Automated Water Level Measurement Using Intelligent Image Processing Of Gauge Scales

Shoudong Zhou[1] and Fubao Yang[2*]

[1] School of Electrical Engineering, Anhui Technical College of Industry and Economy, Hefei 230051, China
[2] Anhui Provincial Water Resources and Hydropower Survey, Design and Research Institute Co., Ltd, Hefei 230088, China
[*] Corresponding author. E-mail: Fubaoyang2@outlook.com

Received: May 05, 2026; Accepted: Jul. 29, 2026

Water level monitoring of rivers, water reservoirs, and dams plays an important role in flood prediction, water resource management, and environmental safety. The conventional water level monitoring techniques include manual observation and sensor-based techniques. The conventional image-based and deep learning-based water level measurement techniques have improved accuracy of the water level measurement system. The dataset used in this research work is the Water Gauge Computer Vision Dataset available in Roboflow Universe, which consists of about 500 real-world water gauge images taken under varying environmental conditions. The proposed water level measurement system differs from the conventional techniques in terms of the usage of YOLOv8-based gauge region detection, hybrid Convolutional Neural Network (CNN)-Optical Character Recognition (OCR) for digit detection, and classical computer vision techniques like edge detection and Hough transform for accurate waterline detection. The experimental results show high detection accuracy with Precision 0.960, Recall 0.955, F1-score 0.958, Accuracy 0.969, and mAP 0.970. The system has high estimation accuracy as well, as shown by MAE 0.036 m, RMSE 0.081 m and MAPE 3.55%, which shows minimal prediction error in the results.

Keywords: Automated Water Level Monitoring, YOLOv8, Gauge Scale Detection, Computer Vision, CNN–OCR, Image Processing

http://dx.doi.org/10.6180/jase.202612_35.031

1. Introduction

Water level monitoring in rivers, reservoirs, and dams is vital for flood prediction, efficient management of water resources, and the prevention of environmental degradation. It is important to ensure the precise monitoring of the water level to make effective decisions by the concerned authorities in response to extreme weather conditions and natural calamities [1]. At present, computer vision techniques and artificial intelligence are found to be effective tools for the efficient monitoring of the water level through the automated process of capturing images from surveillance cameras [2]. The computer vision technique is an efficient approach for the implementation of hydrological monitoring. The proposed intelligent water level monitoring framework supports sustainable agricultural by enabling reliable, non-contact monitoring for efficient water resource management and automated operational systems. The intelligent image processing techniques are also used for the efficient estimation of the water level [3].

Some of the existing research studies have also proposed different approaches for image processing that could be used estimate water level-based images captured by water gauges. The research studies have also proposed Canny edge detection and Hough transform approaches to estimate water level and water gauges based on images captured in rivers [4]. The recent research studies have proposed the use of ANN, CNN and image segmentation approaches to estimate water level based on images captured by water gauges [5].

Contributions

- ✓ Design a framework for automated water level measurement using images of gauge scales through computer vision.
- ✓ Integrate YOLOv8 object detection, image processing, and a CNN-OCR model for accurate gauge object detection.
- ✓ Implement edge detection and Hough Transform techniques accurate waterline detection and water level value estimation.
- ✓ Evaluate the proposed system using various performance measures to show its accuracy and efficiency in water level estimation.

1.1. Problem Statement

Researchers employed the use of the Transformer-based model with the LSTM technique prediction water level. Despite model is effective prediction of long-term dependencies, large amount of data required by the model is not always available in the monitoring stations [6]. The temporal appearance of the water also posed instability in the system. The models required large-scale video datasets for training purposes [7]. The limitations of the current methods indicate a need to design a more efficient and reliable method of water level measurement, which can perform well even with limited information. There is a strong need to design models that reduce complexity, ensuring higher accuracy.

2. Literature survey

Blanch et al. [8] presented research on use of image-based method for automated monitoring of water level in real-time. The authors of the research carried out a long-term application case for assessing the robustness of the presented method. The research focused on the use of computer vision precise identification of water gauge levels. Guo et al. [9] developed an image recognition-based on approach for obtaining water gauge scales in a tidal well environment. The approach utilized image preprocessing and pattern recognition to identify gauge markings. Lee and Jung [10] have developed high-resolution and low-cost image-based water level sensor by implementing high-end image processing algorithms. The method involves the application of digital image processing techniques for accurate measurement of water levels.

Krabbenhoft et al. [11] measured placement bias of global river gauge network to understand the spatial distribution of the monitoring system. The authors analyzed large-scale hydrological data. The results of the research revealed that the river gauges were not distributed uniformly. Muhadi et al. [12] developed a water level monitoring system using deep learning techniques with LiDAR data and surveillance camera images. The research applied data fusion techniques improve accuracy of water level measurement in complex environmental conditions.

From the above literature, it can be observed that many researchers have contributed to the field of automated water level monitoring using techniques such as CNN, ANN, image segmentation using techniques such as U-Net, FCN, object detection using the YOLO algorithm, and Transformer LSTM and Conv LSTM-based models for prediction.

