2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

EdgeMeter: An Edge Computing System and Algorithm for Intelligent Water Meter Recognition

Ximing Li[12][0000-0003-2031-7729],Xiaosheng Xie[1] ,Yue Zhang[1] ,Min Wang[1]

Xiao Du[1] ,Zelin He[1] ,Yubin Guo✉[1][0000-0002-3572-4768]

[1] College of Mathematics and Informatics South China Agricultural University, 510642, Guangzhou, China
[2]Key Laboratory of Smart Agricultural Technology in Tropical South China, Ministry of Agriculture and Rural Affairs, Guangzhou, 510642, China

Abstract.
Accurate and efficient water meter reading recognition system is essential for intelligent water resource management. However, existing systems face several challenges, including the high deployment costs of replacing old meters with smart ones, the limited device lifespan caused by local recognition on embedded devices, and the increased server workload associated with server-based processing. To address these issues, we propose an intelligent water meter recognition system based on a three-layer edge computing architecture. The IoT layer is responsible for data collection, utilizing an HC32F460 chip to capture automatically capture water meter images, compress the images, and transmit them to the edge layer. The edge layer is primarily composed of the YOLO-METER algorithm for water meter reading recognition. Based on YOLO11n, we have improved two modules by integrating FastC3k2 to enhance the extraction of low-contrast features and MRFBlock to refine feature selection and improve the localization of reading regions on the water meter. The cloud layer periodically aggregates water meter readings, performs data analysis, and provides users related information, enabling real-time monitoring and insights. Experiments show that YOLO-METER achieves a 2.4% higher mAP50 and 6% fewer parameters than YOLO11n, enhancing recognition accuracy while reducing computational cost. This system facilitates efficient water usage monitoring, thereby improving operational efficiency and contributing to intelligent resource management.

Keywords: Edge Computing, Smart Meters, Reading Recognition, Deep Learning, IoT.

1 Introduction

Water resources are crucial for human development and survival, playing an essential role in global sustainable development [1]. Efficient management of water resources not only promotes conservation and reduces consumption but also eases environmental burdens, restores ecological balance, and improves public welfare. For water utilities, optimized resource management leads to reduced operational costs, improved network efficiency, minimized leakage losses, and better management of growing urban water demands [2].

However, traditional manual meter reading methods still exist in many water resource management systems. These methods suffer from low efficiency and time-consuming operations, failing to meet the accuracy and real-time control demands of modern urban monitoring[3].

To overcome these challenges, advanced meters like electromagnetic [4], fluid [5], and ultrasonic[6] types enable automated data collection and wireless transmission. However, high costs and infrastructure replacement hinder large-scale adoption. As a result, upgrading existing meters with intelligent technologies without hardware changes has become a key research focus.

In recent years, the rapid advancement of IoT [7] and deep learning technologies has greatly improved image recognition-based dial reading systems. These systems can be broadly categorized into two kinds based on where the algorithms are deployed. The first kind, local-based[8, 9], involves deploying algorithms directly on terminal devices. Here, the terminal captures images, performs reading recognition, and transmits the results to servers. However, this method is limited by the terminal's resources, leading to lower recognition accuracy and reduced equipment life. The second kind, cloud-based[3, 10], deploys the algorithm in the cloud, where the terminal only captures images and sends them for processing. However, as the number of tasks increases, cloud servers may become overloaded, impacting response speed, system stability, and increasing demands on data transmission bandwidth, along with potential security risks.

We propose EdgeMeter, a smart water meter recognition system based on a three-layer edge computing architecture to improve data acquisition, processing, and storage efficiency. The system is shown in **Fig. 1**.

The system consists of three layers: the IoT layer, which utilizes an HC32F460 chip to automatically capture water meter images, compress them to reduce size, and transmit the grayscale images to the edge layer at regular intervals to minimize power consumption; The edge layer, primarily composed of NCT3568 boards, deploys YOLO-METER, a lightweight and efficient recognition algorithm specifically designed for water meter reading detection. The architecture of YOLO-METER is illustrated in **Fig. 1.** YOLO-METER achieves a 2.4% improvement in mAP50 over YOLO11n while reducing the number of parameters by 6%, making it ideal for edge deployment; And the cloud layer, which aggregates the data from the edge layer, provides user services such as consumption tracking, leak detection, and water forecasting, and enables remote management and monitoring of devices.

