

The pointer-type instrument recognition method based on OpenCV

Tianchen Zhu¹, Junchen Bao¹, Xiaosan Zhang¹, Wei Ming¹, Mingju Wang^{1}*

¹Taihe Hospital, Hubei University of Medicine, Shiyan, China

*Corresponding Author. Email: zhutianchen@foxmail.com

Abstract. Pointer-type instruments remain widely used in industrial environments due to their cost-efficiency and reliability. However, manual reading of such instruments is labor-intensive, error-prone, and unsuitable for real-time monitoring. This paper presents a lightweight and interpretable method for automatic pointer-type instrument recognition using classical image processing techniques provided by OpenCV. The proposed approach includes image preprocessing, circular dial detection via the Hough Circle Transform, scale calibration, pointer extraction using probabilistic Hough Line detection, and value computation through geometric analysis. Experimental results on standard analog gauge images demonstrate the method's accuracy in identifying gauge boundaries, calibrating tick marks, and determining the pointer's angular position. The approach enables flexible value mapping and delivers reliable readings under standard lighting conditions. Compared to deep learning methods, this solution offers better computational efficiency and easier integration into resource-constrained systems. It is particularly suitable for industrial automation and retrofitting of legacy analog devices. Future work will focus on improving robustness under complex backgrounds and dynamic conditions.

Keywords: pointer-type instrument, OpenCV, image processing.

1. Introduction

In various industrial and engineering domains, pointer-type instruments have long been a staple for measuring and displaying a wide range of physical quantities, such as pressure, temperature, voltage, and current. Their widespread use can be attributed to several notable advantages, including their cost-effectiveness, robust reliability, and straightforward maintenance requirements. These instruments offer a simple yet effective way to visually represent real-time data, making them an indispensable tool in numerous applications where quick and direct readings are essential.

However, as industries advance and automation becomes increasingly prevalent, the limitations of manual pointer-type instrument reading have become more apparent. Manual reading is not only time-consuming and labor-intensive but also prone to human errors, especially in environments with high-speed processes, hazardous conditions, or a large number of instruments to monitor simultaneously. In addition, in scenarios where continuous and real-time data acquisition is required for further analysis, decision-making, or control purposes, manual reading falls short of meeting the demands of modern industrial systems.

To address these challenges, the development of automated pointer-type instrument recognition methods has gained significant attention in recent years. Among the various approaches available, image recognition technology based on computer vision has emerged as a promising solution. OpenCV, an open-source computer vision library, provides a rich set of tools and algorithms for image processing, feature extraction, and pattern recognition, making it an ideal platform for implementing automated pointer-type instrument recognition systems.

This paper focuses on proposing a pointer-type instrument recognition method based on OpenCV. The primary objective of this research is to develop an efficient and accurate automated system that can replace manual reading, thereby improving the efficiency, reliability, and safety of instrument monitoring processes. By leveraging the powerful capabilities of OpenCV, the proposed method aims to extract key features from pointer-type instrument images, such as the position of the pointer and the scale markings, and then accurately determine the measured value through intelligent image analysis algorithms.

The significance of this research lies in its potential to revolutionize the way pointer-type instruments are utilized in industrial settings. Automated recognition not only reduces the workload on human operators but also enables real-time data transmission and integration with other industrial automation systems, facilitating more efficient process control and decision-making. Furthermore, the proposed method can be easily adapted to different types of pointer-type instruments, making it a versatile and scalable solution for a wide range of applications.

In the following sections of this paper, we will first provide a detailed overview of the related work in the field of pointer-type instrument recognition, highlighting the strengths and limitations of existing methods. Then, we will present the architecture and algorithms of our proposed OpenCV-based recognition method, including image preprocessing, feature extraction, and pointer position detection techniques. Experimental results and performance evaluations will be presented to demonstrate the effectiveness and accuracy of the proposed method. Finally, we will conclude the paper with a discussion of the research findings, potential applications, and future directions for further improvement.