3. Methodology

The methodology for implementing the water level measurement techniques that are proposed for automation, as depicted in Fig. 1 involves application of deep learning and image processing techniques Water Gauge Computer Vision Dataset. Input images are preprocessed and analyzed using YOLOv8 to detect the gauge scale and water level. The proposed methodology supports sustainable agricultural by providing automated vision-based water level monitoring for efficient water resource management. Edge and contour extraction enhance the detected region, followed by hybrid CNN–OCR digit recognition.

3.1. Data Preprocessing

Preprocessing improves detection robustness through Gaussian filtering, histogram equalization, normalization, and image resizing. These techniques reduce noise, enhance contrast, standardize pixel intensities, and optimize image dimensions for deep learning models. Data augmentation, including flipping, rotation, scaling, translation, and brightness adjustment, increases dataset diversity, reduces overfitting, and improves object detection, digit recognition, and waterline estimation under real-world conditions.

3.1.1. Image Resizing

Image resizing standardizes all input images to a fixed dimension required by machine learning and deep learning models. The process of resizing an image is expressed by the following mathematical expression Eq. (1):

$$S_x = \frac{W'}{W}, S_y = \frac{H'}{H} \tag{1}$$

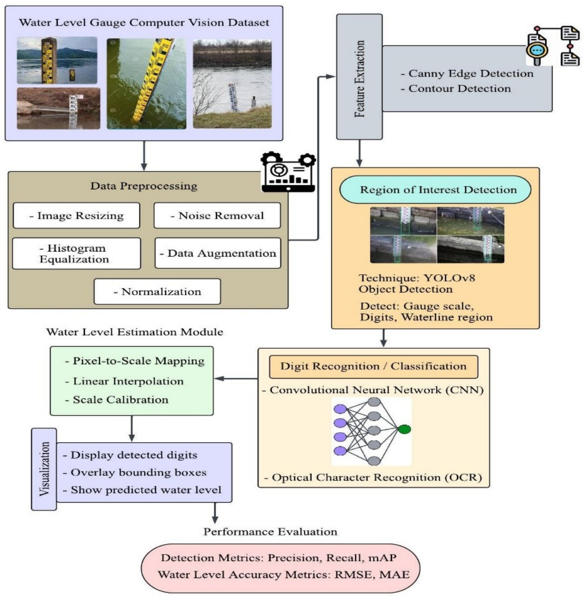


Fig. 1. Architecture of the Proposed Automated Water Level Measurement System Using YOLOv8 and CNN–OCR.

3.1.2. Noise Removal (Gaussian Filter)

Gaussian filtering removes noise caused by lighting variations, reflections, camera sensors, and environmental disturbances while preserving structural details. The mathematical representation of a Gaussian Filter is given by Eq. (2):

$$\mathrm{G(x,y)} = \frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{\mathrm{x^2+y^2}}{2\sigma^2}} \tag{2}$$

where:x and y represent distance of neighboring pixel from center pixel,σ standard deviation that controls amount of smoothing,e is the exponential constant.

3.1.3. Grayscale Conversion

Grayscale conversion transforms RGB images into a single-channel intensity image, suppressing color-related noise and standardizing the input, thereby improving edge detection, digit analysis, and subsequent image processing under varying illumination conditions. The conversion is expressed by Eq. (3):

$$I_{\text{gray}}\,(x,y) = 0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y) \tag{3}$$

where R, G, and B are the red, green, and blue channel intensities at pixel $(\,x,y\,)$.

3.1.4. Histogram Equalization

Histogram equalization enhances image contrast by redistributing pixel intensities, improving the visibility of gauge scales, digits, and waterline boundaries. The cumulative distribution function used for equalization is given by Eq. (4):

$$I_{eq}(x,y) = (L-1) \times \sum_{k=0}^{I(x,y)} \frac{n_k}{N}(\sigma) \tag{4}$$

where L is the number of intensity levels, n_k is the number of pixels with intensity k, and N is the total number of pixels in the image.

3.1.5. Normalization

Normalization rescales pixel intensity values to a standard range, typically [0,1], to ensure consistent input scale across images captured under different exposure and lighting conditions. Normalization is expressed by Eq. (5):

$$I_{\text{morm}}(x,y) - \frac{I(x,y) - I_{\min}}{I_{\max} - I_{\min}} \tag{5}$$

where $I_{\min}$ and $I_{\max}$ are the minimum and maximum intensity values in the image.

3.2. Feature Extraction

Feature extraction highlights structural information from detected gauge regions using Canny edge detection, contour detection, bounding box extraction, and shape–position analysis.

3.2.1. Canny Edge Detection

The Canny Edge Detection method identifies edges in the gauge scale and waterline regions. The gradient magnitude of images is found using the following Eq. (6):

$$M_g = \sqrt{\mathrm{G_x^2} + \mathrm{G_y^2}} \tag{6}$$

where G_x and G_y are gradients in the horizontal and vertical images, respectively.