The main contributions of this paper are as follows.

1. Smart Water Meter System

In this paper, we propose a smart water meter system named EdgeMeter, based on a three-layer edge computing architecture, comprising the IoT, edge, and cloud layers. This system fully leverages the advantages of edge computing, reducing the load on central servers while also extending the life of the devices used for water meter image data collection.

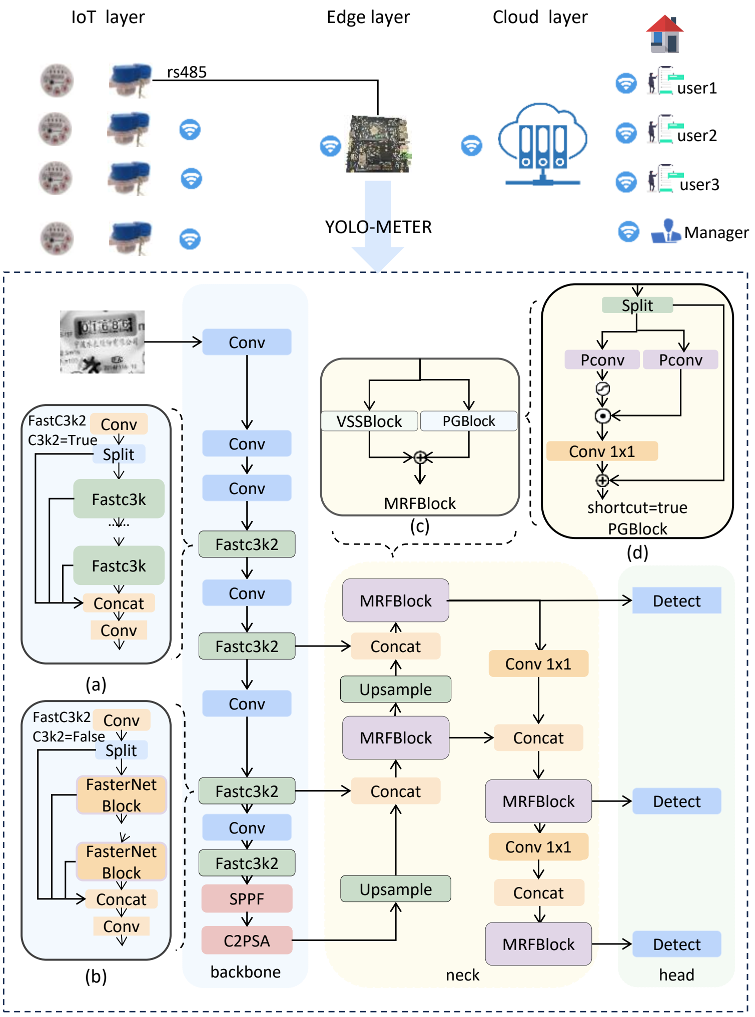
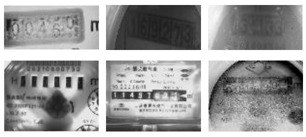


Fig. 1. System architecture of the smart water and electricity metering system, consisting of IoT layer, edge layer, and cloud layer



(*a*) (*b*) (*c*)

Fig. 2. Low-quality captured images;(a): severe blurring;(b): underexposure, overexposure;(c):fog

2. YOLO-METER

Based on YOLO11n, we introduce YOLO-METER, an algorithm designed to enhance meter reading detection. By improving feature extraction in low-contrast areas and refining feature selection in digit-wheel regions, YOLO-METER significantly enhances the accuracy of water meter reading recognition.

3. Water Meter Dataset

We have constructed a dataset comprising 9,658 meter images, including both those captured under normal conditions and those taken in challenging environments, such as blurry, low-light and overexposed conditions. This dataset effectively covers a wide range of real-world scenarios encountered in practical applications.

2 Related Work

2.1 Water Meter Reading Acquisition

Water meter reading Acquisition has been a critical task in automating water consumption monitoring. Based on the method of acquiring the readings, these approaches can be broadly categorized into two main types: non-image-based methods and image-based methods.

Non-image-based water meter reading methods use sensors like ultrasonic[6], fluidic[5], and electromagnetic[4] to collect data without image processing. Though efficient and reliable, they often need specialized infrastructure, making them less flexible in areas with traditional meters.