2. Background

In the realm of industrial automation and monitoring, pointer-type instruments play a pivotal role in providing real-time data on various physical parameters such as pressure, temperature, voltage, and current [1]. Despite the advent of digital displays, pointer-type instruments remain prevalent due to their cost-effectiveness, robust reliability, and ease of maintenance. However, the traditional method of manually reading these instruments is labor-intensive, error-prone, and time-consuming, especially in environments where continuous monitoring is required [2].

To address these limitations, researchers have explored various automated recognition methods for pointer-type instruments. These methods can be broadly categorized into two groups: those based on traditional computer vision techniques and those leveraging deep learning algorithms [3].

Traditional Computer Vision Techniques: Early attempts at automated pointer-type instrument recognition relied heavily on traditional computer vision techniques such as Hough Transform for detecting circular dials and pointers, and edge detection algorithms for identifying scale markings. For instance, Sablatnig et al [4]. proposed a method based on the Hough Transform for recognizing arc-shaped meter pointers. Yang et al [5]. utilized shadow reduction and Hough Transform to detect pointers and their centers of rotation in substation pointer-type meters. Although these methods demonstrated some success, they often suffered from low accuracy and sensitivity to environmental conditions such as lighting variations and background noise.

Deep Learning Algorithms: With the advent of deep learning, researchers have turned to convolutional neural networks (CNNs) for more robust and accurate pointer-type instrument recognition [6]. Mask R-CNN, U-Net, and Faster R-CNN are among the popular CNN architectures employed for this purpose [7]. While these deep learning-based methods have achieved higher accuracy, they often require substantial computational resources and are prone to overfitting when trained on limited datasets [8].

OpenCV, an open-source computer vision library, provides a rich set of tools and algorithms for image processing and analysis. It has been widely used in various computer vision applications due to its efficiency, flexibility, and cross-platform compatibility. In the context of pointer-type instrument recognition, OpenCV offers several advantages over traditional deep learning approaches:

Lightweight and Efficient: OpenCV algorithms are typically less computationally intensive compared to deep learning models, making them suitable for real-time applications on embedded systems or low-power devices [9].

Customizable and Flexible: OpenCV provides a wide range of image processing functions that can be easily customized and combined to suit specific application requirements [10]. This flexibility allows researchers to tailor the recognition pipeline to the characteristics of different pointer-type instruments.

Interpretability: Unlike deep learning models, which often operate as black boxes, OpenCV algorithms are based on well-understood image processing principles [11]. This interpretability facilitates debugging, optimization, and integration with existing systems [12].

Given these advantages, this paper proposes an automated pointer-type instrument recognition method based on OpenCV. The method combines image preprocessing, feature extraction, and pointer position detection techniques to accurately extract the instrument's reading from images. By leveraging OpenCV's powerful capabilities, the proposed method aims to achieve high accuracy, robustness, and efficiency in various industrial environments.

3. Method

The proposed method for pointer-type instrument recognition is implemented using OpenCV and consists of several key stages: image preprocessing, gauge detection, scale calibration, pointer detection, and value computation. Each step is designed to ensure the accurate and efficient extraction of the pointer's angle and the corresponding measurement value from the instrument image. The overall workflow is optimized for robustness, computational efficiency, and adaptability to different gauge types.

3.1. Image preprocessing

Image preprocessing is a crucial initial step that significantly affects the performance of all subsequent operations. The process begins by loading the input image in a standard format and converting it into a grayscale image using OpenCV's `cvtColor`

function. Converting to grayscale simplifies the image data by removing color information, which is not essential for detecting geometric features such as circles and lines.

To further enhance the quality of feature extraction, a Gaussian blur filter is applied using the `GaussianBlur` function. This step helps to smooth the image, reduce high-frequency noise, and prevent false detections in the following stages, particularly in circle and edge detection. A carefully chosen kernel size ensures that the blur does not excessively suppress relevant features. These preprocessing steps help isolate essential components such as the gauge boundary, tick marks, and pointer while suppressing irrelevant background details and minor imperfections in lighting or texture.

3.2. Gauge detection using hough transform

After preprocessing, the next step is to locate the circular dial of the instrument using the Hough Circle Transform algorithm provided by OpenCV. This transform is particularly suitable for detecting circular objects within a noisy image by converting the image space into a parameter space and searching for peaks that represent circular patterns.