3.2.2. Contour Detection

Contour Detection is used to detect continuous boundaries around the detected objects. The contour area can be calculated as Eq. (7):

$$\mathrm{A} = \sum_{\mathrm{i=1}}^{\mathrm{n}} (\mathrm{x_i y_{i+1} - x_{i+1} y_i}) \tag{7}$$

where $(\,\mathrm{x_i, y_i}\,)$ represents contour coordinates.

3.3. Region of Interest Detection Using YOLOv8

Region of Interest Detection **is one of the** important steps in the proposed water level measurement system. These important regions include the scale appearing in the image, the digit appearing in the image ranging from 0 to 9, and the waterline region. Bounding box representation given by following Eq. (8):

$$B = (x, y, w, h) \tag{8}$$

where x and y are coordinates center of bounding box, and w and h are width and height of detected object.

The architecture in Fig. 2 comprises three components: the Backbone, which extracts multi-scale features using convolutional layers, the C2fGhost layer for efficient feature enhancement, and the SPPF layer; the Head, which performs final object detection through the Detect layer.

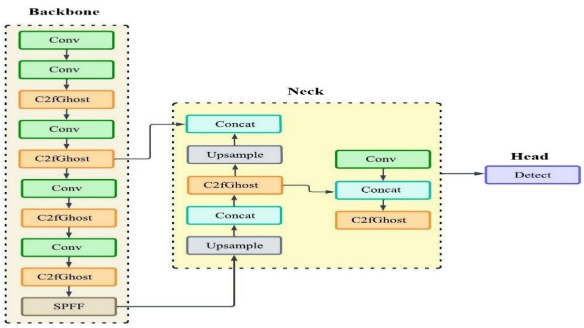


Fig. 2. Architecture of YOLOv8.

3.4. Digit Recognition

Digit recognition is the process by which the identified digit is extracted using machine learning techniques and computer vision. In the proposed system, digit recognition is carried out after the ROI is detected to obtain the actual numerical values from the water gauge.

3.4.1. *Convolutional Neural Network (CNN)*

The Convolutional Neural Network (CNN) recognizes and classifies gauge-scale digits by automatically learning hierarchical features such as edges, textures, and shapes through convolution.

CNN model structure in Fig. 3 starts with an input image, which undergoes multiple convolutions and ReLU layers to learn essential features like edge, shape, and texture. The CNN module for digit classification consists of three convolutional blocks. Conv1 (32 filters, 3×3, ReLU) followed by 2×2 max pooling extracts low-level edge and stroke features from the 32×32 grayscale digit crop. Conv2 (64 filters, 3×3, ReLU) with pooling captures curvature and shape patterns that distinguish visually similar digits (e.g., 3 vs. 8). Conv3 (128 filters, 3×3, ReLU) with pooling encodes higher-level digit-specific representations. The resulting feature map is flattened and passed through a fully connected layer (128 units, ReLU) with dropout (0.3) to reduce overfitting, followed by a softmax output layer producing class probabilities for digits 0-9.

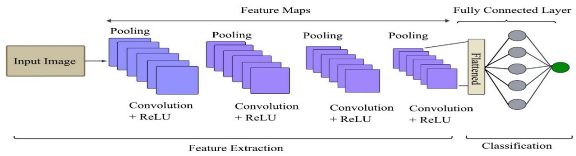


Fig. 3. Convolutional Neural Network Architecture.

3.4.2. *Optical Character Recognition for Digit Recognition*

Optical Character Recognition (OCR) serves as a second-level recognition process to extract machine-readable numerical values from gauge scale images. The thresholding function is generally considered to be Eq. (9):

$$I_b(x, y) = \begin{cases} 1, & \text{if } I(x, y) > T \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where T is threshold value. The recognition process is generally based on template matching. The recognition value is considered to be Eq. (10):

$$S_{ocr} = \sum \left(I_{input} \cdot I_{template} \right) \tag{10}$$

where I_{input} is input digit and $I_{template}$ is stored template.

3.5. Hybrid CNN–OCR Model for Digit Recognition

The hybrid model of CNN-OCR combines the power of deep learning and traditional OCR methods to improve the accuracy of digit recognition. In the proposed hybrid CNN–OCR framework, the detected digit regions extracted from the YOLOv8-based Region of Interest are first resized and normalized before being processed by the CNN model. The CNN acts as the primary classifier, producing a softmax confidence score for each predicted digit. If this confidence exceeds 0.85, the CNN prediction is accepted directly.

Hybrid Convolutional Neural Network – Optical Character Recognition, as presented in Fig. 4 combines deep learning techniques with traditional text recognition techniques to effectively identify and interpret digits in images captured by water gauges.

3.6. Waterline Detection

Waterline detection identifies the horizontal water surface after ROI and digit detection. Image processing techniques locate the waterline and determine its pixel position, enabling accurate estimation of the corresponding water level on the gauge scale.