Image-based methods for water meter reading recognition can be broadly divided into four categories: traditional machine learning-based methods, object detection-based methods, sequence modeling methods, and hybrid multi-stage methods. Each of these approaches utilizes different techniques to process and interpret water meter images, offering varying degrees of accuracy and adaptability to different environments and meter designs.

Traditional machine learning methods have been widely applied to water meter recognition. Zhao et al. [11] used handcrafted features, Oliveira et al. [12] applied KNN, and Edward [13] improved SVM with features like histograms and contour profiles. These approaches rely on image feature extraction for reading recognition.

Object detection-based methods are popular for automatically locating key regions in meter images. Liang et al. [14] found YOLOv3 more accurate than Faster R-CNN. Zhu et al. [15] improved YOLOv4 with data augmentation and spatial attention. Martinelli et al. [16] used YOLOv5, while Li et al. [3] proposed a lightweight spliced CNN. Zhang et al. [17, 18] enhanced recognition by detecting keypoints and dial features. Peng and Chen [19] introduced an RFCN-based two-stage model for improved reading accuracy.

Sequence modeling methods treat meter reading as a sequence recognition task. Yang et al. [20] proposed a fully convolutional network for fast reading, but it only handles fixed-digit meters. Xiu et al. [21] introduced an end-to-end approach, though its high resource demands limit deployment on edge devices.

Lastly, Hybrid multi-stage methods integrate detection, correction, and classification to boost recognition accuracy. Ktari et al. [22] combined YOLOv4 with OCR, Chen et al. [23] used U-Net for segmentation and VGG16 for recognition, while Zhao et al. [24] applied U-Net for correction and used their WMRRM model for character recognition.

Object detection-based methods offer a good balance between accuracy and deployment, effectively addressing common issues like tilt, blur, and poor lighting.

2.2 Water Meter System

The water meter system can be categorized into local-based and cloud-based solutions, depending on where the water meter readings are obtained.

For local-based solutions, Ye et al. [6] proposed a LoRa-enabled smart meter with STM32 for local processing. Han and Kim [9] introduced MO-CNN to reduce memory use, enabling deployment on low-power devices. Other approaches use ultrasonic meters with M-BUS [25], flow meters with valve control [26], turbine generators for signal processing [27], and flow sensors with LoRa transmission [28]. Wireless systems using LPWAN [29] and real-time monitoring setups [30] have also been developed.

In contrast, cloud-based solutions transmit meter images to remote servers for processing. Alvisi et al.[10] introduced SWaMM, which collects water meter data and sends it to the cloud for recognition. Li et al.[3] transmitted water meter images to the server for recognition via 4G-network modules.

3 System Design and Method

3.1 System design

The system, which is named EdgeMeter, is primarily divided into three layers: the IoT layer, edge layer, and cloud layer, ensuring the system's efficiency and flexibility in data acquisition, processing, and storage.

3.2 IoT layer

The IoT layer is responsible for water meter data acquisition, capturing water meter images at regular intervals and sending them to the edge layer, which is mounted on top of the water meter dial, as shown in **Fig. 3.** Structural Design of the IoT Layer Enclosure. The IoT layer includes a power supply, a control switch, a camera module, LEDs, a 4G data transmission module (SLM332), and an HC32F460.

The HC32F460 is based on the ARMv7-M architecture, known for its low power consumption. In addition to the RUN mode, it also features a Sleep mode, in which only the MCU's timer remains powered while the other modules are in a power-off state, ensuring very low power consumption. This makes it highly suitable for the short-duration operation required for water meter data capture.The power supply consists of a 3.6V Li-ion battery, which provides power to the entire device. To extend the battery life, we leverage the features of the HC32F460 chip. A timer (RTC) is configured to output a high-level signal at specified intervals. This signal activates the MOSFET, which supplies power to the other components on the board. When powered on, the circuit's LEDs light up, the camera captures an image, and the HC32F460 enters RUN mode to process the data. The front and back views of the board are shown in **Fig. 4**.

The camera module captures images of the water meter. To ensure optimal image quality, LED lights on both sides of the board illuminate during the capture process, providing sufficient brightness for clearer images. After the picture is taken, the HC32F460 chip performs a compression operation. This involves grayscaling the image, converting it to base64 format, and then transmitting it.

