

Article

An Implementation of Creep Test Assisting System with Dial Gauge Needle Reading and Smart Lighting Function for Laboratory Automation

Dezheng Kong ¹, Nobuo Funabiki ^{1,*}, Shihao Fang ¹, Noprianto ¹, Mitsuhiro Okayasu ¹
and Pradini Puspitaningayu ²

¹ Graduate School of Environmental, Life, Natural Science and Technology, Okayama University, Okayama 700-8530, Japan; pxe6n2q@s.okayama-u.ac.jp (D.K.); pkpb8c9q@s.okayama-u.ac.jp (S.F.); py3o92mw@s.okayama-u.ac.jp (N.); okayasu@okayama-u.ac.jp (M.O.)

² Department of Electrical Engineering, Universitas Negeri Surabaya, Surabaya 60231, Indonesia; pradinip@unesa.ac.id

* Correspondence: funabiki@okayama-u.ac.jp

Abstract: For decades, *analog dial gauges* have been essential for measuring and monitoring data at various industrial instruments including production machines and laboratory equipment. Among them, we focus on the instrument for *creep test* in a mechanical engineering laboratory, which evaluates material strength under sustained stress. Manual reading of gauges imposes significant labor demands, especially in long-duration tests. This burden further increases under low-lighting environments, where poor visibility can lead to misreading data points, potentially compromising the accuracy of test results. In this paper, to address the challenges, we implement a *creep test assisting system* that possesses the following features: (1) to save the installation cost, a web camera and *Raspberry Pi* are employed to capture images of the dial gauge and automate the needle reading by image processing in real time, (2) to ensure reliability under low-lighting environments, a smart lighting mechanism is integrated to turn on a supplementary light when the dial gauge is not clearly visible, and (3) to allow a user to stay in a distant place from the instrument during a creep test, material break is detected and the corresponding message is notified to a laboratory staff using *LINE* automatically. For evaluations, we install the implemented system into a material strength measuring instrument at Okayama University, Japan, and confirm the effectiveness and accuracy through conducting experiments under various lighting conditions.

Keywords: creep test; Raspberry Pi; dial gauge; needle reading; smart lighting



Academic Editor: Luciano Feo

Received: 28 January 2025

Revised: 27 March 2025

Accepted: 31 March 2025

Published: 2 April 2025

Citation: Kong, D.; Funabiki, N.; Fang, S.; Noprianto; Okayasu, M.; Puspitaningayu, P. An

Implementation of Creep Test Assisting System with Dial Gauge Needle Reading and Smart Lighting Function for Laboratory Automation. *Technologies* **2025**, *13*, 139. <https://doi.org/10.3390/technologies13040139>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Currently, analog meters are still widely utilized in industrial productions and laboratory equipment for data measurements and monitoring [1]. Despite advancements in digital technology and automation, analog meters remain indispensable in various fields, particularly in factory settings where manual reading of meter values continues to be a critical task [2,3].

However, this process is time-consuming and labor-intensive, especially in low-light environments such as nighttime or cloudy weather. In addition, the reduced visibility of the gauges can increase the likelihood of misreading or missing critical data points, potentially compromising the accuracy of the results.

Among analog meters, a *dial gauge* is a type of the precision instrument commonly employed in a variety of tests, including *creep testing*. Creep tests evaluate the strength of materials under sustained stress by measuring deformation over time until the material is broken [4]. Figure 1 illustrates a fundamental setup of a creep testing system, showing the material under sustained stress. The three colored points at the bottom represent the rollers that contact the specimen and the arrows indicate the directions of the applied load and the corresponding reactions. Essentially, the middle colored point applies the force on the specimen, while the two outer colored points provide support.

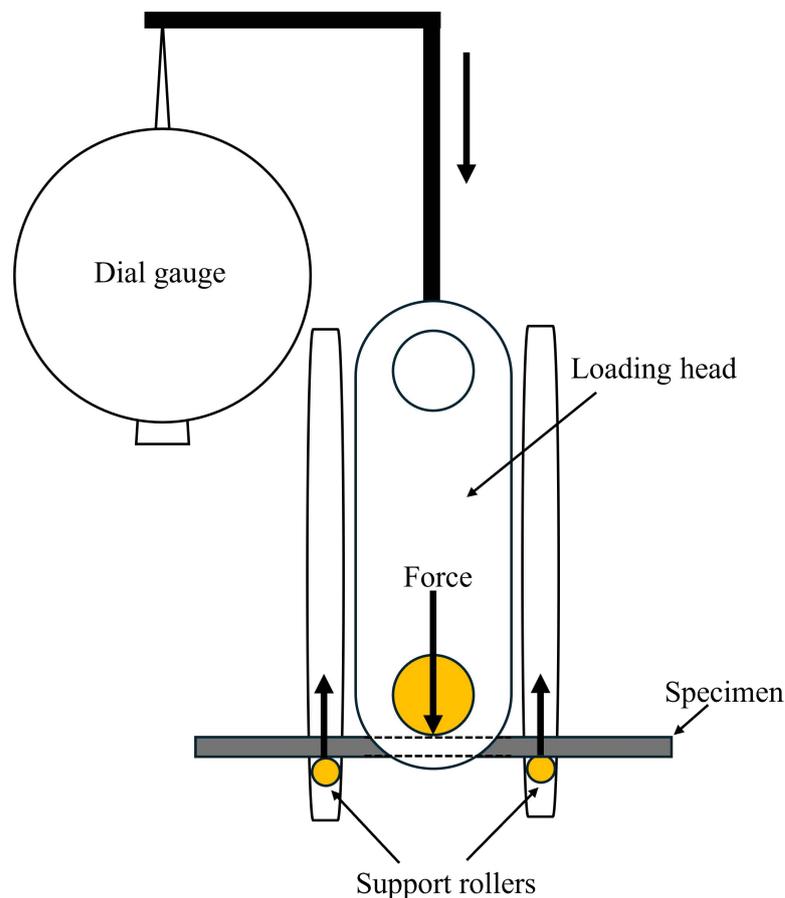


Figure 1. Three-point bending creep test.

In such tests, *dial gauges* [5] are used to measure material deformation, where they present several challenges. First, they lack the capability to output digital data, requiring experimenters to manually extract readings from recorded video footage, which is time-consuming and error-prone. Second, under low-lighting conditions, such as nighttime or cloudy weather, the visibility of a dial gauge is significantly reduced, leading to misreading or missed data points and compromising the measurement accuracy. Third, when the main needle of a dial gauge overlaps with the sub-needle, the visual distinction becomes difficult, further increasing the risk of misinterpretation. Lastly, the determination of the material failure currently relies on manual observations, which can delay the response and introduce variability into the evaluation process. These challenges collectively hinder the efficiency and reliability of the creep test data collection and analysis. Figure 2 shows a standard dial gauge.

To address the identified challenges in automating the creep test data collection, we propose an innovative system employing *Raspberry Pi* [6] and a *web camera* for digitizing analog dial gauge readings through image processing techniques. This system first utilizes

Hough Transform [7] and edge detection [8] to identify the needle and calculate its angle. Then, the detected needle angle is converted into digital measurement data. To enhance accuracy, several image processing techniques such as dilation, erosion [9], and masking are applied to suppress background noise and mitigate interference from the sub-pointer. Additionally, the system integrates a smart lighting mechanism to ensure reliable performances under low-light conditions by dynamically activating and deactivating supplementary lighting based on gauge visibility.

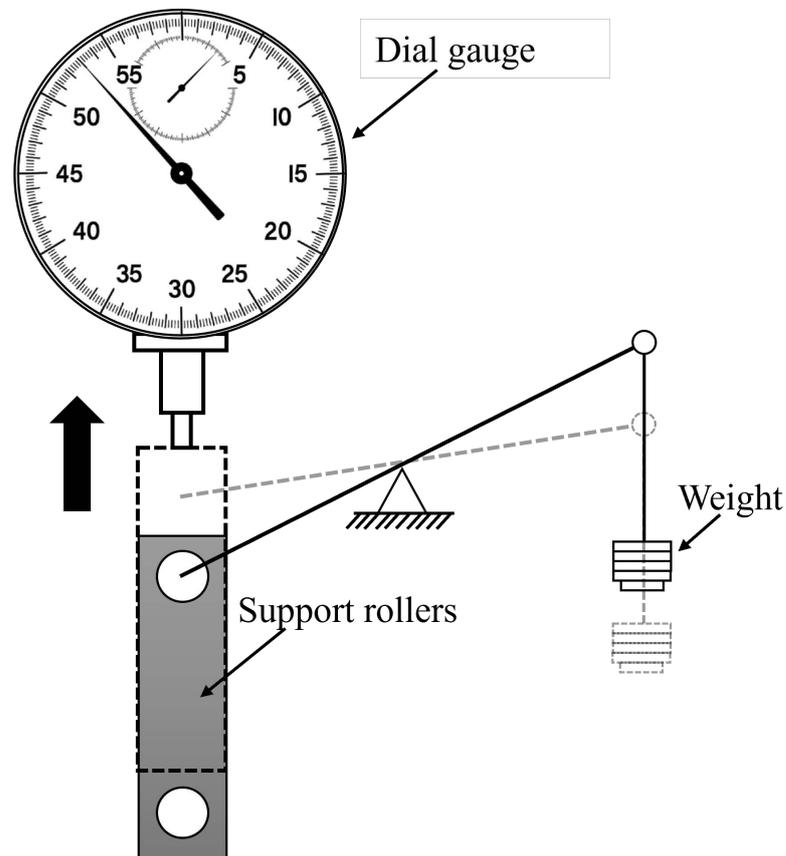


Figure 2. Overview of creep testing system using dial gauge.

