second on a Raspberry Pi 4 Model B (4 GB RAM) with a mean reading error between 0.1% and 0.5%. The system achieves competitive accuracy relative to state-of-the-art deep learning methods while operating at a fraction of the computational and financial cost, making it a viable solution for retrofitting legacy industrial facilities.

Index Terms — Computer Vision, Industrial Automation, OpenCV, Hough Transform, IIoT, Edge Computing, Analogue Gauge Reading, Image Processing, Raspberry Pi, Predictive Maintenance.

I. INTRODUCTION

A. The Persistent Role of Analogue Instrumentation

In the contemporary landscape of industrial automation, characterized by the accelerating adoption of the Industrial Internet of Things (IIoT) and cyber-physical systems, a technologically paradoxical reality persists across global manufacturing, energy, and chemical processing sectors. Despite the widespread availability of digital transducers, smart sensors, and fieldbus-integrated instrumentation, analogue gauges continue to serve as the primary measurement instruments for monitoring critical process variables including pressure, temperature, fluid flow rate, and rotational speed [1][2].

This reliance is neither coincidental nor an oversight of industrial engineers. Analogue gauges, many of which trace their mechanical lineage to the Bourdon tube pressure gauge invented in the 19th century, offer a fundamentally unique set of operational characteristics: they function passively without requiring electrical power, they are intrinsically safe

in explosive or volatile atmospheres classified as ATEX Zone 0 or Zone 1 environments, and they exhibit a long operational lifespan with predictable failure modes [3]. In hazardous process plants, offshore oil rigs, nuclear power facilities, and chemical production units, these characteristics are not merely desirable—they are often mandated by safety regulations and industrial standards.

Moreover, the capital investment already embedded in installed analogue instrumentation is enormous. A large petrochemical refinery may have thousands of pressure gauges, temperature dials, and flowmeter indicators distributed across its piping infrastructure. Replacing all such devices with digital alternatives—the so-called "rip-and-replace" strategy—entails prohibitive capital expenditure, extended plant shutdown periods, potential safety recertification, and significant process disruption [3][7]. For small and medium-sized enterprises (SMEs) operating legacy facilities, such an upgrade path is economically unfeasible.

B. The Imperative for Automated Digitization

The operational coexistence of mechanical analogue gauges and digital automation systems creates what is aptly described as a "data silo" problem. Gauge readings are observable exclusively to human operators who must conduct manual inspection rounds at periodic intervals—typically every two to eight hours depending on the process criticality. This manual paradigm introduces a cascade of operational inefficiencies and safety vulnerabilities [4][5].

The "Parallax Effect," arising from angular misalignment between the observer's line of sight and the gauge pointer, introduces systematic reading errors that can reach 1–2 full scale divisions on standard industrial gauges [5][6]. Operator fatigue, particularly during night shifts or in extreme environmental conditions (heat, noise, confined spaces), further degrades measurement reliability. Transcription errors during manual logging—where operators record readings on paper forms that are later digitally entered—compound the data quality problem.

Beyond data quality, the absence of continuous, real-time gauge monitoring fundamentally undermines modern predictive maintenance (PdM) frameworks. Undetected pressure spikes, gradual thermal drift, or progressive flow restriction—each detectable through continuous time-series analysis of gauge readings—may remain unobserved between manual inspection cycles, leading to catastrophic equipment failure, environmental incidents, or personnel injury. The economic consequences of unplanned downtime in process industries can amount to hundreds of thousands of dollars per hour.

C. The Industrial IoT and Digitalization Challenge

The Industry 4.0 framework envisions seamless data integration across the entire manufacturing value chain, from field instruments to enterprise resource planning (ERP) systems, enabling closed-loop control, predictive analytics, and autonomous decision-making. However, the digital transformation of legacy facilities is constrained by the analogue instrumentation problem. Computer vision (CV) emerges as a non-invasive, cost-effective solution that treats the analogue gauge face as a visual communication channel—decoding the information encoded in the angular position of the pointer without any physical modification to the gauge itself [3][7].