3.6.1. *Edge Detection*

Edge detection identifies abrupt intensity changes to locate waterline boundaries, gauge scale edges, and digit contours.The gradients in both directions are obtained using

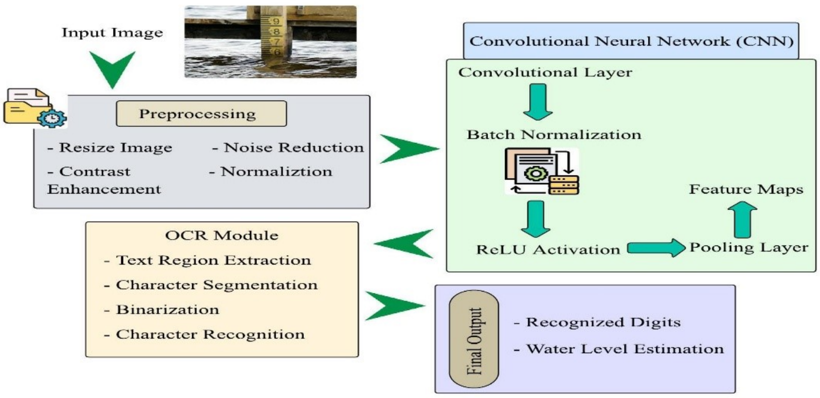


Fig. 4. Architecture of Hybrid Convolutional Neural Network– Optical Character Recognition.

partial derivatives of image intensity function I(x,y)given in Eq. (11):

$$G_x = \frac{\partial I}{\partial x}, G_y = \frac{\partial I}{\partial y} \tag{11}$$

3.6.2. *Hough Line Transform*

The Hough Line Transform identifies the waterline after edge detection by mapping image space to parameter space, enabling unique line representation. Eq. (12) represents the equation for the Hough Transform, which is the representation of the line in the Cartesian coordinate system with slope-intercept form:

$$y = mx + c \tag{12}$$

where m is slope and c are y-intercept. The proposed waterline identification method combines Canny edge detection, contour extraction, and the Hough Transform for robust performance under varying outdoor conditions. Canny edge detection extracts gauge and water surface boundaries while suppressing noise.

3.6.3. *Pixel-to-Scale Mapping and Water Level Estimation*

After detecting the waterline, its vertical pixel coordinate is converted into the corresponding physical water level using a linear pixel-to-scale mapping. The estimated water level is computed in Eq. (13):

$$L - L_{\min} + \frac{(y_w - y_{\min})}{(y_{\max} - y_{\min})}(L_{\max} - L_{\min}) \tag{13}$$

where L is the estimated water level (m), y_w is the detected waterline pixel coordinate, $y_{\min}$ and $y_{\max}$ are the calibrated pixel coordinates of the gauge, and $L_{\min}$ and $L_{\max}$ represent the corresponding minimum and maximum water levels. The three modules form a connected pipeline: YOLOv8 localizes gauge, digit, and waterline regions; CNN-OCR recognizes digits as calibration points; edge detection with Hough Transform localizes the waterline.

3.7. Visualization

Visualization represents the detection and prediction results by displaying identified digits, detected regions, and predicted water level on the processed image, with digits highlighted at their respective positions for verification.

4. Experimental setup

4.1. Performance Evaluation

Water Level Accuracy Metrics assess prediction accuracy by comparing calculated water levels, derived from pixel-to-scale mapping, against ground truth values. RMSE can be expressed mathematically by Eq. (14):

$$RMSE = \sqrt{\frac{1}{n}\sum(L_i - \hat{L}_i)^2} \tag{14}$$

Where: L_i represents actual water level value, $\hat{L}_i$ represents predicted water level value, and n represents total number of observations. Unlike RMSE, MAE considers all errors equally without considering squared values. It is given by Eq. (15):

$$MAE = \frac{1}{n}\sum|L_i - \hat{L}_i| \tag{15}$$

Where: L_i is actual water level value and, $\hat{L}_i$ is predicted water level value.

4.2. Dataset Description

The Water Gauge Computer Vision Dataset [13] provided by Roboflow Universe is intended for use in computer vision-based water level monitoring systems. The dataset includes around 500 images and is thus useful for training and testing deep learning models. The Water Gauge

Computer Vision Dataset comprises approximately 500 real-world images annotated with instance segmentation and bounding boxes across multiple classes digits (0–9), gauge scale, and waterline collected under diverse outdoor conditions including lighting variation, reflections, shadows, and viewpoint changes. Preprocessing steps such as resizing, Gaussian filtering, normalization, and data augmentation.

4.3. Software Configuration

The software configuration for the implementation of the proposed system is a Windows 11 OS, 64-bit version, running on an x64-based architecture. The system has a 12th Gen Intel Core i5-12400 processor, running at a speed of 2.50 GHz. This processor is efficient for carrying out computations. The system has 8 GB of RAM, of which 7.75 GB is usable. The implementation of the system is done by using Python version 3.13.7. This allows for the use of other libraries such as OpenCV, TensorFlow, and PyTorch.