For evaluations of the proposed system, we confirmed the effectiveness through extensive experiments under diverse lighting conditions, demonstrating improvements in measurement accuracy, lighting efficiency, and energy consumption. The proposed needle detection method achieved an average error of 0.0010 mm, which supports its sufficient precision in digitizing analog readings.

The implemented smart lighting mechanism effectively reduced the supplementary lighting duration, significantly decreasing energy consumption. In long-duration creep tests, especially those that span several days or weeks, leaving the supplementary light on continuously can result in substantial cumulative power usage, particularly when multiple test systems are running in the same laboratory. Specifically, lighting time was reduced by 30.0% under the same experimental conditions compared to the conventional method that keeps lighting constantly on. This reduction not only demonstrates the system's energy efficiency but also helps to minimize visual disturbance in shared laboratory environments. It ensures the robust performance in automating the creep test data collection, maintaining consistent image quality under varying ambient lighting conditions and providing precise measurements, sustainable operations, and adaptability across various environmental conditions, making it highly suitable for practical applications.

The rest of this paper is organized as follows: Section 2 reviews related works on automated data collection and IoT-based solutions for mechanical testing instruments. Section 3 discusses the challenges in monitoring creep test instruments, focusing on manual monitoring issues and the importance of reliable data collection. Section 4 presents the proposed *creep test assisting system*, covering system architecture, image processing for needle detection, smart lighting, data processing, database integration, and user interface design. Section 5 evaluates the system through experiments and performance analysis. Section 6 concludes the paper and discusses future work.

2. Related Works

In this section, we review relevant studies in the literature on automated systems for reading and monitoring analog gauges. These prior works can be categorized into three groups based on their technical approaches: robot-assisted gauge monitoring systems, vision-based needle reading methods, and IoT-integrated remote monitoring frameworks.

2.1. Robot-Assisted Gauge Monitoring Systems

Wang et al. [10] proposed an automatic reading system for analog instruments based on computer vision and an inspection robot for power plants. The system integrates computer vision with robotic platforms to facilitate automatic monitoring of analog instruments in industrial environments. However, it relies on inspection robots for data collection, which increases the system's complexity and cost.

Huang et al. [11] designed a robotic gauge monitoring system using a Hikvision PTZ camera, Hangzhou, China and a Siasun Robot & Automation Co., Ltd. inspection robot, Changchun, China,, capable of aligning with multiple analog instruments automatically. Their vision-based needle detection supports various gauge types. However, the alignment process demands substantial computational resources, limiting real-time performance.

2.2. Vision-Based Needle Reading Methods

Tian et al. [12] proposed a pointer location algorithm for computer vision-based automatic reading recognition of pointer gauges. This algorithm improved the accuracy of pointer detection and reading recognition, demonstrating effectiveness in industrial environments. However, it did not address the challenges of real-time monitoring, which are critical for long-duration experiments in dynamic environments.

Lauridsen et al. [13] presented an image processing pipeline for automated recognition and translation of pointer movements in analogue circular gauges. Their method processes video frames to identify key parts of the gauge and determine the pointer's angle, producing a digital time series of the measurements. However, their approach primarily focuses on static image analysis and does not address real-time monitoring.

Dumberger et al. [14] proposed an autonomous real-time gauge reading system for industrial environments. Their system effectively detects gauges, identifies pointer needles, and extracts measurement values without requiring prior knowledge of gauge configurations, making it well-suited for hazardous industrial settings. However, their system depends on autonomous robots, which may not be cost-effective for laboratory-scale applications.

Howells et al. [15] proposed a real-time analogue gauge transcription system for mobile phones using convolutional neural networks (CNNs). The system achieved high accuracy, with pointer angle errors of less than one degree, and introduced large-scale synthetic and real-world datasets for training and testing. However, their system is designed specifically for mobile platforms and focuses on standalone gauge reading, lacking integration with experimental setups and IoT-based monitoring.

Trairattanapa et al. [16] proposed a system for real-time multiple analog gauge reading designed for autonomous robot applications. This system uses computer vision to monitor and interpret multiple gauges simultaneously, enabling efficient data collection in industrial environments. However, it does not address low-light conditions or provide remote notification mechanisms.

2.3. IoT-Integrated Remote Monitoring Frameworks

Peixoto et al. [17] proposed an end-to-end solution for analog gauge monitoring using computer vision integrated into an IoT platform. This system captures images of analog gauges and transmits the processed data to a remote IoT server for monitoring and analysis. The prototype achieved high accuracy in industrial environments, highlighting its practical applicability. However, the system did not include features for handling low-light conditions or providing real-time notifications, which are critical for long-duration experiments where environmental factors may affect data visibility.

Smith et al. [18] proposed an automated analog gauge monitoring system using computer vision integrated with an IoT framework. However, the system lacks robust solutions for low-light conditions, which can lead to reduced accuracy in poorly illuminated industrial environments.

Garcia et al. [19] introduced a real-time remote monitoring framework that integrates IoT devices with machine vision. Nonetheless, the framework does not support real-time notifications, which is essential for ensuring prompt responses in dynamic industrial settings.

Chen et al. [20] developed an IoT-based system for remote monitoring of analog meters through image processing techniques. Yet, the system lacks an adaptive image pre-processing module to effectively mitigate environmental noise interference, potentially compromising data quality over prolonged periods.

3. Challenges in Monitoring Creep Test Instruments

In this section, we examine the challenges associated with monitoring creep test instruments, focusing on the limitations of manual monitoring methods.

3.1. Key Issues in Manual Monitoring of Dial Gauges

Manual monitoring of dial gauges in creep test instruments presents multiple challenges that hinder efficiency and data reliability. These issues are particularly evident in scenarios involving long-duration tests, low-light environments, and energy-intensive processes. Each of them are explored below to provide a comprehensive understanding of limitations faced in traditional monitoring methods.

3.1.1. Labor-Intensive Monitoring in Long-Duration Tests

Creep tests demand meticulous monitoring due to their prolonged durations, often requiring staff to record measurements manually over weeks or months. This repetitive and labor-intensive task introduces not only operational inefficiency but also risks of errors due to human fatigue. The presence of multiple testing instruments further compounds workloads, making it impractical to ensure the consistent accuracy across all devices. Furthermore, manual monitoring increases dependency on highly skilled personnel, which is costly and resource-intensive. These challenges underline inadequacies of manual approaches and highlight the necessity for automation to enhance the accuracy and efficiency in long-duration testing. In [21], Yang et al. highlighted that the automation of pointer meter readings using computer vision technology can substantially alleviate labor intensity while improving the accuracy of measurements.

3.1.2. Visibility Problems in Low-Light Environments

Creep test instruments are often located in environments with inadequate or inconsistent lighting. For instance, during long-term experiments lasting several weeks, the intensity of natural light can vary significantly due to weather changes or diurnal cycles, which can introduce inconsistencies in gauge visibility. This inconsistency not only increases the risk of misreading, but also complicates the calibration process. Furthermore, the reliance on portable lamps or overhead lighting often leads to uneven illuminations, which interferes with the edge detection algorithm during image processing.

This issue becomes especially critical during off hours when ambient light is limited. In such situations, operators often struggle to accurately read dial gauges, leading to increased measurement inaccuracies. In addition, prolonged exposure to low-light conditions can strain operators' visions, further affecting the reliability of collected data. Additionally, the reliance on temporary lighting, such as portable lamps, introduces uneven illuminations, exacerbating visibility problems. These factors collectively degrade the quality of monitoring and underscore the importance of incorporating solutions that ensure consistent visibility in all lighting conditions. In [22], Barbosa et al. demonstrated that leveraging smartphone-integrated systems with adaptive lighting significantly improves the gauge visibility in low-light conditions, ensuring reliable data collection.