CV-based gauge reading systems can be integrated into existing plant infrastructure using commercially available cameras mounted on fixed brackets adjacent to gauge panels. The digitized readings can be transmitted via IIoT protocols (MQTT, OPC-UA, Modbus/TCP) to SCADA systems, cloud platforms, or local edge computing nodes, effectively bridging the gap between mechanical instrumentation and the digital industrial ecosystem [9][10].

D. Scope and Objectives

This research presents the design, implementation, and experimental validation of a robust CV system for analogue gauge digitization using the OpenCV library. The principal objectives are:

- 1) To develop a modular, real-time image processing pipeline employing geometric feature extraction based on Hough Transforms for reliable gauge localization and pointer angular measurement.
- 2) To implement a semi-automated calibration methodology that eliminates the dependency on large labeled training datasets while ensuring adaptability across diverse gauge scales, ranges, and configurations.
- 3) To optimize the pipeline for deployment on low-cost, constrained edge computing hardware, specifically the Raspberry Pi 4 Model B, demonstrating its viability for economical industrial retrofitting.
- 4) To provide a rigorous quantitative comparison of the proposed approach against state-of-the-art classical and deep learning-based gauge reading methods.

II. RELATED WORK

A. Classical Computer Vision Approaches

Classical CV-based analogue gauge reading (AGR) methods are grounded in deterministic geometric modeling, exploiting the well-defined circular geometry of gauge bezels and the elongated linear geometry of gauge pointers. The foundational technique employs the Hough Circle Transform (HCT) to identify the gauge bezel boundary and the Hough Line Transform (HLT) or its probabilistic variant (PHLT) to extract the pointer orientation [11].

Wang et al. [2] introduced a Scale-Mark-Based Gauge Reading (SGR) system that applies Principal Component Analysis (PCA) to the set of detected scale marks to robustly determine the gauge center and zero angle. Their system achieved a mean error below 0.5% even under challenging conditions involving perspective distortion and variable illumination, demonstrating the effectiveness of geometric analysis of scale marks as supplementary structural cues. This approach significantly enhances robustness compared to methods relying solely on pointer segmentation.

Chen and Wang [12] applied classical CV techniques to the domain of analogue multimeter reading, detecting the function selector switch state and pointer angle through edge-based segmentation. Their work illustrated the generalizability of Hough-based methods to non-standard gauge formats, including multi-scale instruments with overlapping pointer ranges.

Milana et al. [5] (GAUREAD) developed an autonomous system targeting unstructured environments where gauges may appear at arbitrary orientations and distances. Their pipeline incorporated perspective correction prior to gauge localization, addressing a key limitation of fixed-camera classical methods. Similarly, Tran et al. [6] applied CV techniques in power substation environments, achieving satisfactory results under the controlled but often harsh lighting conditions typical of electrical switchgear installations.

B. Deep Learning Approaches

The advent of deep convolutional neural networks (CNNs) has fundamentally transformed the capability profile of gauge reading systems, enabling end-to-end learning that bypasses the need for explicit geometric modeling.

Object Detection Architectures: Families of object detectors based on the YOLO (You Only Look Once) architecture have been applied to gauge localization tasks in unstructured environments [5][6]. YOLO-based detectors offer real-time inference with competitive detection accuracy, enabling simultaneous localization of multiple gauges within a single camera field of view. However, their performance is contingent on the availability of large, carefully annotated training datasets.

Semantic Segmentation Models: Yang et al. [7] proposed a lightweight segmentation architecture coupling a MobileNetV2 backbone with the DeepLabv3+ decoder for automatic reading of analogue dial gauges in outdoor electrical substation environments. This system handled varying measurement ranges and degraded visual conditions caused by weathering and solar glare. The lightweight backbone was selected to balance accuracy with deployment feasibility on embedded processors.

End-to-End Regression: Reitsma et al. [1] presented a learning-based framework for pressure gauge reading that directly regresses the gauge reading from the input image without requiring explicit knowledge of the scale layout or pointer geometry. Evaluated on a real-world dataset collected from field deployments, their method achieved a relative reading error of approximately 2%, representing the performance achievable without the benefit of gauge-specific calibration. Jelidi et al. [4] extended this paradigm to vehicle dashboard validation, benchmarking multiple vision-based methods under automotive testing conditions.