4.4. Hyperparameter Configuration

The system applies a structured hyperparameter pipeline covering image preprocessing (224×224 resizing, percentile-based pixel calibration, Gaussian blur), edge and line detection, and YOLO bounding box handling for ROI extraction. The training process was performed for 100 epochs using a batch size of 16 to achieve stable convergence. During inference, a confidence threshold of 0.25 was applied to retain only reliable object detections, while an Intersection over Union (IoU) threshold of 0.50 was used during Non-Maximum Suppression (NMS) to eliminate overlapping bounding boxes and improve localization accuracy. A learning rate of 0.001, batch size of 16, and 100 training epochs provided the best balance between convergence speed and detection accuracy.

5. Results

The results show the efficiency of the proposed framework for the automated monitoring of the water level, where the gauge region, the waterline and the scale markings can be effectively recognized from the images.

5.1. Performance Analysis of Proposed Work

5.1.1. *Image Preprocessing*

The Preprocessed Image is the representation of the output of the data preprocessing stage for the raw water gauge image shown in Fig. 5. The noise and unwanted changes in the image are removed by applying filters such as Gaussian filtering results in a smoother image.

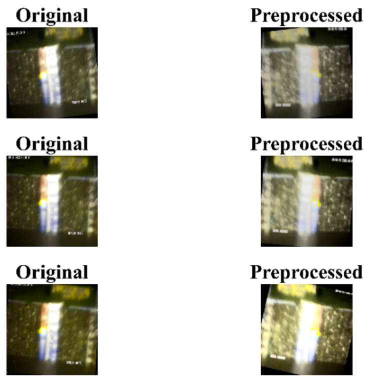


Fig. 5. Preprocessed Image.

5.2. Feature Extraction & Waterline Detection

Process of Edge Detection and Waterline Identification Using the Water Gauge Images. First, the images are converted to grayscale and undergo the Canny Edge Detection algorithm, as depicted in Table 1.

5.2.1. *ROI Detection Using YOLOv8*

Region of Interest (ROI) detection stage of the proposed system YOLOv8 model is employed to locate the important objects in the water gauge image, as depicted in Fig. 6.

Table 2 presents the robustness evaluation of the proposed YOLOv8-based water level measurement framework under different environmental conditions. The results demonstrate consistently high detection performance across varying lighting, reflection, and noise levels.

5.2.2. *Water Level Prediction*

Comparison between the actual water level and the predicted water level by the suggested system is shown in Fig. 7. It can be observed that the scattered points are very close to the ideal line $y = x$.

5.2.3. *Detection Performance Metrics*

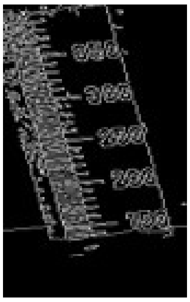
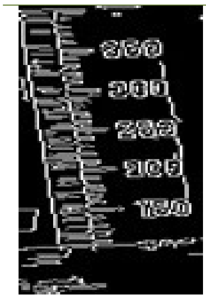
Performance of the detection of the proposed model on five different parameters is shown in Fig. 8. All the performance values are high, ranging from 0.955 to 0.970.

5.2.4. *Water Level Accuracy Metrics*

Water level prediction of accuracy metrics for four different statistical measures is represented in Fig. 9. The system achieves low Mean Absolute Error and Root Mean Squared Error with a high R^2 indicating strong correlation between predicted and actual water levels.

Table 3 shows that the proposed framework maintains consistent prediction accuracy across all water level ranges. The MAE and RMSE remain low, while the mean residuals

Table 1. Edge & Waterline Detection.

Edge Detection	Contour / Waterline Detection
	
	
	

stay close to zero, indicating stable and unbiased estimation.

5.2.5. *Residual Error Analysis*

The pattern of the residual error over 1000 image indices values varies from -0.2m to +0.3m, is shown in Fig. 10.

5.2.6. *Performance Evaluation Metrics*

Performance evaluation results shown in Table 4 clearly prove the effectiveness of the proposed system for both detection and water level estimation purposes.

The obtained detection and estimation metrics demonstrate a strong relationship in the context of automated hydrological monitoring. Higher detection metrics, including Precision (0.960), Recall (0.955), F1-score (0.958), and mAP (0.970), indicate that the proposed YOLOv8-based framework accurately detects the gauge scale. Table 5 presents a comparative evaluation of the proposed system against five existing water level measurement approaches, categorized into three algorithmic paradigms lightweight object