3.1.3. High Energy Consumption Issues

The implemented smart lighting mechanism effectively reduced the supplementary lighting duration, which significantly decreased energy consumption. In long-duration creep tests lasting for several days or weeks, continuously turning on the supplementary light can result in substantial cumulative power usage. It becomes particularly significant when multiple testing are running at the same time. Specifically, it was found that lighting time was reduced by 30.0% compared to the conventional method that keeps lighting constantly on under the same experimental conditions. This reduction not only demonstrates the proposed system's energy efficiency but also helps to minimize the visual disturbance in shared laboratory environments. It ensures the robust performance in automating the creep test data collection, maintaining the consistent image quality under varying ambient lighting conditions, and providing precise measurements, sustainable operations, and adaptability across various environmental conditions, making it highly suitable for practical applications.

In conventional creep testing environments, reliance on continuous lighting for visual monitoring can significantly increase energy consumption, especially for long-duration experiments. Laboratories often maintain illumination for 24 h to ensure the dial gauge visibility, even when no one is present there, resulting in inefficient energy usage, elevated operational costs, and unnecessary environmental burdens. Moreover, the use of supplementary lighting to enhance image clarity under low-light conditions further amplifies this issue. As more test devices are introduced in laboratory environments, the cumulative energy demand becomes unsustainable over time. The current energy price rise has made it even more serious.

Addressing this challenge requires a solution that balances the measurement reliability with the power efficiency. The proposed smart lighting mechanism is designed to automatically activate the supplementary light only when the dial gauge needle is not clearly visible and to turn it off otherwise. Compared to the baseline method of lighting on constantly, our system reduces the total lighting time by 30.0% under identical test conditions, thereby reducing the energy consumption and minimizing the unnecessary illumination. In [23], Liu et al. proposed a system integrating smart lighting mechanisms to optimize the energy use while ensuring adequate illumination for precise gauge reading.

3.2. Importance of Reliable Data Collection for Creep Testing

Reliable data collection is critical for ensuring the accuracy and validity of creep test results, since these tests are instrumental in evaluating the long-term mechanical properties of materials under sustained stress. Even small errors in data recording can lead to incorrect conclusions of material performances, potentially resulting in flawed designs or unsafe structures in real-world applications. Moreover, the long duration of these tests necessitates consistent and accurate data monitoring to capture gradual changes in material behavior. Inconsistent or missing data not only compromise the reliability of results but also render the entire testing process inefficient. Therefore, implementing a robust data collection system is essential to enhance accuracy, reduce manual errors, and ensure that the results are dependable and repeatable for critical engineering applications.

4. Proposed Creep Test Assisting System

In this section, we present the design and functionality of the *creep test assisting system*. This system integrates sensor data collections, image processing, and smart decision-making to automate and enhance the efficiency of creep tests.

4.1. System Architecture

The system architecture is designed to provide a cost-effective modular approach for automating dial gauge monitoring. It integrates hardware and software components to ensure seamless operation and data reliability.

4.1.1. Hardware

Figure 3 illustrates the architecture of the *creep test assisting system*. The overall workflow of the system begins with hardware components. The system primarily consists of a dial gauge, a web camera, and *Raspberry Pi*. The dial gauge is responsible for recording the material's micro-deformation in real time during the creep test under applied stress. To ensure the complete recording of its changes during each measurement cycle, a web camera is fixed directly in front of the gauge. The web camera is configured to capture high-resolution images at scheduled intervals, transmitting the images to *Raspberry Pi* via a *USB* interface for subsequent processing.

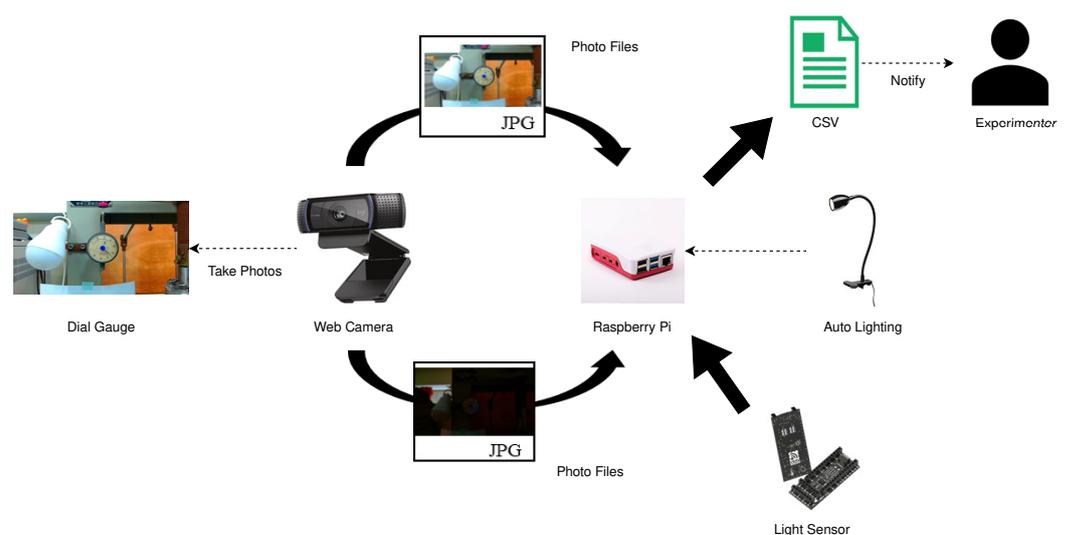


Figure 3. System architecture of creep test assisting system.

4.1.2. Image Processing

Upon entering the image processing module, the captured raw images are first converted to grayscale, reducing the interference from unnecessary color information and improving processing efficiency. Next, *Gaussian* blur [24] is applied to the images to eliminate noise caused by light fluctuations or equipment vibrations. The denoised images are processed with *Canny* edge detection to extract critical features, including the scale circle and needle outline. Subsequently, the system employs the *Hough* transform algorithm to accurately locate the dial gauge's center and the edges of the scale circle, calculating the needle's angular position. This angular position is then mapped to the scale range of the dial gauge to derive numerical measurement values, ensuring highly accurate readings even in complex environments.

4.1.3. Smart Lighting

To ensure the image clarity in low-light conditions, the system incorporates a smart lighting module. By connecting to a light intensity sensor, *Raspberry Pi* continuously monitors the ambient light level in the laboratory and analyzes the brightness histogram of the captured images. When the light intensity sensor detects insufficient lighting or the image brightness is deemed inadequate, the system automatically activates the auxiliary *LED* light to provide uniform and stable illumination for the dial gauge. This process continues until the lighting conditions meet the requirements for accurate image processing.

4.1.4. Data Processing

The data processing module serves as the core of the decision-making capability. This module receives real-time measurement data from the image processing module and performs time-series analysis to record the stress and strain variations over time. Based on predefined fracture criteria, the system detects material fractures by identifying sharp drops in stress. Upon detecting such events, the system triggers a control signal to terminate the experiment and sends an experimental termination notification to laboratory personnel via the *LINE* messaging platform. This notification includes the experimental time, the measured value at the time of fracture, and the corresponding experimental image, enabling laboratory personnel to respond promptly and take further action.

4.1.5. Data Storage

All the experimental data, including timestamps, measured values, ambient light intensity, and image processing results, are stored in the local database of the data storage. This database ensures secure and rapid data storage and retrieval, avoids the potential risks of network dependency, and supports offline access.

4.1.6. User Interface

The user interface module provides a platform for interactions between the system and the laboratory personnel. The user interface can be accessed through web or mobile applications, displaying real-time gauge readings, stress–strain variation curves, and environmental parameters in intuitive formats. Additionally, the interface supports browsing historical data and allows experimental data to be exported in standard formats for subsequent analysis or report generation.

4.2. Image Processing for Needle Detection

The methodology for reading the dial gauge is described here, including the dial detection, the image preprocessing, the noise reduction, the *Hough* transform for needle detection, and the conversion of the detected needle angle into the measurement value.

4.2.1. Hough Transform for Circle Detection

Before applying the image preprocessing and the needle detection, the position and boundary of the dial is extracted using the *Hough* circle transform [25], which is a geometry-based detection method that identifies a circle by searching for the center and radius in the parameter space.

The *Hough* circle transform is defined by the following equation in the parameter space:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (1)$$

where (a, b) represents the circle center coordinates and r stands for the radius. To represent a circle in the parameter space, this equation can be rewritten by

$$a = x - r \cdot \cos \theta, \quad b = y - r \cdot \sin \theta \quad (2)$$

By iterating this equation over each edge point at (x, y) in the image with a different radius value r and recording the votes for each (a, b, r) , the point with the highest accumulated votes is selected as the optimal circle center and radius.