IoT-Integrated Learning Systems: Peixoto et al. [3][10] developed an end-to-end IIoT solution combining CV-based gauge reading with cloud connectivity, demonstrating continuous remote monitoring over extended periods. Their pipeline integrated deep learning-based gauge segmentation with MQTT-based telemetry and cloud dashboarding, illustrating the complete architecture required for a production-grade IIoT gauge monitoring deployment. Ninama et al. [14] demonstrated the application of deep transfer learning to circumvent the need for large domain-specific datasets by leveraging pre-trained convolutional feature extractors.

C. Hybrid and Edge-Optimized Approaches

GaugeTracker [9] presented an AI-powered cost-effective monitoring system that combines classical localization with learned pointer segmentation, targeting deployment on resource-constrained hardware. This hybrid strategy—combining the efficiency of geometric methods with the adaptability of learned components—represents an emerging design philosophy in practical AGR systems.

Edge deployment considerations are increasingly central to AGR research. Systems targeting Raspberry Pi, NVIDIA Jetson Nano, or ESP32-class hardware must carefully balance model complexity against inference latency and power consumption. While Jetson Nano enables GPU-accelerated deep learning at the edge, its cost and power profile are prohibitive for large-scale industrial deployments requiring hundreds of monitoring nodes [7][14].

D. The Research Gap

A fundamental trilemma characterizes AGR system design: achieving simultaneously high Robustness, computational Efficiency, and Data Independence. Deep learning methods [1][7] deliver superior robustness under adverse conditions but demand large labeled training datasets and GPU-capable hardware. Classical geometric methods [2][12] are computationally efficient but can be brittle under visual noise, variable illumination, and non-standard gauge configurations. IoT-integrated solutions [3][10] address connectivity requirements but often add system complexity and infrastructure dependency.

This work targets the intersection of efficiency and data independence by proposing a calibrated classical pipeline that achieves accuracy competitive with uncalibrated deep learning approaches at a fraction of the computational cost. It is positioned specifically for the large and underserved market segment of SMEs and legacy facilities where dataset collection and GPU deployment are impractical.

III. SYSTEM ARCHITECTURE

A. Overall Pipeline Design

The proposed system operates as a sequential multi-stage image processing pipeline. As illustrated conceptually in Fig. 1, the pipeline comprises five primary stages: (1) image acquisition and preprocessing, (2) gauge localization, (3) pointer segmentation and filtering, (4) angular measurement, and (5) value interpolation and output. Each stage is implemented as an independent, modular processing unit, enabling individual optimization and replacement without disrupting adjacent stages.

The pipeline is designed for both online (streaming video) and offline (static image) operation modes. In streaming mode, each video frame undergoes the complete processing chain independently, with stateful operations such as pointer angle temporal smoothing optionally applied across frames to reduce measurement jitter.

B. Hardware Configuration

The target deployment hardware is the Raspberry Pi 4 Model B equipped with 4 GB of LPDDR4 SDRAM, a quad-core ARM Cortex-A72 processor running at 1.5 GHz, and operating under the Raspberry Pi OS (64-bit, Debian Bullseye). The imaging subsystem employs the Raspberry Pi Camera Module v2 (Sony IMX219 sensor), configured to capture video at 640×480 pixel resolution at 30 frames per

second over the CSI-2 camera interface. This resolution was selected to balance spatial detail sufficient for pointer angular discrimination with the frame processing rate achievable on the target processor.

For validation purposes, testing was additionally conducted on a standard laptop PC (Intel Core i5-10th Gen, 8 GB RAM) to provide a performance baseline independent of the ARM platform. The complete bill of materials for the edge monitoring unit costs approximately USD 85, comprising the Raspberry Pi board, camera module, power supply, and enclosure.

C. Software Stack

The software implementation utilizes Python 3.9 with OpenCV 4.7.0 as the core image processing library, NumPy 1.23 for numerical computation, and the Python picamera2 library for camera interface control. For IIoT connectivity, the paho-mqtt library enables MQTT-based telemetry transmission to upstream data collection services. The system operates entirely offline—no cloud connectivity, proprietary APIs, or subscription services are required for core gauge reading functionality.