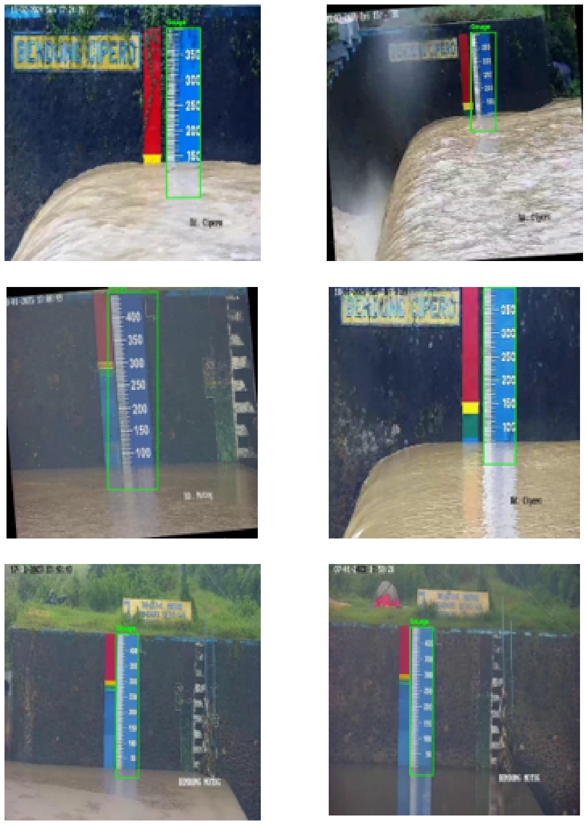


Fig. 6. ROI Detection.

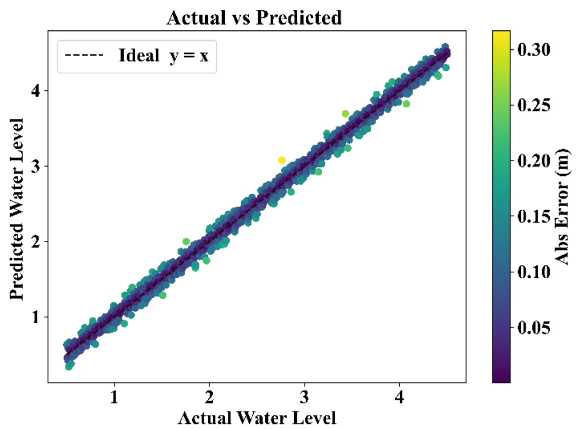


Fig. 7. Actual vs Predicted Scatter.

detection, segmentation-based to benchmark performance across diverse computer vision strategies.

Table 6 compares the computational efficiency of the proposed framework with existing models. The proposed YOLOv8 framework achieves a balanced trade-off between detection accuracy and computational cost, requiring only 3.2 M parameters, 26.3 ms inference time per image (38

Table 2. Robustness Evaluation of the Proposed YOLOv8-Based Water Level Measurement System Under Different Environmental Conditions.

Environmental Condition	No. of Images	Precision	Recall	F1-score	mAP@0.5	MAE (m)
Normal lighting	120	0.971	0.966	0.968	0.978	0.028
Low lighting	90	0.948	0.941	0.944	0.952	0.043
Strong reflection	85	0.936	0.928	0.932	0.941	0.052
Image noise / blur	80	0.929	0.921	0.925	0.934	0.058
Mixed outdoor conditions	125	0.953	0.947	0.95	0.96	0.039

Table 3. Robustness Evaluation of the Proposed YOLOv8-Based Water Level Measurement System Under Different Environmental Conditions.

Water Level Range (m)	Samples	MAE (m)	RMSE (m)	Mean Residual (m)	Std. Residual (m)
0-1	180	0.031	0.067	0	0.029
1-2	210	0.034	0.074	-0	0.032
2-3	240	0.036	0.081	0	0.035
3-4	200	0.039	0.086	-0	0.038
4-5	170	0.044	0.094	0	0.041
Overall	1000	0.036	0.081	0	0.035

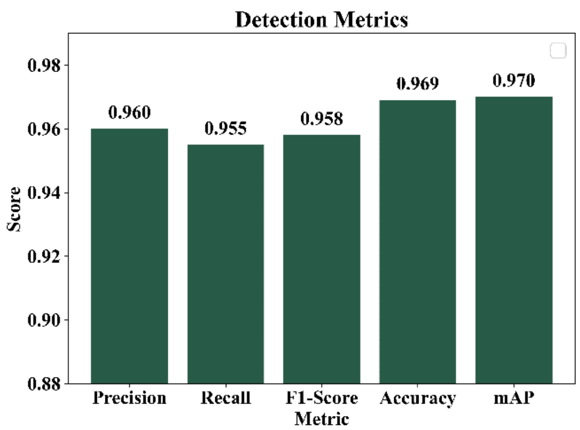


Fig. 8. Detection Metrics.

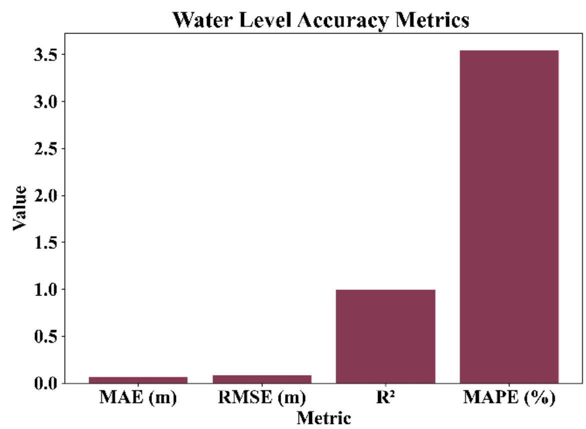


Fig. 9. Water Level Accuracy Metrics.