To detect the dial gauge circle, first, the edge detection algorithm is applied to the input image to extract salient edge points, ensuring the input data for the circle detection are geometrically well defined. During the parameterized search phase, all of the possible circle combinations are examined within a predefined radius range (100 to 300 pixels) and the step size (20 pixels). Each edge point is mapped to the parameter space based on potential radius values, generating a distribution of accumulated votes. The optimal combination of the center and radius is determined by selecting the highest-voted point in the parameter space. These detected optimal parameters are then used to crop the image and extract the dial region while recording the center coordinates and radius for further analysis. As shown in Figure 4, our program effectively detects the circle region for the dial gauge, providing a reliable basis for subsequent processing. The green circle highlights the detected dial gauge region used for subsequent image processing. The label above the gauge is written in Japanese, indicating the manufacturer's name.



Figure 4. Dial gage circle detection.

4.2.2. Preprocessing and Noise Reduction

To ensure the accurate detection of the primary needle and to minimize interference from secondary needles and background noise, the input image of the detected dial under-

goes the following preprocessing steps. (1) The input image is converted from a color image to a grayscale image, and all areas except for the dial are cropped to reduce unnecessary information. (2) The grayscale image undergoes binarization, making the pixel value into 0 (black) or 255 (white) based on a set threshold, generating a high-contrast edge image. This step emphasizes the structural features of the needle and scale while eliminating redundant background noise. As shown in Figure 5a, the background is effectively removed after binarization, leaving the needle and dial structure more distinct. (3) The following dilation and erosion operations are applied to the binarized image. These steps aim to further enhance the needle's contour and reduce false edges caused by noise or secondary needles:

- Dilation: Expands the highlighted regions to fill gaps in the image, making the needle's contour more continuous.
- Erosion: Shrinks the highlighted regions to remove small noise points while refining the needle's actual contour.

The dilation and erosion operations use kernels of size 3×3 and 7×7 , respectively. This parameter selection achieves a balance between detection accuracy and computational efficiency. As shown in Figure 5b, the integrity of the needle edge is significantly improved after the processes.

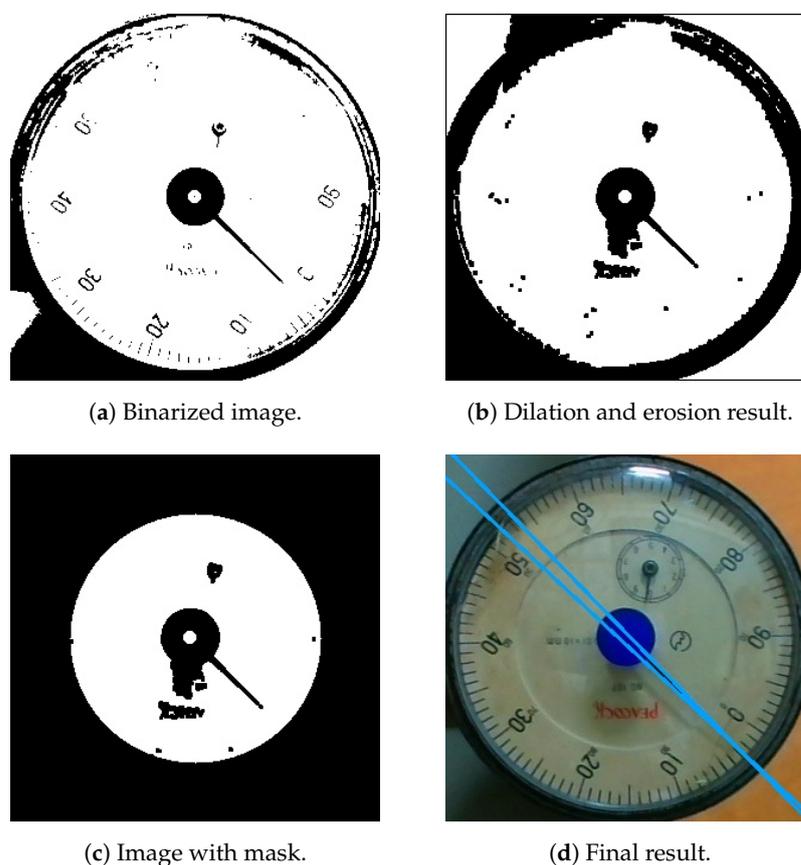


Figure 5. Image processing steps.

Finally, a circular mask is applied to limit the analysis area to the center of the dial. The mask radius is calculated as

$$r_{\text{mask}} = \min(h - 100, w - 100) \quad (3)$$

where h and w represent the height and width of the cropped image containing only the dial. The masked image sets the areas outside the dial region to black, effectively eliminating

the background interference. As shown in Figure 5c, the processed image retains only the center region of the dial, free of extraneous background disturbances. With these preprocessing steps completed, the image is ready for needle detection.

4.2.3. Hough Transform for Needle Detection

After locating the dial using the *Hough* circle transform, the next step involves detecting the needle's position on the dial using another *Hough* line transform for the line detection. The detection stability and accuracy are further enhanced through geometric filtering and feature analysis.

For line detection, the *Hough* line transform maps the edge points from *Cartesian* coordinates to the parameter space, identifying the potential lines in the parameter space. In *Cartesian* coordinates, a line can be expressed as

$$y = ax + b \quad (4)$$

where a is the line's slope and b is its y -intercept. However, this representation fails for vertical lines ($\theta = 90^\circ$), where the slope a becomes undefined.

$$\rho = x \cdot \cos \theta + y \cdot \sin \theta \quad (5)$$

where ρ represents the perpendicular distance from the origin to the line and θ stands for the angle between the perpendicular and the horizontal axis. Edge points (x, y) from the image are transformed into the parameter space, where the intersections of sinusoidal curves indicate the existence of a line. The number of intersections corresponds to the line's likelihood.

In practice, the *OpenCV* function `HoughLines()` is used, where a voting threshold is set to select reliable lines. Higher vote counts correspond to longer and more stable lines. As shown in Figure 5d, the system successfully detects the needle's edges using the *Hough* transform, which serves as a basis for further analysis. Additionally, the blue lines in the figure represents the extension of the detected needle edges.

To improve the line detection accuracy, the geometric filtering and the angle analysis are applied as follows:

First, eight candidate lines are detected using the *Hough* transform instead of the traditional two-line approach. This parameter selection is based on experimental observations, where typically six to eight edge lines are often detected in the needle region of the dial. Increasing the number of candidate lines ensures that more potential needle edges are captured, reducing the risk of missing critical information. However, the number of candidate lines is limited to eight to prevent noises from introducing false positives. As shown in Figure 6a, this proposal successfully identifies eight candidate lines in the needle region, which are indicated by red lines in the figure.

Next, the geometric filtering is applied to all candidate line combinations. Specifically, it calculates the midline of each pair of candidate lines and verifies whether this midline passes through the center of the dial. Additionally, the intersection points of the candidate lines are checked to ensure that they lie within the dial region. Only line pairs meeting these geometric criteria are considered potential needle outlines. As shown in Figure 6b, our program successfully identifies midlines that pass through the dial center and meet the geometric filtering conditions, as indicated by the blue lines in the figure.

Finally, the angle analysis is performed on the filtered line pairs. Experimental results show that the angle between the edges of the needle is typically around 2° . The line pairs with angles close to 2° are filtered, treating these pairs as the needle's outlines. The average angle of these line pairs is then computed to determine the needle's actual position. As shown in Figure 6c, the final needle outlines are accurately identified based on the geometric and angle filtering criteria, as indicated by the red lines in the figure.

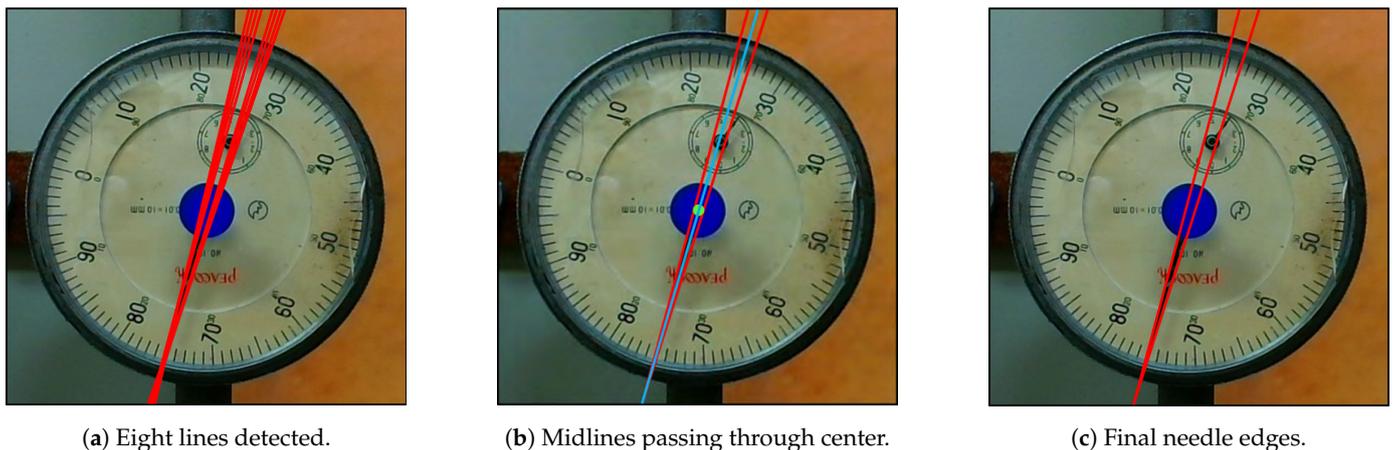


Figure 6. Needle detection steps.