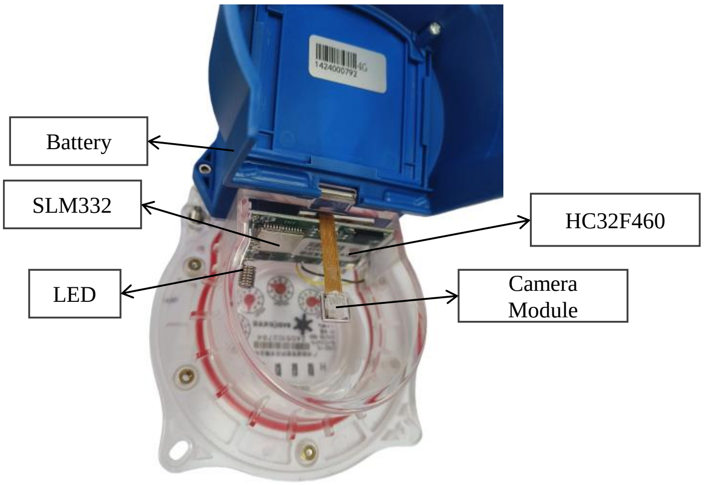


Fig. 3. Structural Design of the IoT Layer Enclosure

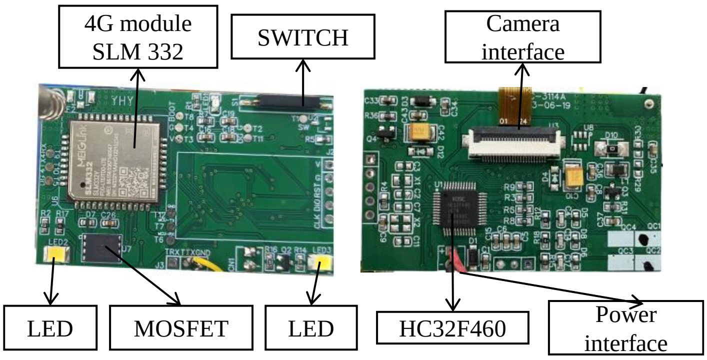


Fig. 4. Detailed View of the Circuit Board in the IoT Layer

3.3 Edge layer

The edge layer is responsible for processing images received from the IoT layer to recognize and analyze water meter readings. The NCT3568, a high-performance and energy-efficient development motherboard based on the RK3568, serves as the core of this layer. It is equipped with a 4G module for communication via the HTTP protocol. The YOLO-METER recognition algorithm and its associated database are deployed on the NCT3568 to perform the meter reading analysis.

The edge layer mainly processes the images sent from the IoT layer to complete the recognition and analysis of water meter readings. NCT3568 is a high-performance and low-power consumption development motherboard device based on the RK3568.Edge layer is equipped with 4G modules, and communicates with the http protocol. NCT3568 deploys the recognition algorithm YOLO-METER and database.

Dataset.

We collected a total of 9,658 images of wheel dials through the IoT layer, which were split into 5,800 images for the training set, 1,919 images for the validation set, and 1,859 images for the test set, as shown in Table 1. Since the mechanical water meter has gradual increment values, there are intermediate states between readings. To account for this, we defined 20 categories: 10 integer categories (0, 1, 2, ..., 9) and 10 decimal categories (0.5, 1.5, 2.5, ..., 9.5).

To further analyze the composition of the dataset, we visualized the class distribution across the training, validation, and test sets, as shown in **Fig. 5**.. Since class 0 contains a significantly larger number of samples than the other classes, we applied a logarithmic transformation to the sample counts. This transformation improves the visibility of categories with smaller sample sizes, providing a clearer comparative overview of all classes. As depicted in **Fig. 5**, the proportions of each class are relatively consistent across the different subsets. This suggests that the dataset maintains a balanced distribution, which is crucial for ensuring fair model training and evaluation.

Table 1. The number of images in training, validation, and test sets.

Subset	Number of Images
Train	5880
validation	1919
Test	1859
Total	9658

Algorithm Architecture.