IV. METHODOLOGY

A. Image Acquisition and Pre-processing

The raw video frame $I(x, y)$ captured at 640×480 resolution first undergoes colorspace conversion from the BGR format native to OpenCV to grayscale, reducing the three-channel input to a single-channel luminance image $G(x, y)$. This dimensionality reduction accelerates all subsequent processing operations and eliminates sensitivity to gauge color and lighting color temperature:

$$G(x, y) = 0.299 \cdot R(x, y) + 0.587 \cdot B(x, y) + 0.114 \cdot B(x, y)$$

Gaussian spatial filtering is applied to suppress high-frequency sensor noise, which is particularly prevalent in industrial environments with vibration and electromagnetic interference. The smoothed image is computed by convolution with the Gaussian kernel:

$$G(x, y) = (1 / 2\pi\sigma^2) \cdot \exp(-(x^2 + y^2) / 2\sigma^2)$$

A kernel size of 5×5 with $\sigma = 1.5$ is empirically determined to provide adequate noise suppression while preserving edge sharpness critical for the subsequent Canny edge detection stage. Contrast Limited Adaptive Histogram Equalization (CLAHE) is then applied with a clip limit of 2.0 and a tile grid size of 8×8 pixels to enhance local contrast. CLAHE is preferred over global histogram equalization because it corrects for spatially non-uniform illumination—such as shadows cast by the gauge housing or specular reflections from the gauge glass—without amplifying noise in already well-illuminated regions [8].

B. Gauge Localization via Hough Circle Transform

The preprocessed grayscale image is subjected to the Hough Circle Transform (HCT) to identify the circular gauge bezel. The HCT operates by accumulating votes in a three-dimensional parameter space (a, b, r) representing candidate circle centers and radii. For each edge pixel (x, y) identified by the Sobel gradient operator, all circles passing through that pixel cast votes along the corresponding arc in the accumulator space [11].

To reliably detect the gauge bezel while rejecting spurious circles arising from concentric features such as the pointer hub, scale tick marks, or dial graphics, the following HCT parameters are applied: minimum inter-circle distance (minDist) set to 80% of the image minimum dimension; accumulator threshold (param2) tuned conservatively to 40; and radius range constrained to $[60, 220]$ pixels, corresponding to gauges occupying 20–70% of the image width. The detected circle with the highest accumulator score is designated as the gauge bezel, and a square Region of Interest (ROI) circumscribing the detected circle is extracted for subsequent processing.

C. Pointer Segmentation and Filtering

Within the extracted ROI, Canny edge detection is applied with a dual hysteresis threshold ($T_{low} = 50$, $T_{high} = 150$) to extract the structural edge map. The Probabilistic Hough Line Transform (PHLT), which operates on the binary edge map and returns finite line segments rather than infinite lines, is then applied to detect candidate line segments. PHLT parameters are set as: ρ resolution = 1 pixel, θ resolution = $\pi/180$ radians, minimum vote threshold = 30, minimum segment length = $r/3$, and maximum gap = 10 pixels, where r is the detected gauge radius.

Raw PHLT output typically contains dozens to hundreds of detected segments including scale tick marks, digit strokes, and housing boundaries. A three-stage filtering logic isolates the gauge pointer:

- 1) Length Filter: Segments shorter than $r/3$ are discarded, as tick marks and digit strokes are substantially shorter than the pointer.
- 2) Centrality Constraint: Segments whose nearest point to the gauge center (a, b) exceeds a tolerance $\delta = r/10$ are rejected, since the pointer necessarily passes through or near the dial center.
- 3) Collinearity Clustering: Surviving segments are grouped by angular proximity (within 5°) and spatially merged into a single composite pointer vector by computing the weighted average angle, with weights proportional to segment length.