4.2.4. Angle-Based Measurement Conversion Method

After detecting the needle, the detected needle angle is converted into the actual measurement value. The detailed calculation process is as follows.

First, the initial angle θ_i of the needle is recorded at the start of the experiment when the needle points to the “0” scale. For each detected angle θ_t , the difference θ_d between these two angles is calculated using the following formula:

$$\theta_d = \begin{cases} \theta_i - \theta_t, & \text{if } \theta_i \geq \theta_t \\ 360 - (\theta_t - \theta_i), & \text{if } \theta_i < \theta_t \end{cases} \quad (6)$$

Next, the angle difference θ_d is converted into an actual measurement value L using the following equation:

$$L = \frac{\theta_d}{360} [\text{mm}] \quad (7)$$

Additionally, special conditions are considered. During prolonged experiments, the needle may rotate more than one full circle. To address such scenarios, an automatic correction mechanism is incorporated. When it detects that the needle’s angle exceeds 360° , it adds 1 mm to the current measurement value to account for the additional rotation.

Furthermore, in cases where material rupture occurs during the experiment, the needle typically points to the dial’s maximum scale value of 10.2 mm. The final measurement value is calculated for such cases using the following formula:

$$L = 10.2 - l \quad (8)$$

where l is the initial deformation value recorded at the start of the experiment.

These methods ensure the accuracy and reliability of measurements under various conditions, providing a clear basis for data processing and analysis.

4.2.5. Material Fracture Detection

After creating the mask, the system analyzes the pixel distribution of the marker region and calculates its centroid position. By monitoring changes of the centroid’s coordinates, the system determines whether the spindle of the test machine has separated from the dial gauge. When a significant displacement of the centroid is detected and is held over a period of time, the system regards that the specimen has fractured. Figure 7 shows the states of the test machine before the material fracture.

The dial gauge indirectly measures material stress changes by detecting the slight vertical movement of the test machine’s spindle caused by material deformation. When the specimen fractures, as shown in Figure 8, the test machine is separated from the dial

gauge and comes to a complete stop. By monitoring the positional changes in the machine's spindle, the fracturing of the specimen is detected.

To achieve real-time detection, a prominent pink marker is affixed to the contact point between the test machine and the dial gauge. By processing the captured images, the system effectively extracts the marker region and analyzes its motion trajectory to know whether the specimen has fractured.

Specifically, the captured *RGB* [26] images are converted to the *HSV* [27] color space, which enables a more intuitive separation of color and brightness information. The system defines a precise *HSV* value range for the pink marker, where this color was chosen for its high contrast against typical laboratory backgrounds. To optimize noise removals, the *HSV* range was fine-tuned through iterative testing under different lighting conditions, ensuring reliable detections under variable brightness levels.

After creating the mask, the system analyzes the pixel distribution of the marker region and calculates its centroid position. By monitoring changes in the centroid's coordinates, the system determines whether the spindle of the test machine has separated from the dial gauge. When a significant displacement of the centroid is detected and sustained over a period of time, the system concludes that the specimen has fractured.



Figure 7. Before material fracture.

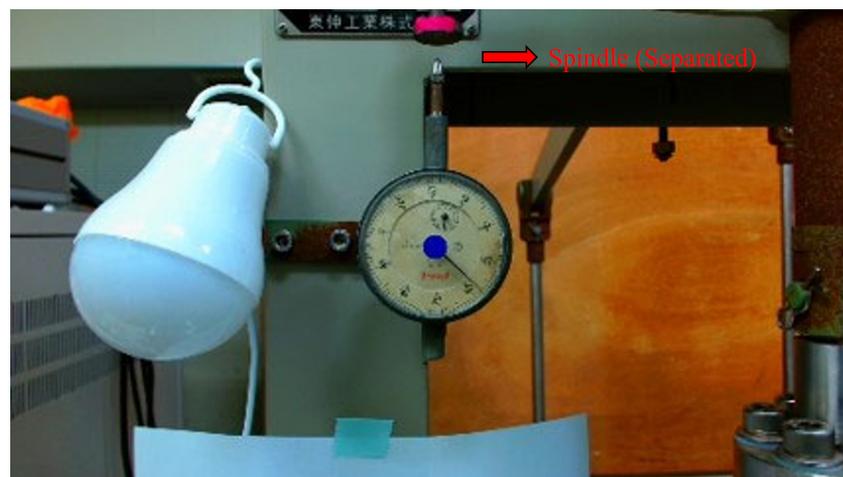


Figure 8. After material fracture.

4.3. Smart Lighting

This section outlines an intelligent lighting function in the proposed system for dynamic control of supplementary lighting under ambient light conditions.

In laboratory environments, an intelligent lighting function can significantly enhance the experimental efficiency and precision while promoting the efficient energy utilization. During daylighting hours, the web camera can clearly capture the dial gauge image without additional lighting. However, under insufficient lighting conditions at nights or in cloudy days, the proper activation of supplementary lighting is essential to ensure the accurate recognition of the dial gauge and needle.

To address this issue, an intelligent lighting function is designed for this creep test assisting system. It monitors ambient lighting conditions in real time and dynamically adjusts the supplementary light's operation. By maintaining the high precision in experimental data collections under varying lighting conditions, the system effectively minimizes energy wastes and reduces interferences with other laboratory experiments.

To achieve the intelligent lighting control in this laboratory, the function combines a light sensor and a camera to ensure reliable and accurate decision-making. First, the light sensor, connected to *Raspberry Pi* via a *Monostick* [28] USB Dongle, continuously collects real-time ambient light intensity data in the laboratory. *Raspberry Pi* compares the received light intensity data with a predefined threshold, which was determined through experiments under various laboratory lighting conditions. Experimental results showed that 300 lux [29] and an average *RGB* value of 180 provided optimal visibility for needle detection, ensuring the high accuracy and minimal energy consumption.

Simultaneously, the camera captures real-time images of the dial gauge and converts them into grayscale images. The average *RGB* values of the pixels in the entire image are calculated to further evaluate the lighting conditions. If the light sensor's detected value falls below the given threshold and the camera's computed image brightness is insufficient, the function integrates the results of both assessments to turn on the supplementary light, ensuring clear images of the dial gauge. Once the lighting conditions return to normal, the function automatically turns off the supplementary light, thereby conserving energy and reducing disruption to other experiments.

By leveraging the combined functionality of the light sensor and the camera, the function can dynamically control the supplementary light under complex lighting conditions, achieving the energy-efficient and effective lighting management.

The hardware components for the intelligent lighting function are illustrated in Figure 9. The light sensor, connected to *Raspberry Pi* via a *USB* dongle, collects ambient light data and transmits it in real time. Concurrently, the camera captures images of the dial gauge, which are analyzed to further evaluate light intensity. The integrated results of these methods are processed at *Raspberry Pi* to control the operation of the supplementary light.

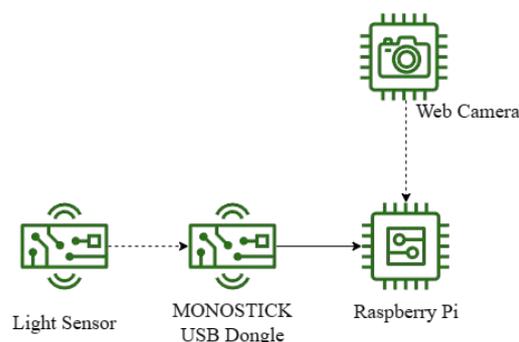


Figure 9. Hardware components for intelligent lighting.

4.4. Data Processing for Decision-Making

This section explains how the system dynamically evaluates experimental conditions by simultaneously measuring *RGB* [26] values and illuminance (lux). The system captures real-time images of the dial gauge using a camera, calculates the average *RGB* values across

the entire image to characterize the color intensity, and uses a light sensor to measure lux values, which represent ambient light intensity [29]. This dual measurement mechanism not only assists in identifying the color details of the dial gauge, such as determining the exact position of the pink marker, but also evaluates whether the current lighting conditions are suitable for recognizing the dial gauge and needle.