In this paper, we propose YOLO-METER, an improved version of YOLO11, tailored for the characteristics of the word-wheel dial dataset. Firstly, we introduce FastC3k2, an enhancement based on the FasterNet architecture proposed by Chen et al. [31]. FastC3k2 reduces the model's parameter count and FLOPs while improving feature extraction from images. Secondly, the feature fusion component in the original YOLOv4 design uses convolutional operations, which suffer from a limited receptive field and weak ability for autonomous feature selection. To address these limitations, we propose a new module, MRFBlock, based on Mamba and a gating mechanism. This module significantly expands the model's receptive field and enhances its ability to autonomously select features, improving multi-scale fusion and, consequently, the model's detection performance and accuracy. The overall architecture of the algorithm is shown in **Fig. 1**.

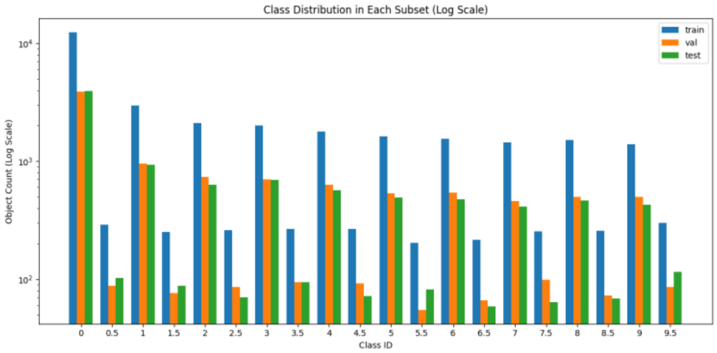


Fig. 5. Class Distribution Comparison Across Train, Validation, and Test Sets

FastC3k2.

Chen et al.[32] in order to design a fast neural network, the relationship between floating point numbers and access to memory was analyzed and a new partial convolution (PConv) was proposed, based on which Fasternet was further proposed.

The architecture of FastC3k2 is shown in **Fig. 1**(a). Input X first passes through Conv, and then splits into two branches X_1 and X_2 .FastC3k also includes the FasterNet Block, as shown in **Fig. 6**.The input $Z^{(i-1)}$ is convolved by Conv to get O, followed by FasterNet Block operation, the result of the calculation is then spliced with O, and finally another convolution calculation is carried out to get $Z^{(i)}$.

When C3k2 is False, the intermediate output Z^{i} is computed by FasterNet Block. FasterNet Block first Partial Convolution, then 1x1 convolution, BN layer, activation function using ReLU, followed by a layer of 1x1 convolution layer, and then finally the results of the computation and the input at the beginning of the residual connection [32]. The structure is illustrated in **Fig. 1**(b).

MRFBlock.

The 2D-Selective-Scan Mechanism (SS2D) [35] is the core component of the Visual State Space Block (VSSBLOCK), which is utilized to construct the hidden state space for cross-modal feature fusion. The VSSBLOCK passes through a linear layer, followed by the Depthwise Convolution (DWConv) and SILU activation functions. It then passes through the SS2D, undergoes layer normalization, and concludes with an additional linear layer, as shown in **Fig. 7**.

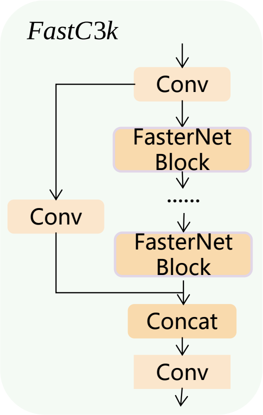


Fig. 6. FastC3k

MRFBlock consists of two parts, VSSBLOCK and PGBlock. These two branches leverage the respective strengths of Mamba and CNN, fusing the features at the end. The structure is illustrated in **Fig. 1**(c). The input Z^{l-1} passes through VSSBlock and PGBlock, and the output Z^{l} is obtained, as shown in Eq.(1).

PGBlock efficiently filters features using CNN and valves, incorporating PConv for feature fusion without excessive computational cost. Its structure is shown in **Fig. 1**. By splitting the input X^{l-2} into two branches, PGBlock performs computations in parallel.