The blurred grayscale image is passed through the `HoughCircles` function, which returns a set of candidate circles that potentially match the dial's contour. Since multiple circles may be detected due to visual artifacts or reflections, an averaging strategy is applied to determine the most likely center and radius of the gauge. This is done by computing the mean coordinates and radius from all detected circles within a confidence threshold.

These extracted parameters—the center point and radius—form the geometric basis for further operations, including aligning the scale, constraining pointer detection, and calculating angular displacement. The robustness of this step is critical; misidentifying the gauge center or radius would propagate errors throughout the pipeline.

3.3. Scale calibration and annotation

With the circular boundary of the gauge established, the next step involves simulating and calibrating the scale markings. These scale ticks represent discrete measurement intervals (e.g., every 10 degrees) and serve as visual references for both users and the algorithm when interpreting pointer angles.

Using the known center and radius of the dial, a set of radial lines is drawn from the center outward at equal angular intervals, typically ranging from 0° to 360° , depending on the gauge design. These lines represent tick marks and are overlaid onto the image using the `line` function. Optionally, numerical labels can be generated at fixed intervals (e.g., every 10°) using the `putText` function to improve visualization.

This artificial calibration ensures consistency across different gauges, especially when the original image lacks clear or uniformly spaced tick marks. Additionally, this step facilitates later value computation by allowing a direct mapping between pointer angle and measurement values.

3.4. Pointer detection

Pointer detection is performed using the Probabilistic Hough Line Transform, which is well-suited for identifying linear features in binary images. Before this, a thresholding operation is applied to the grayscale image to produce a binary image where the pointer and scale lines are highlighted. An inverse binary threshold is typically used to isolate dark lines (such as the pointer) against a lighter background.

The `HoughLinesP` function is then employed to detect straight lines in the image. To ensure that only the actual pointer is retained, several geometric constraints are introduced. First, the candidate lines are filtered based on their proximity to the center of the gauge—valid pointer lines must originate from or near the center and extend outward toward the edge. Second, their lengths are compared against a defined fraction of the gauge's radius to exclude excessively short or long lines resulting from image noise or background artifacts.

After filtering, the remaining line segments are evaluated based on orientation, consistency, and location. The line that best fits the expected position and orientation of the pointer is selected as the final pointer candidate. This process ensures high reliability in distinguishing the true pointer from visual clutter.

3.5. Angle and value computation

The most probable pointer line is selected from the filtered candidates, and its angle with respect to the gauge center is computed using the arctangent function. This angle is then converted from polar to Cartesian degrees and mapped to the gauge's defined measurement range. The final result is the current value indicated by the pointer, which is then output or stored as needed.

4. Results

To evaluate the effectiveness of the proposed method, a series of pointer-type instrument images were processed using the developed algorithm. The system was tested on images containing circular analog gauges under standard lighting conditions and relatively uncluttered backgrounds. The original image before processing is shown in Figure 1:



Figure 1. Original image of the instrument before processing

4.1. Visualization of gauge detection and calibration

The first output image (Figure 2) shows the result of gauge detection and scale calibration. The algorithm correctly identifies the circular dial and marks scale lines at 10-degree intervals. Numerical values are annotated around the dial, providing visual reference points for interpreting the pointer's orientation.



Figure 2. A Result of gauge detection and scale calibration. The circular dial is correctly identified with scale lines marked at 10-degree intervals, and numerical values are annotated around the dial for reference

4.2. Pointer detection and line filtering

After binarizing and thresholding the grayscale input image, the pointer is detected using the probabilistic Hough Line Transform. The system filters out irrelevant lines by comparing distances from the gauge center to each line's endpoints. Figure 3 shows the result of pointer detection. The green line represents the detected pointer. Its position, length, and direction are consistent with visual inspection of the pointer's location in the original image. This demonstrates the effectiveness of the filtering strategy based on radial constraints.