Experimental results show that when the average *RGB* value of an image exceeds 180, it meets the requirements. As shown in Figure 10, the three-day experimental measurements of average *RGB* values indicate that when the *RGB* value is low, images are not discernible, whereas values above 180 allow for sufficient detail recognitions. Additionally, to ensure that experiments are conducted under appropriate lighting conditions, we measured variations in the light intensity in the laboratory under different scenarios. As shown in Figure 11, the laboratory's light intensity fluctuates significantly due to natural lights and intermittent operations of other experimental equipment, which may cause temporary activations or deactivations of indoor lighting.

After extensive experimental validation, it was determined that when the measured light intensity exceeds 300 lux, *Raspberry Pi* can accurately recognize the dial gauge and the needle position under both natural and artificial lighting conditions. Consequently, the system establishes two thresholds based on experimental results: an average *RGB* value of 180 and a light intensity of 300 lux, which serve as the foundation for data processing and decision-making to ensure efficient and precise experimental performance.

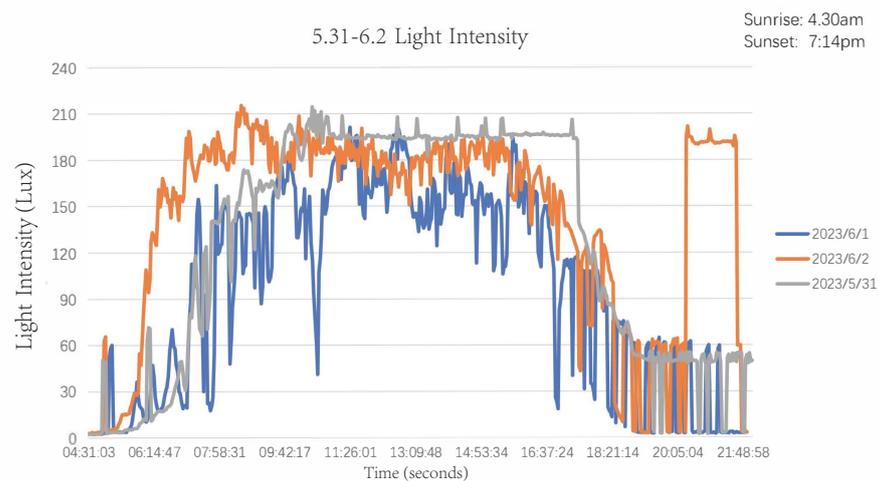


Figure 10. Measured RGB values in creep testing system.

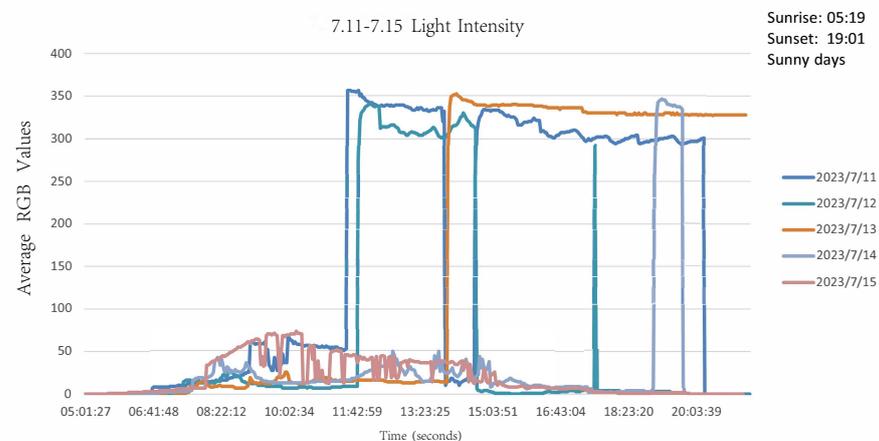


Figure 11. Measured light intensity in creep testing system.

4.5. Database

The proposed system implements a local database on *Raspberry Pi* to store and manage experimental data while providing real-time data access. Using the *Flask* [30] web application framework, a local server is built in *Raspberry Pi*. Experimental data are stored in the CSV format and made accessible to users via a web interface. Additionally, the server allows users to select specific dates or files for data visualization, dynamically generating corresponding line charts. This feature improves the efficiency of data management and enables experimenters to quickly monitor experimental progress.

4.6. User Interface

The user interface consists of two components: a web interface and *LINE Notify* [31] notifications, which provide, respectively, real-time visualizations of experimental data and remote notifications of significant events.

Through the web interface, users can view the progress and results of ongoing experiments at any time and download experimental data files for offline analysis. As shown in Figure 12, users can intuitively monitor experimental progress and data trends via a browser. Experimental data are also available for download in the CSV format, facilitating further detailed analysis. Additionally, users can select specific dates or files through the web interface to generate corresponding line charts, enhancing the flexibility of data visualization.

To ensure that experimenters are always informed about the experimental status, the system employs the *LINE Notify* API for remote notifications. As shown in Figure 13, the system sends real-time notifications to users when the specimen fractures or the experiment concludes. Additionally, the non-English information in the figure indicates that the received time is 01:18. Furthermore, the system supports the scheduled notifications, periodically sending updates on experimental progress and measurement results. The notification intervals can be freely configured by the user, ensuring that experimenters can stay updated on the experimental progress even when they are outside the laboratory.

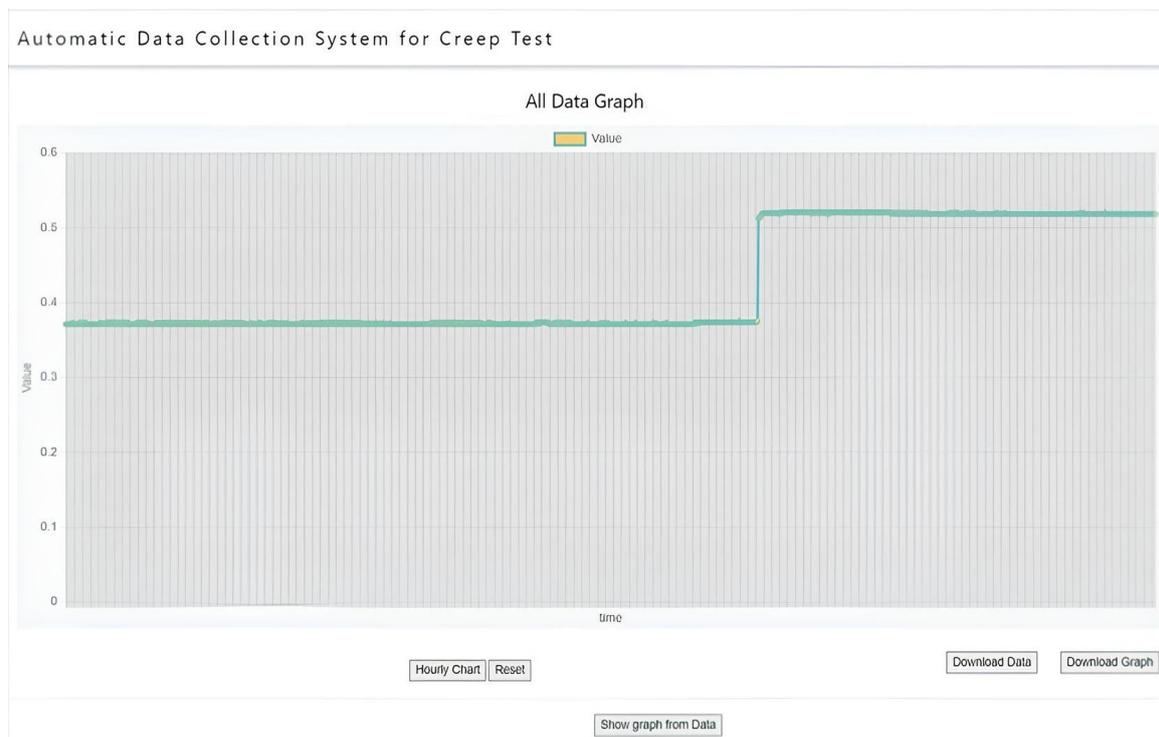


Figure 12. Web interface.



Figure 13. LINE notifications.

5. Experimental Evaluation

In this section, we conduct experiments to evaluate the performance and reliability of the proposed system.

5.1. Experimental Environment

In the experimental setup, we utilize the hardware and software configurations in Table 1.

Table 1. Hardware and software configurations in experiments.

Hardware	Software
Raspberry Pi 4 Model B	Ubuntu 20.04 LTS
Logicool C920N	Python 3
MONOSTICK USB Dongle	OpenCV 4.5.3
Light Sensor	Flask 1.1.2

In our experiments, *Raspberry Pi 4 Model B* equipped with *Logicool C920N* and *MONOSTICK USB Dongle* served as the central processing unit, managing the image processing and IoT functionalities. The software configuration included *Ubuntu* [32] as the operating system, *Python 3* [33] for the development environment, and *OpenCV* [34] for real-time image processing. During the creep test, the system collected measurement data at one-minute intervals from the start of the experiment until the test specimen fractured, enabling real-time fracture detections. The camera was located at the same height as the dial gauge and was placed 20 cm away from it to ensure accurate readings. The experiments took place in the experiment room of #1 Engineering Building at Okayama University, Okayama, Japan.