D. Angular Measurement and Vector Analysis

The composite pointer vector is characterized by its orientation angle θ_{curr} , computed as the arc-tangent of the endpoint displacement vector projected onto the gauge coordinate system. The raw OpenCV angle convention (measured from the positive x-axis, clockwise) is

normalized to the standard clockwise-from-12-o'clock convention used by analogue gauges:

$$\theta_{\text{norm}} = (90^\circ - \theta_{\text{raw}}) \bmod 360^\circ$$

Temporal smoothing is optionally applied across successive frames using an Exponential Moving Average (EMA) filter with smoothing factor $\alpha = 0.3$, which suppresses frame-to-frame angular jitter caused by micro-vibrations or image noise while maintaining adequate tracking responsiveness to genuine pointer movement:

$$\theta_{\text{smooth}}[t] = \alpha \cdot \theta_{\text{curr}}[t] + (1 - \alpha) \cdot \theta_{\text{smooth}}[t-1]$$

E. Semi-Automated Calibration Protocol

The proposed calibration methodology replaces the conventional approach of optical character recognition (OCR) of scale markings—which is unreliable on aged, dirty, or non-standard gauges—with a structured human-in-the-loop interaction. During an initial one-time calibration session for each gauge type, the operator manually positions the gauge pointer at its minimum scale value V_{min} and records the corresponding pointer angle θ_{min} by pressing a keyboard key. The procedure is repeated for the maximum scale value V_{max} to record θ_{max} . The physical minimum and maximum values (V_{min} , V_{max}) and their units are entered via a command-line prompt.

The calibration parameters $\{\theta_{\text{min}}, \theta_{\text{max}}, V_{\text{min}}, V_{\text{max}}\}$ are serialized to a JSON configuration file indexed by gauge identifier. During subsequent runtime operation, the real-time gauge reading is computed by linear interpolation:

$$V = V_{\text{min}} + [(\theta_{\text{curr}} - \theta_{\text{min}}) \times (V_{\text{max}} - V_{\text{min}})] / (\theta_{\text{max}} - \theta_{\text{min}})$$

This formulation assumes a linear relationship between pointer angle and measured value, which holds for all standard analogue gauges operating within their normal measurement range. Non-linear scale gauges (e.g., logarithmic or power-law scales) can be accommodated by extending the calibration to include intermediate reference points and applying piecewise linear interpolation across calibrated segments. The calibration process for a single gauge type requires less than two minutes and does not require any disassembly or modification of the gauge.

F. Output and Integration

The computed gauge reading V is formatted with units and transmitted via three concurrent output channels: (1) real-time overlay rendered on the video frame displaying the detected circle, pointer line, and reading value for operator verification; (2) timestamped CSV logging to local storage for offline analysis; and (3) MQTT telemetry publication to a configurable broker endpoint for IIoT integration. The system supports simultaneous monitoring of multiple gauges by instantiating independent pipeline instances, one

per camera, with synchronized MQTT publishing under distinct topic hierarchies.

V. IMPLEMENTATION DETAILS

A. Algorithmic Description

Algorithm 1 summarizes the complete per-frame processing procedure of the proposed gauge reading pipeline:

Algorithm 1: Per-Frame Gauge Reading Pipeline

```

INPUT: Frame F, calibration params  $\{\theta_{\text{min}}, \theta_{\text{max}}, V_{\text{min}}, V_{\text{max}}\}$  OUTPUT: Gauge reading V, annotated frame F'
1. G ← Grayscale(F)
2. G_b ← GaussianBlur(G, kernel=5×5, σ=1.5)
3. G_c ← CLAHE(G_b, clipLimit=2.0, tileGrid=8×8)
4. (a, b, r) ← HoughCircles(G_c, minDist=0.8·min(W,H), param2=40, minR=60, maxR=220)
5. ROI ← Crop(G_c, center=(a,b), radius=r)
6. E ← Canny(ROI, T_low=50, T_high=150)
7. S ← ProbabilisticHoughLines(E, ρ=1, θ=π/180, threshold=30, minLen=r/3, maxGap=10)
8. S_f ← LengthFilter(S, r) ∩ CentralityFilter(S, a, b, r)
9. θ_curr ← CollinearityCluster(S_f)
10. θ_norm ← NormalizeAngle(θ_curr)
11. θ_smooth ← EMA(θ_norm, α=0.3)
12. V ← V_min + (θ_smooth - θ_min) · (V_max - V_min) / (θ_max - θ_min)
13. F' ← DrawOverlay(F, circle=(a,b,r), pointer, V)
14. RETURN V, F'

```

B. Edge Device Deployment Strategy

Deployment on the Raspberry Pi 4 (4 GB) requires several platform-specific optimizations. OpenCV is compiled from source with NEON SIMD acceleration enabled for the ARM Cortex-A72 architecture, yielding approximately 20% throughput improvement over the pre-built package binaries. The camera capture pipeline leverages the picamera2 library's zero-copy buffer interface to eliminate redundant frame copies between the camera ISP and the application processing thread. Multi-threading is employed to decouple frame capture, processing, and MQTT transmission, preventing I/O-bound telemetry operations from introducing processing latency jitter.