Figure 3. Result of pointer detection using probabilistic Hough Line Transform. The green line represents the detected pointer, with its position, length, and direction consistent with the original image, demonstrating the effectiveness of the filtering strategy

4.3. Value extraction

Using the angular difference between the detected pointer and the reference zero angle, the program calculates the corresponding measurement value. The user inputs the minimum and maximum angle positions and their associated values.

5. Conclusion

In this study, we developed and implemented an automatic reading algorithm for analog pointer-type gauges using classical image processing techniques. The proposed method leverages circle detection via the Hough Transform for gauge localization, followed by tick mark generation and pointer detection through probabilistic line extraction. The algorithm demonstrates robustness in detecting the gauge boundary, calibrating angular positions, and accurately locating the pointer.

Experimental results show that the system is capable of extracting numerical readings from images of mechanical gauges with high visual clarity. The calibration process allows flexible mapping between angular displacements and actual measurement values, making the method adaptable to different instrument types and scales. Furthermore, by applying geometric filtering constraints on the detected lines, the algorithm effectively isolates the valid pointer from extraneous visual elements.

This approach offers a low-complexity, low-cost alternative to more complex deep learning-based methods, making it suitable for scenarios where computational efficiency and interpretability are critical. While current implementation assumes a clean background and well-illuminated gauges, future work will explore enhancements such as adaptive thresholding, pointer color segmentation, and integration with real-time video streams.

In conclusion, the system provides a reliable and interpretable solution for automated gauge reading, with potential applications in industrial monitoring, smart inspection, and retrofitting of legacy analog devices.

References

- [1] Lin, Y., Zhong, Q., Sun, H.: A pointer type instrument intelligent reading system design based on convolutional neural networks. *Front. Phys.* 8, 618917 (2020).

- [2] Huo, F., Li, M., Zhang, Y., Zhao, Z., Wang, L.: New identification method of linear pointer instrument. *Multimed. Tools Appl.* 82(3), 4319–4342 (2023).
- [3] Lai, Y.: A comparison of traditional machine learning and deep learning in image recognition. In: *J. Phys.: Conf. Ser.*, vol. 1314(1), 012180. IOP Publishing, Bristol (2019).
- [4] Sablatnig, R., Kropatsch, W.G.: Automatic reading of analog display instruments. In: *Proc. 12th Int. Conf. on Pattern Recognition*, vol. 1, pp. 578–580. IEEE, Los Alamitos (1994).
- [5] Yang, X., Ma, S.: An automatic reading recognition method for pointer spring tube pressure instrument. In: *2019 Chinese Control And Decision Conference (CCDC)*, pp. 3296–3301. IEEE, Nanchang (2019).
- [6] Paluru, N., Agarwal, H., Tripathi, S., et al.: Anam-net: Anamorphic depth embedding-based lightweight CNN for segmentation of anomalies in COVID-19 chest CT images. *IEEE Trans. Neural Netw. Learn. Syst.* 32(3), 932–946 (2021).
- [7] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 2961–2969. IEEE, Venice (2017).
- [8] Bejani, M.M., Ghatee, M.: A systematic review on overfitting control in shallow and deep neural networks. *Artif. Intell. Rev.* 54(8), 6391–6438 (2021).
- [9] García, G.B., García, R.L., García, R.R.: *Learning Image Processing with OpenCV*. Packt Publishing, Birmingham (2015).
- [10] Shilkrot, R., Escriva, D.M.: *Mastering OpenCV 4: A Comprehensive Guide to Building Computer Vision and Image Processing Applications with C++*. Packt Publishing, Birmingham (2018).
- [11] Kawakura, S., Yoshimura, Y., Otomo, K., Fujita, K.: Visual analysis of agricultural workers using explainable artificial intelligence (XAI) on class activation map (CAM) with characteristic point data output from OpenCV-based analysis. *Eur. J. Artif. Intell. Mach. Learn.* 2(1), 1–8 (2023).
- [12] Vieira, R., Silva, T., Almeida, J., Santos, J., Barbosa, J.: Performance evaluation of computer vision algorithms in a programmable logic controller: An industrial case study. *Sensors* 24(3), 843 (2024).