5.2. Results and Analysis

This section presents the results of two experiments conducted to evaluate the performance of the proposed *creep test assistance system*. These experiments lasted 155 h and 40 min, and 359 h and 4 min, respectively. Both experiments followed a similar procedure, from the material placement to the specimen rupture, with different applied stable pressures resulting in varying rupture durations. The results focus on the system's accuracy, efficiency, robustness, and fracture detection capabilities, substantiated by detailed analysis and statistical evidence.

5.2.1. Experiment 1

During the first experiment, data were collected at one-minute intervals from the dial gauge, yielding a total of 9340 data points. The average measurement error was 0.00103, indicating the high precision for the needle position detection. The smart lighting function was activated 15 times under varying lighting conditions, with a total usage time of 108 h and 55 min, accounting for 69.9% of the total experimental duration.

Figure 14 provides a visual comparison between system measurements and manual readings, demonstrating their strong correlation. The result confirms that the system maintained the consistent accuracy across different lighting environments, further validating

the effectiveness of the smart lighting mechanism. This result also highlights the system's ability to reduce the energy consumption while ensuring the reliable measurement accuracy. By focusing on pre-rupture data, the system demonstrated its capability to monitor and record experimental data with minimal human interventions.

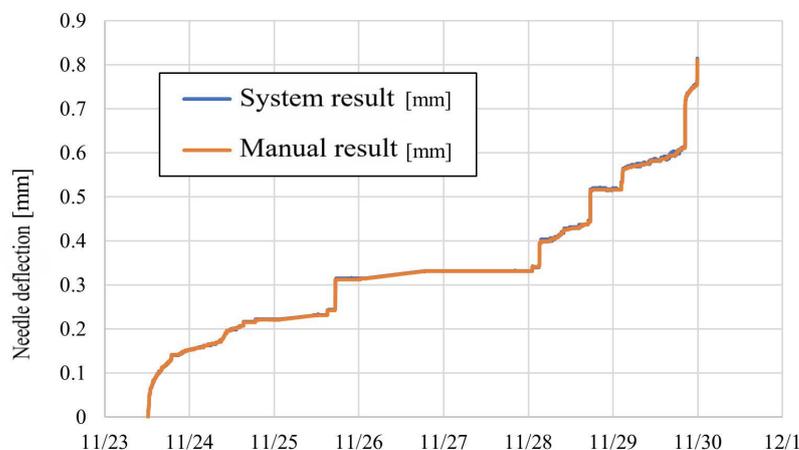


Figure 14. System and manual measurements in Experiment 1.

5.2.2. Experiment 2

In the second experiment, the system operated for 21,544 min, collecting the same number of data points. The average measurement error was 0.00097, indicating further improvement in accuracy, likely due to optimized image processing algorithms and long-term operational stability. The smart lighting function was activated 42 times due to fluctuating environmental conditions, with a total usage time of 251 h and 18 min, accounting for 70.1% of the total experimental duration. Figure 15 shows a comparison between the system and manual measurements, confirming the system's consistent performance. The data reveal the enhanced stability over extended durations, with error trends indicating minimal drifts. The higher activation frequency of the smart lighting mechanism during this longer experiment underscores its dynamic adaptability to environmental changes. These findings highlight the system's robustness in maintaining high accuracy and energy efficiency over prolonged testing periods.

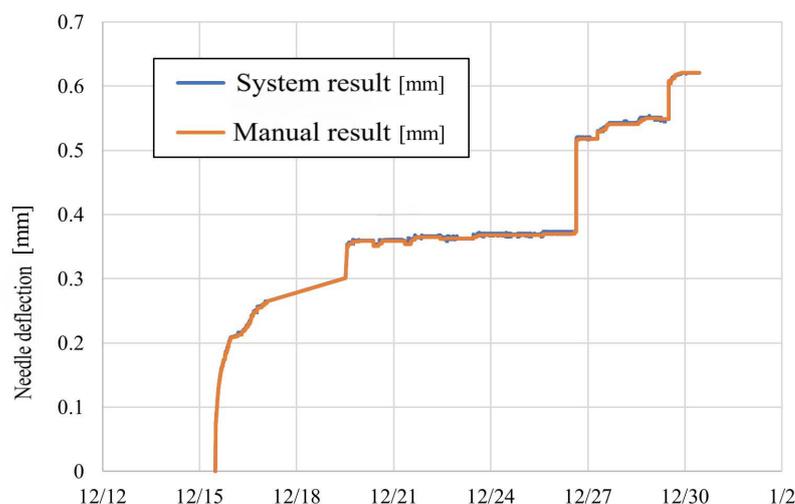


Figure 15. System and manual measurements in Experiment 2.

5.2.3. Fracture Detection

The fracture detection capability of the system was evaluated during the second experiment. Figure 16 illustrates the process where the “rupture state” variable transitioned from 0 (intact material) to 1 (ruptured material). During the test, the marker’s y-coordinate gradually decreased from 120 px to 113 px as the material deformed. Upon rupture, the y-coordinate abruptly dropped to 33 px, triggering an update in the rupture state. This transition activated the *LINE* notification to the laboratory personnel, ensuring prompt awareness of the fracture event. The analysis demonstrates the system’s sensitivity to rapid changes in material conditions and its ability to accurately classify rupture events. Additionally, the automated notification mechanism achieved an average response time of 5 s, further validating the system’s practical applicability in laboratory environments. These results confirm the reliability of the fracture detection method described in Section 4 and highlight its effectiveness in ensuring timely interventions during critical experimental moments.

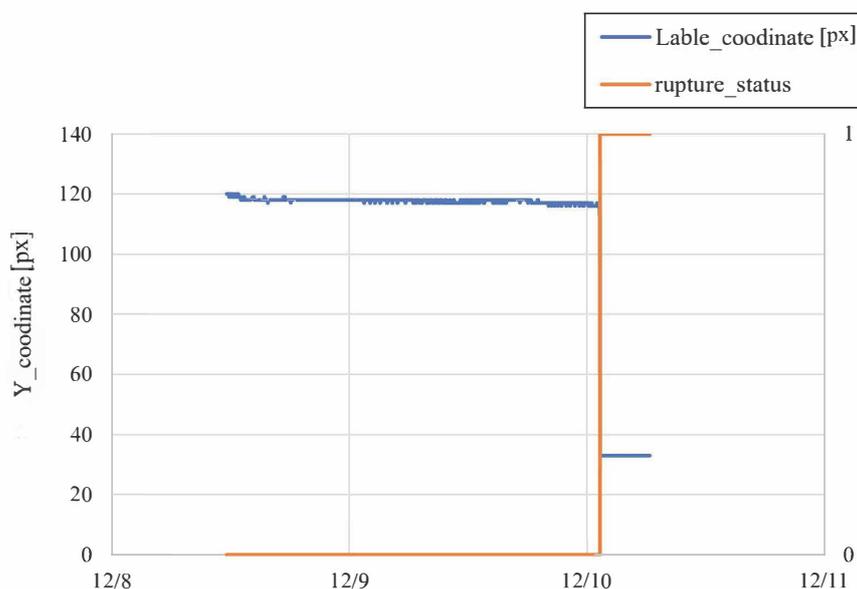


Figure 16. Rupture state in creep testing system.

5.2.4. Summary

The final mean error of 0.00100 reflects the system’s high accuracy across varying experimental durations. Table 2 summarizes the key parameters and results of the experiments.

Table 2. Summary of experimental results.

Experiment	Duration (h:m)	Average Error	Final Mean Error
Experiment 1	155:40	0.00103	0.00100
Experiment 2	359:04	0.00097	

The two experimental results indicate that the *creep test assisting system* maintained an error rate below 0.0011 across both tests, meeting the laboratory precision standard. Data visualizations and statistical analysis results validate the system’s long-term stability and the consistent performance across both short-duration and long-duration tests. In addition, the system introduces functions that enhance its practical applicability: the smart lighting mechanism significantly reduces energy consumptions by lighting only when necessary, showcasing its dynamic responsiveness and energy-saving potential; the material fracture detection function effectively identifies the transition from the intact to ruptured states, with

timely notifications ensuring rapid responses; and the overall system achieves the robust automation performance without relying on costly robotic equipment. The results collectively demonstrate the system's advantages in terms of the accuracy, reliability, energy efficiency, and usability for long-duration experimental monitoring in laboratory environments.