Table 4. Performance Evaluation Metrics of the Proposed System.

Category	Metric	Value
Detection Metrics	Precision	0.960
	Recall	0.955
	F1-Score	0.958
	Accuracy	0.969
	mAP	0.970
Water Level Accuracy Metrics	MAE (m)	0.036
	RMSE (m)	0.081
	R^2 Score	0.995
	MAPE (%)	3.55

FPS), and 1.8 GB RAM.

5.3. Ablation Study

The ablation study in **??** reveals that there is a performance improvement by progressively adding each of the modules to the system.

The ablation result reveals the distinct contribution of each module. Preprocessing improves precision from 0.82 to 0.88 and cuts MAPE from 12.5% to 8.2%, confirming that noise removal and contrast enhancement sharpen digit and waterline clarity before detection. YOLOv8 ROI localization gives the largest detection gain, raising mAP to 0.94 and reducing MAE to 0.10 m by eliminating background interference.

Table 5. Performance Evaluation Metrics of the Proposed System.

Author	Method / Model	Model Category	Precision	Recall	mAP / Accuracy	MAE (m)	RMSE (m)
Bai et al. [14]	SSD-based Method	Lightweight Object Detection	0.91	0.89	0.92 (mAP)	0.12	0.15
Liang et al. [15]	Mask R-CNN + Faster RCNN	Segmentationbased	0.93	0.92	0.94 (Accuracy)	0.10	0.13
Wang et al. [16]	CNN-based Recognition	Segmentationbased	0.92	0.91	0.93 (Accuracy)	0.09	0.11
Qiu et al. [17]	Dual Attention + Transformer	Transformerbased	0.94	0.93	0.95 (mAP)	0.08	0.10
Proposed Method	YOLOv8 + CNN-OCR	Hybrid	0.960	0.955	0.970 (mAP)	0.036	0.081

Table 6. Computational Performance Comparison of the Proposed YOLOv8 Framework with Representative Deep Learning Models.

Model	Parameters (M)	Inference Time (ms)	FPS	RAM (GB)	Hardware
ViT Detector	85.4	71.4	14	5.8	GPU
Swin Transformer	49.6	55.6	18	4.9	GPU
YOLOv5n	1.9	23.8	42	1.6	CPU/GPU
Proposed YOLOv8 Framework	3.2	26.3	38	1.8	CPU/GPU

Fig. 10. Residual Error.

6. Discussion

The results confirm effective detection and estimation performance. YOLOv8 accurately localizes gauge components despite illumination and reflection variations, while the hybrid CNN-OCR model ensures reliable digit recognition. The proposed approach supports sustainable water resource monitoring for agricultural and industrial systems. However, the system's performance may degrade under challenging real-world conditions such as water surface reflections, partial occlusion of the gauge scale, uneven shading, blurry or worn digit markings, and low-visibility conditions like fog or nighttime imaging.

7. Conclusion and future enhancement

The proposed computer vision framework enables accurate, automated, and cost-effective non-contact water level monitoring using YOLOv8, edge-based waterline detection, and hybrid CNN-OCR digit recognition. The proposed framework can support and industrial water management by providing reliable, automated water level monitoring for reservoirs, irrigation canals, and water storage infrastructure. The proposed framework maintains moderate computational complexity by limiting intensive computations to YOLOv8-based ROI detection and CNN-based digit recognition, while using lightweight image processing techniques for subsequent stages.

The proposed framework can be extended to multi-camera river monitoring networks for synchronized water level tracking and integration with large-scale hydrological systems. Real-time data transmission can support automated flood early-warning applications. Future work focuses on improving robustness under challenging conditions, including reflections, occlusions, uneven illumination, blurred digit markings, fog, and nighttime imaging through enhanced preprocessing and training strategies.

Declarations

Data Availability

https://universe.roboflow.com/watermeasurement/water-gauge-t7rn3

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Table 7. Ablation Table.

Model Variant	Description	Precision	Recall	F1-Score	Accuracy	mAP	MAE (m)	RMSE (m)	R^2	MAPE (%)
Baseline	No preprocessing	0.82	0.80	0.81	0.78	0.83	0.21	0.26	0.91	12.5
Preprocessing	Resize + Filter	0.88	0.86	0.87	0.85	0.89	0.15	0.19	0.95	8.2
YOLOv8 Detection	ROI detection added	0.93	0.92	0.92	0.90	0.94	0.10	0.13	0.98	5.6
Feature Extraction	Edge + Contour	0.95	0.94	0.94	0.92	0.96	0.08	0.10	0.99	4.2
Final Intelligent Estimation Model	YOLOv8 + Waterline Detection (Edge-based)	0.96	0.955	0.958	0.967	0.97	0.036	0.081	0.995	3.55

Funding Statement

This work was supported by the National Key Research and Development Program Project (Grant No. 2021YFC3000105) and the Major Natural Science Research Project of Anhui Provincial Universities (Grant No. 2022AH040337).