For Pi Zero 2W deployment (512 MB RAM), the resolution is reduced to 320×240 and the CLAHE tile grid is reduced to 4×4. Processing rate on the Pi Zero 2W is approximately 8–12 FPS, sufficient for slow-changing process variables such as storage tank pressure or thermal process temperature where measurement update rates of 1–5 Hz are adequate.

C. Failure Mode Handling

The system incorporates explicit failure detection and graceful degradation logic. If HCT returns no valid circle (e.g., due to severe glare or occlusion), the system retains the last valid ROI parameters for up to 10 consecutive frames before flagging a localization failure. If PHLT returns no candidate segments passing the filter criteria, the last valid pointer angle is retained with a quality flag set to STALE. These mechanisms prevent spurious readings from being transmitted during transient disruptions while maintaining continuous MQTT presence for upstream monitoring systems.

VI. EXPERIMENTAL RESULTS

A. Experimental Setup and Dataset

Experimental validation was conducted using a curated test dataset comprising 500 static images captured from five distinct gauge types spanning three measurement domains: pressure (0–10 bar, 0–160 psi), temperature (0–150°C), and rotational speed (0–3000 RPM). Gauges were photographed under three controlled illumination conditions: uniform diffuse lighting, directional side lighting inducing partial shadow, and high-intensity frontal lighting creating specular reflection. Ground truth readings were recorded by three independent observers and averaged to mitigate parallax error in the reference measurement.

Video stream performance testing was conducted using a Raspberry Pi 4 Model B (4 GB) connected to a Raspberry Pi Camera Module v2 (Sony IMX219) at 640×480 resolution. Computational benchmarking was averaged over 1,000 consecutive processed frames after an initial 100-frame warm-up period to account for OS scheduler settling.

B. Accuracy Assessment

Method	Mean Error (%)	Max Error (%)	Hardware Required	Training Data
Proposed (calibrated)	0.1 – 0.5	1.2	RPi 4 (CPU)	None
Wang et al. [2] (SGR)	< 0.5	N/A	PC (CPU)	None
Reitsma et al. [1] (DL)	~2.0	~6.0	GPU recommended	Large dataset
Peixoto et al. [3] (IoT)	~0.95	N/A	PC + Cloud	Moderate
Yang et al. [7] (DL-Seg)	< 1.5	N/A	Jetson Nano	Large dataset

Table I: Comparative Accuracy of Gauge Reading Approaches

The calibrated pipeline achieves a mean reading error of 0.1% to 0.5% across all test conditions, directly comparable to the best-performing classical method of Wang et al. [2] and substantially outperforming uncalibrated deep learning regression [1] which reports a 2% relative error. The maximum observed error of 1.2% occurred under extreme directional lighting conditions creating a strong shadow gradient across the pointer, temporarily degrading Canny edge detection sensitivity.

C. Computational Performance

Hardware Platform	Avg. Latency (ms)	FPS	CPU Utilization	Memory (MB)
Raspberry Pi 4 (4 GB)	35.4	28.2	62%	148
Raspberry Pi Zero 2W	95.8	10.4	89%	64
Intel i5 Laptop (PC)	12.1	82.6	18%	142
DeepLabv3+ (RPi 4)	> 250	< 4	100%	512+

Table II: Computational Performance Across Hardware Platforms

On the primary target platform (Raspberry Pi 4, 4 GB), the proposed pipeline achieves 28.2 FPS with an average per-frame latency of 35.4 ms, meeting the real-time processing threshold (defined as ≥ 25 FPS in the IEC 62304 medical device software standard commonly applied by analogy in industrial CV systems). Profiling reveals that the PHLT stage accounts for approximately 45% of total processing time, followed by CLAHE (22%) and HCT (18%). Comparative benchmarking against DeepLabv3+-based segmentation on the same hardware confirms that deep learning inference exceeds 250 ms per frame on CPU without GPU acceleration, rendering real-time performance unachievable at this hardware tier [7][14].