5.3. Discussion

In recent years, deep learning techniques, such as convolutional neural networks, have achieved remarkable results in image recognition tasks and have been successfully applied to gauge reading scenarios. However, our system is deployed on edge devices such as Raspberry Pi, which have limited computational capacity and power resources. Incorporating deep learning models directly may result in significant latency increases and higher energy consumption, which contradicts our design goals for real-time responsiveness and low-power operation. Furthermore, as the system must operate continuously for several days or weeks, sustained high computational loads could lead to system instability. MobileNet and Tiny-YOLO offer promising solutions for edge devices by balancing accuracy and resource usage. As a future direction, we plan to investigate the integration of such models, provided that the real-time performance of the system can be preserved.

6. Conclusions

This paper presented the design and implementation of the *creep test assisting system* to address problems at laborious and error-prone manual monitoring in mechanical engineering laboratories under various light conditions. The experiments at a mechanical engineering laboratory in Okayama University, Japan, showed the real-time needle recognition with an average error of 0.001 mm using image processing techniques. Moreover, the integration with a smart lighting mechanism reduced energy consumption by 30%, ensuring the consistent monitoring accuracy under varied lighting environments. In future works, we will focus on enhancing the system's capability and usability by adopting high-precision equipment, such as infrared cameras, to enable all-weather needle recognitions and by simplifying the system's design to improve the maintainability and ensure the long-term operational efficiency.

Author Contributions: Conceptualization, D.K. and N.F.; methodology, D.K. and N.F.; software, D.K., S.F., P.P. and N.; validation, D.K., N.F. and M.O.; investigation, D.K., S.F. and P.P.; resources, M.O.; writing—original draft, D.K.; writing—review and editing, N.F.; supervision, N.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors thank the reviewers for their thorough reading and helpful comments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Smith, J.; Brown, T.; Wang, L. Analog Dial Gauge Reader for Handheld Devices. In Proceedings of the 2013 IEEE International Conference on Automation Science and Engineering (CASE), Melbourne, Australia, 19–21 June 2013; pp. 305–310. [[CrossRef](#)]
2. Leon-Alcazar, J.; Alnumay, Y.; Zheng, C.; Trigui, H.; Patel, S.; Ghanem, B. Learning to Read Analog Gauges from Synthetic Data. In Proceedings of the 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2024; pp. 8601–8610. [[CrossRef](#)]

3. Chavan, S.; Yu, X.; Saniie, J. High Precision Analog Gauge Reader Using Optical Flow and Computer Vision. In Proceedings of the 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA, 19–21 May 2022; pp. 171–175. [CrossRef]
4. Wikipedia. Creep-Testing Machine. 2025. Available online: https://en.wikipedia.org/wiki/Creep-testing_machine (accessed on 26 January 2025).
5. Fujita, Y.; Hamamoto, Y. Automatic Reading of Analog Meters Using Image Processing. *IEEJ Trans. Electron. Inf. Syst. C* **2009**, *129*, 901–908. (In Japanese)
6. Upton, E.; Halfacree, G. *Raspberry Pi User Guide*, 4th ed.; Wiley: Hoboken, NJ, USA, 2021.
7. Duda, R.; Hart, P. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM* **1972**, *15*, 11–15. [CrossRef]
8. Ziou, D.; Tabbone, S. Edge Detection Techniques—An Overview. *Pattern Recognit. Image Anal.* **1998**, *8*, 537–559.
9. Gonzalez, R.; Woods, R. *Digital Image Processing*, 4th ed.; Pearson: London, UK, 2020.
10. Wang, J.; Huang, J.; Cheng, R. Automatic Reading System for Analog Instruments Based on Computer Vision and Inspection Robot for Power Plant. In Proceedings of the 2018 10th International Conference on Modelling, Identification and Control (ICMIC), Guiyang, China, 2–4 July 2018; pp. 1–6. [CrossRef]
11. Huang, J.; Wang, J.; Tan, Y.; Wu, D.; Cao, Y. An Automatic Analog Instrument Reading System Using Computer Vision and Inspection Robot. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 6322–6335. [CrossRef]
12. Tian, E.; Zhang, H.; Hanafiah, M. A Pointer Location Algorithm for Computer Vision-Based Automatic Reading Recognition of Pointer Gauges. *Open Phys.* **2019**, *17*, 86–92. [CrossRef]
13. Lauridsen, J.; Grassmé, J.; Pedersen, M.; Jensen, D.; Andersen, S.; Moeslund, T. Reading Circular Analogue Gauges using Digital Image Processing. In Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019), Prague, Czech Republic, 25–27 February 2019; pp. 373–382. [CrossRef]
14. Dumberger, S.; Edlinger, R.; Froschauer, R. Autonomous Real-Time Gauge Reading in an Industrial Environment. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Vienna, Austria, 8–11 September 2020; pp. 1–6. [CrossRef]
15. Howells, B.; Charles, J.; Cipolla, R. Real-Time Analogue Gauge Transcription on Mobile Phone. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 19–25 June 2021; pp. 1–6. [CrossRef]
16. Trairattanapa, V.; Phimsiri, S.; Utintu, C.; Cherdchusakulcha, R.; Tosawadi, T.; Thamwivatthana, E.; Tungjitnob, S.; Tangamonsiri, P.; Takutrua, A.; Keomeesuan, A.; et al. Real-Time Multiple Analog Gauges Reader for an Autonomous Robot Application. In Proceedings of the 2022 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Chiang Mai, Thailand, 5–7 November 2022; pp. 1–6.
17. Peixoto, J.; Sousa, J.; Carvalho, R.; Santos, G.; Cardoso, R.; Reis, A. End-to-End Solution for Analog Gauge Monitoring Using Computer Vision in an IoT Platform. *Sensors* **2023**, *23*, 9858. [CrossRef] [PubMed]
18. Smith, J.; Kumar, R.; Lee, T. Automated Monitoring of Analog Gauges Using Computer Vision and IoT Integration. *Int. J. Ind. Inform.* **2018**, *12*, 45–56. [CrossRef]
19. Garcia, L.; Patel, S.; Wong, M. Real-Time Remote Monitoring of Industrial Equipment Using IoT and Machine Vision. In Proceedings of the IEEE International Conference on Industrial Technology, Toronto, ON, Canada, 7–10 March 2017; pp. 112–117. [CrossRef]
20. Chen, H.; Nguyen, D.; Park, J. Development of an IoT-Based System for Remote Monitoring of Analog Meters. *J. Sens. Actuator Netw.* **2019**, *8*, 89–98. [CrossRef]
21. Yang, C.; Zhu, R.; Yu, X.; Yang, J.; Wang, Z. Real-time reading system for pointer meter based on computer vision. *Artif. Intell. Rev.* **2023**, *56*, 145–157. [CrossRef]
22. Barbosa, J.; Graça, R.; Santos, G.; Vasconcelos, M.J.M. Automatic analogue gauge reading using smartphones and computer vision. In Proceedings of the 2023 8th International Conference on IoT and Big Data (IoTBD), Prague, Czech Republic, 21–23 April 2023; pp. 52–61. [CrossRef]
23. Liu, Y.; Liu, J.; Ke, Y. A Detection and Recognition System of Pointer Meters for Industrial Environments with Smart Lighting Integration. *Measurement* **2020**, *152*, 107333. [CrossRef]
24. Research, W. Gaussian Filter—Mathematical Overview. 2013. Available online: <https://reference.wolfram.com/language/ref/GaussianFilter.html> (accessed on 25 January 2025).
25. Saito, F.; Miyajima, T. Pointer Detection for Circular Analog Meters Using Hough Transform. *Precis. Eng.* **2004**, *70*, 219–223. [CrossRef]
26. Wikipedia. RGB Color Model. 2025. Available online: https://en.wikipedia.org/wiki/RGB_color_model (accessed on 25 January 2025).
27. Singh, H. *Practical Machine Learning and Image Processing*; Apress: Berkeley, CA, USA, 2019.

28. Co, M. MONOSTICK PAL Light Sensor User Guide. 2023. Available online: <https://mono-wireless.com/jp/index.html> (accessed on 25 January 2025).
29. Wikipedia. Illuminance. 2025. Available online: <https://en.wikipedia.org/wiki/Illuminance> (accessed on 25 January 2025).
30. Flask. Flask Web Development Framework. 2023. Available online: <https://flask.palletsprojects.com> (accessed on 25 January 2025).
31. LY Corporation. LINE Notify API Documentation. 2020. Available online: <https://notify-bot.line.me/doc/en/> (accessed on 25 January 2025).
32. Canonical Ltd. Ubuntu Official Website. 2025. Available online: <https://ubuntu.com> (accessed on 25 January 2025).
33. Python Software Foundation. Python Documentation, Version 3.x. 2025. Available online: <https://docs.python.org/3> (accessed on 25 January 2025).
34. OpenCV. OpenCV Documentation. 2023. Available online: <https://docs.opencv.org> (accessed on 25 January 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.