Author Contribution

Shoudong Zhou contributed to the conceptualization, methodology development, implementation of the intelligent image processing framework, data analysis, and preparation of the manuscript.

Fubao Yang supervised the research work, contributed to the study design, validation of the results, and manuscript review and editing.

Ethical Approval

This study does not involve human participants or animals. All data used in this research were obtained from publicly available datasets; therefore, ethical approval was not required.

Consent to Participate

Not applicable.

Consent to Publication

The authors consent to the publication of this manuscript.

Competing Interests

The authors declare that they have no competing interests.

References

[1] G. Qiao, M. Yang, and H. Wang, (2022) *"A water level measurement approach based on YOLOv5s"* **Sensors** **22**(10): 3714. DOI: 10.3390/s22103714.

[2] Z. Xie, J. Jin, J. Wang, R. Zhang, and S. Li, (2023) *"Application of deep learning techniques in water level measurement: Combining improved SegFormer-UNet model with virtual water gauge"* **Applied Sciences** ***13***(9): 5614. DOI: 10.3390/app13095614.

[3] W.-C. Liu and W.-C. Huang, (2024) *"Evaluation of deep learning computer vision for water level measurements in rivers"* **Heliyon** ***10***(4): e25989. DOI: 10.1016/j.heliyon.2024.e25989.

[4] C. A. G. Santos, M. A. Ghorbani, E. Abdi, U. Patel, and S. Sadeddin, (2025) *"Estimating water levels through smartphone-imaged gauges: A comparative analysis of ANN, DL, and CNN models"* **Water Resources Management** ***39***(4): 1639–1654. DOI: 10.1007/s11269-024-04038-w.

[5] N. Mohamadiazar, A. Ebrahimian, and H. Hosseiny, (2024) *"Integrating deep learning, satellite image processing, and spatial-temporal analysis for urban flood prediction"* **Journal of Hydrology** ***639***: 131508. DOI: 10.1016/j.jhydrol.2024.131508.

[6] D. Zhang and J. Tong, (2023) *"Robust water level measurement method based on computer vision"* **Journal of Hydrology** ***620***: 129456. DOI: 10.1016/j.jhydrol.2023.129456.

[7] W. Jung, J. Kim, H. Jo, S. Lee, and B. Kim, (2025) *"Video-based deep learning approach for water level monitoring in reservoirs"* **Water** ***17***: 2525. DOI: 10.3390/w17172525.

[8] X. Blanch, J. Grundmann, R. Hedel, and A. Eltner, (2026) *"AI image-based method for a robust automatic real-time water level monitoring: A long-term application case"* **Hydrology and Earth System Sciences** ***30***(3): 797–824. DOI: 10.5194/hess-30-797-2026.

[9] H. Guo, Y. Wang, and Y. Zhang, (2022) *"A water gauge scale capturing method in tidal well based on image recognition"* **Journal of Physics: Conference Series** ***2320***(1): 012029. DOI: 10.1088/1742-6596/2320/1/012029.

[10] J. H. Lee and J. K. Jung, (2024) *"Development of image-based water level sensor with high-resolution and low-cost using image processing algorithm"* **Sensors** ***24***(23): DOI: 10.3390/s24237699.

[11] C. A. Krabbenhoft et al., (2022) *"Assessing placement bias of the global river gauge network"* **Nature Sustainability** ***5***(7): 586–592. DOI: 10.1038/s41893-022-00873-0.

[12] N. A. Muhadi, A. F. Abdullah, S. K. Bejo, M. R. Mahadi, A. Mijic, and Z. Vojinovic, (2024) *"Deep learning and LiDAR integration for surveillance camera-based river water level monitoring in flood applications"* **Natural Hazards** ***120***(9): 8367–8390. DOI: 10.1007/s11069-024-06503-6.

[13] Roboflow. *Water gauge dataset overview*. Roboflow Universe. Accessed 28 March 2026. 2026.

[14] G. Bai et al., (2021) *"An intelligent water level monitoring method based on SSD algorithm"* **Measurement** ***185***: 110047. DOI: 10.1016/j.measurement.2021.110047.

[15] L. Liang et al., (2023) *"Study and application of image water level recognition calculation method based on Mask RCNN and Faster R-CNN"* **Applied Ecology and Environmental Research** ***21***(6): 5039–5053. DOI: 10.15666/aeer/2106_50395053.

[16] X. Wang, Z. Li, Y. Zhang, and G. An, (2023) *"Water level recognition based on deep learning and character interpolation strategy for stained water gauge"* **River** **2**(4): 506–517. DOI: 10.1002/rvr2.69.

[17] R. Qiu, Z. Cai, Z. Chang, S. Liu, and G. Tu, (2023) *"A two-stage image process for water level recognition via dual-attention CornerNet and CTransformer"* **The Visual Computer** ***39***(7): 2933–2952. DOI: 10.1007/s00371-022-02501-6.