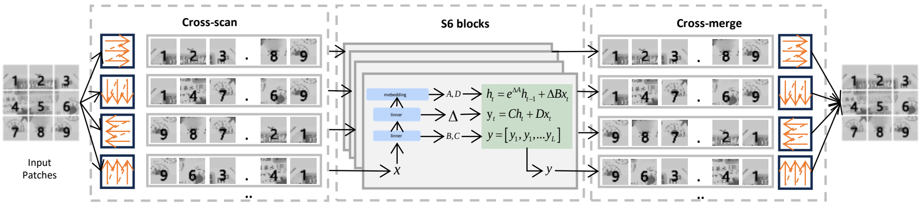


Fig. 7. ss2d(2D-Selective-Scan Mechanism)

$$Z^{l} = \text{VSSBlock}\left(Z^{l-1}\right) + \text{PGBlock}\left(Z^{l-1}\right) \tag{1}$$

PGBlock uses ReLU as the activation function to control the information flow of X_1^{L-1}, which is then merged with X_2^{l-1} via dot product, refined with global features to blend channel information via Conv_{1x1}. Finally, the residual connection is fused with the original input X^{l-2} fusion. The PG Block captures more global features while bringing only a slight increase in computational cost, and the resulting output feature X^{l} is defined as

$$X^{l} = \text{Conv}_{1\times 1}\left(X_1^{l-1} \odot \Phi\left(X_2^{l-1}\right)\right) + X^{l-2} \tag{2}$$

Where Φ represents the activation function (ReLU)

PGBlock captures more global features with minimal computational overhead. It uses a gating mechanism combined with convolution to preserve spatial information and capture detailed image features. Unlike traditional MLP, PGBlock transfers global features pixel-by-pixel, enriching contextual expression and improving feature modeling capabilities. The structure is illustrated in **Fig. 1**(d).

3.4 Cloud layer

The cloud layer serves as the core platform for data aggregation, user interaction, and system management. It provides real-time water usage monitoring, consumption forecasting, historical data analysis, and leak alerts. Additionally, it supports remote device management, enabling technicians to perform diagnostics, track performance, and detect faults. This integration enhances intelligent water management and system reliability.

4 Experiment and Discussion

4.1 Experimental indicators

Experimental metrics using Precision, Recall, mAP, Flops, Parameters. The hyperparameters for the experiments were set as follows: batch size = 32, epochs = 300, weight decay = 0.0005, an initial learning rate of lr0 = 0.01, and a final learning rate of lrf = 0.0001. The variables for the ablation experiments primarily focused on FastC3k2 and MRFBlock.

4.2 Comparison Experiment

The comparison models are categorized into one-stage and two-stage models. The two-stage model includes Fast R-CNN[33], which uses ResNet50 as its backbone network. The one-stage models include RTDETR ， YOLOV5x 、YOLOV6,YOLOV7,YOLOV8,YOLOV9[34],YOLOV10.

As shown in Table 2, YOLO-METER achieves a mAP of 94.9, which is significantly higher than the other models. Compared to RTDETR, YOLOv5, YOLOv6, YOLOv7, YOLOv8, YOLOv9, and YOLOv10, YOLO-METER's mAP is improved by 7.6%, 2.7%, 1.9%, 3.6%, 2.6%, 1.6%, and 2.4%, respectively. Additionally, YOLO-METER has fewer parameters than the other models, with 25.93M fewer than RTDETR, 0.13M fewer than YOLOv5, 1.73M fewer than YOLOv6, 34.03M fewer than YOLOv7, and 0.53M fewer than YOLOv8. While YOLOv9 and YOLOv10 have fewer parameters than YOLO-METER, their accuracy is not as high, and their floating point operations exceed those of YOLO-METER. Overall, YOLO-METER strikes an effective balance between accuracy and speed, making it highly suitable for deployment on edge devices.

Table 2. Comparison of Different Network Architectures

Network Architecture	mAP	Precision	Recall	mAP@50-95	Flops	Parameters
Fast R-CNN	93.1	89.9	90.2	85.1	182	41.3
RTDETR	87.3	90.8	91.2	76.5	100.7	28.4
YOLOv5	91.1	91.1	91.5	78.7	7.7	2.6
YOLOv6	93.1	91.6	91.3	84.3	11.8	4.2
YOLOv7	91.3	92.1	92.3	79.9	103	36.5
YOLOv8	92.7	90.9	93.2	85.2	8.1	3.0
YOLOv9	93.3	91.2	92.0	85.7	7.6	1.97
YOLOv10	92.5	91.5	91.0	84.7	6.5	**2.2**
YOLO11	92.5	90.4	90.7	85.3	6.5	2.6
YOLO-METER	**94.9**	**92.5**	**94.3**	**87.6**	**6.3**	2.47

4.3 Ablation study

As shown in Table 4, replacing C3K2 with FastC3k2 reduces parameters by 11.9% and FLOPs by 13.4%, while increasing mAP by 0.7%. Though MRFBlock adds slight computational overhead, it boosts mAP by 1.2%. Combining both modules cuts parameters and FLOPs further while improving mAP by 2.4%. As illustrated in **Fig. 8**, YOLO-METER accurately detects bounding boxes even in blurry images and effectively avoids interference from surrounding digits.