D. Robustness Analysis

The proposed system maintains reliable performance (mean error < 0.5%) across 83% of the test image set. The primary failure modes identified are:

- 1) **Specular Glare:** High-intensity reflections from the gauge glass completely saturate the pointer region, preventing edge detection. Encountered in 8% of test images. Mitigated by polarizing filter on camera lens.
- 2) **Severe Perspective Distortion:** Angles of incidence exceeding 25° cause apparent foreshortening of the pointer, biasing the measured angle. Encountered in 5% of images. Mitigated by enforcing perpendicular camera mounting during installation.

3) Multiple Pointer Ambiguity: Bi-metallic or compound gauges with two coaxial pointers confuse the centrality filter. Encountered in 4% of test images. Requires manual disambiguation rule configuration.

VII. DISCUSSION

A. Technical Analysis

The experimental results confirm that the proposed classical pipeline, when augmented with gauge-specific semi-automated calibration, achieves accuracy equivalent to state-of-the-art classical methods and substantially superior to uncalibrated deep learning regression. This outcome validates the central hypothesis that the calibration step—which encodes expert knowledge of the gauge scale without requiring image-level annotation—effectively compensates for the generalizability limitations inherent to purely geometric approaches.

The use of CLAHE as a preprocessing step contributes disproportionately to robustness under variable illumination. Without CLAHE, mean error increases to approximately 1.8% under directional lighting conditions due to degraded Canny edge detection on low-contrast pointer edges. This finding underscores the importance of local contrast normalization as a preprocessing investment in industrial environments where lighting conditions are heterogeneous and often uncontrolled.

B. Industrial Applicability and Cost Analysis

The complete edge monitoring unit, comprising the Raspberry Pi 4, camera module, power supply, weatherproof enclosure, and mounting hardware, can be produced at approximately USD 85–110 per gauge monitoring node. This represents a cost reduction of one to two orders of magnitude compared to commercial IIoT sensor replacement programs (USD 500–2,000 per point for new digital transmitters with appropriate explosion-proof ratings). For a facility with 200 gauges, the total infrastructure cost of a CV-based monitoring system (USD 17,000–22,000) compares favorably with a rip-and-replace digital upgrade (USD 100,000–400,000) when hardware, installation, and recertification costs are accounted for.

The one-time calibration overhead of approximately two minutes per gauge type, combined with the system's ability to reuse calibration parameters across multiple identical gauge instances, renders the setup process practical for facility-scale deployment. A calibration management system indexing gauge types by serial number and model enables configuration sharing across a fleet of identical gauges.

VIII. FUTURE WORK

Several research directions present opportunities to extend the proposed framework:

1) Automatic Scale Recognition: Integration of lightweight OCR-based or deep learning-based scale mark reading to automate the calibration step, eliminating operator involvement entirely. Few-shot learning approaches trained on synthetic gauge dial images may provide a viable path to data-efficient scale recognition.

2) Multi-Gauge Scene Processing: Extension of the pipeline to simultaneously process multiple gauges within a single camera field of view, enabling panel-level monitoring with a single camera node and reducing hardware cost per gauge further.

3) Non-Circular Gauge Support: Adaptation of the pipeline to handle non-circular gauge formats including linear scales, thermometers, and bar-type indicators prevalent in HVAC and fluid systems.

4) Anomaly Detection Integration: Augmentation with time-series anomaly detection algorithms operating on the continuous gauge reading stream, enabling automated alerts for abnormal rate-of-change events indicative of process upsets or equipment failures.

5) Hindi and Multilingual Labeling: Extension of the OCR calibration module to support Devanagari script and other regional language scale annotations, improving applicability in South Asian industrial deployments.

6) Federated Learning for Adaptation: Deployment of a federated learning protocol across a fleet of installed monitoring nodes to collaboratively refine pointer segmentation models from operational data without centralizing raw imagery, preserving data privacy while improving system-wide performance.

IX. CONCLUSION

This paper has presented a comprehensive, modular computer vision framework for the automated real-time digitization of analogue circular gauges, targeted at cost-effective industrial retrofitting in the context of Industry 4.0 and IIoT integration. The proposed pipeline employs classical geometric image processing—Hough Circle Transform for gauge localization, Probabilistic Hough Line Transform for pointer extraction, and angular interpolation for value computation—augmented by adaptive preprocessing with CLAHE and a structured semi-automated calibration protocol.

Experimental validation on a diverse test dataset demonstrates a mean reading accuracy of 0.1–0.5% under controlled conditions and real-time processing at 28.2 FPS on the Raspberry Pi 4 Model B, a low-cost ARM-based edge computing platform. These results establish that the proposed approach achieves accuracy equivalent to the best published classical methods, substantially outperforms uncalibrated deep learning regression, and operates at 7× lower latency than deep segmentation models on equivalent hardware.

The system's reliance solely on classical CV algorithms and open-source software libraries (OpenCV, Python), combined with its operation on commodity hardware costing less than USD 110 per monitoring node, positions it as a financially accessible and technically viable solution for the digitalization of legacy instrumentation in SMEs and process industries where capital investment for digital sensor replacement is prohibitive.

The principal limitations of the current implementation—sensitivity to severe specular glare and manual calibration requirement—are well-defined and mitigable through physical camera installation design and future automated calibration extensions respectively. The framework thus provides a robust, practically deployable bridge between mechanical analogue instrumentation and the digital IIoT ecosystem, addressing a critical gap in the industrial digitalization landscape.

REFERENCES

- [1] M. Reitsma, J. Keller, K. Blomqvist, and R. Siegwart, "Under pressure: learning-based analog gauge reading in the wild," in Proc. 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 2024, pp. 14–20.
- [2] 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, Art. no. 7490, Oct. 2022.
- [3] 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, Art. no. 9858, Dec. 2023.
- [4] A. Jelidi, D. Kraus, O. Saukh, J. Pestana, P. Priller, and K. Römer, "Benchmarking Vision-based Analog Gauge Reading Methods for Vehicle Dashboard Validation," in Proc. 2025 IEEE International Automated Vehicle Validation Conference (IAVVC), 2025.
- [5] E. Milana, O. H. Ramírez-Agudelo, and J. E. Schmiedt, "Autonomous Reading of Gauges in Unstructured Environments (GAUREAD)," *Sensors*, vol. 22, no. 17, Art. no. 6681, 2022.
- [6] M.-Q. Tran, H. T. Tran, and V. Pham, "Computer Vision System for Reading Analog Gauges at Power Substation," in Proc. 2021 International Symposium on Electrical and Electronics Engineering (ISEE), 2021.
- [7] Y. Yang, W. Liao, S. Fan, J. Hou, and H. Tang, "Automatic Reading Method for Analog Dial Gauges with Different Measurement Ranges in Outdoor Substation Scenarios," *Information*, vol. 16, no. 3, Art. no. 226, 2025.
- [8] S. Patel, "Project Synopsis on Analogue Gauge Detection using OpenCV," Anjuman College of Engineering and Technology, Nagpur, India, 2025.
- [9] D. Kong et al., "GaugeTracker: AI-Powered Cost-Effective Analog Gauge Monitoring System," arXiv preprint arXiv:2410.xxxxx, 2024.
- [10] J. Peixoto et al., "Development of an Analog Gauge Reading Solution Based on Computer Vision and Deep Learning for an IoT Application," *Telecom*, vol. 3, no. 4, pp. 564–580, 2022.
- [11] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.
- [12] Y.-S. Chen and J.-Y. Wang, "Computer Vision-Based Approach for Reading Analog Multimeter," *Applied Sciences*, vol. 8, no. 8, Art. no. 1268, Jul. 2018.
- [13] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [14] H. Ninama et al., "Computer vision and deep transfer learning for automatic gauge reading detection," *Scientific Reports*, vol. 14, 2024.