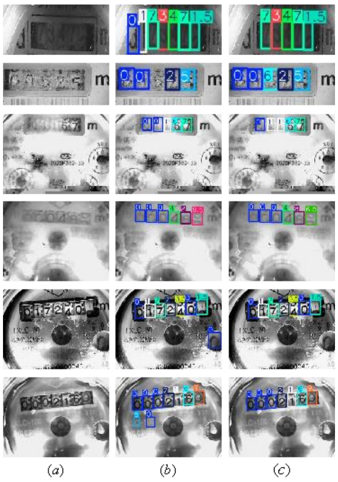


Fig. 8. (a)Original images;(b)Inference results of YOLO11n;(c)Inference results of our model

Table 3. Comparison of Different Network Architectures

Network Architecture	Activation Function	mAP	Precision	Recall	mAP@50-95	Flops	Parameters
YOLO11 + MRFBlock	GELU	93.1	91.2	92.2	85.7	6.9	2.74
YOLO11 + MRFBlock	Leaky ReLU	93.4	91.4	92.5	86.0	6.8	2.74
YOLO11 + MRFBlock	PReLU	93.3	91.3	92.4	85.8	6.8	2.80
YOLO11 + MRFBlock	RELU	**93.7**	**91.7**	**92.7**	**86.3**	**6.7**	**2.74**

Table 4. Comparison of Different Network Architectures

Network Architecture	FastC3k2	MRF-Block	mAP	Precision	Recall	mAP @50-95	Flops	Parameters
YOLO11			92.5	90.4	90.7	85.3	6.5	2.6
YOLO11	✓		93.2	90.6	91.5	85.5	**5.8**	**2.29**
YOLO11		✓	93.7	91.7	92.7	86.3	6.7	2.74
YOLO11	✓	✓	**94.9**	**92.5**	**94.3**	**87.6**	6.3	2.47

To assess the effect of activation functions on PGBlock performance, we tested ReLU, Leaky ReLU, PReLU, and GELU across multiple network setups. As shown in Table 3, activation choice notably impacts accuracy and efficiency. ReLU achieved the best mAP and recall, proving effective for fine-grained feature extraction and robust representation learning.

We plotted the same layer of sensory wildness visualization as shown in **Fig. 9**,The MRFBlock module significantly improves the sensory wildness of the model.

To illustrate FastC3k2's ability in detecting character box features, this study uses Grad-CAM++ to generate heat maps. These maps visualize the areas most influential to the model’s predictions. Redder regions indicate stronger attention, while bluer areas show weaker focus. As shown in **Fig. 10**, FastC3k2 helps the model concentrate more effectively on key image details, highlighting its advantage in feature extraction.

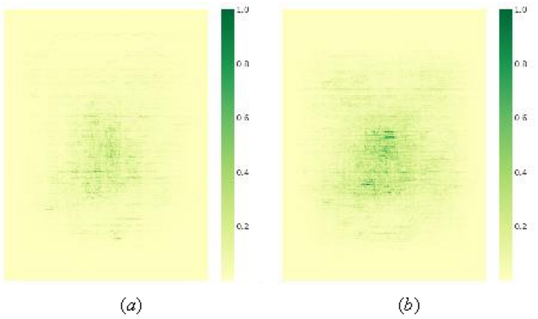


Fig. 9. (a)wildness of YOLO11; (b)wildness of YOLO11+MRFBlock

4.4 Water analysis

In this analysis, the daily water consumption data of some water meter dials in Panyu District, Guangdong Province were collected and statistically examined, as shown in Table 5. The water meters were categorized based on pipe caliber, primarily including 15, 20, 25, and 80 calibers, with 15 and 20 calibers being the most common in residential water consumption. The analysis of average daily water consumption across different months provides a valuable benchmark for users.

Water use early warning relies on analyzing historical consumption data. Holt's Linear Trend Method [35] was used to predict consumption, with smoothing coefficients set to 0.8 for level and 0.2 for trend.

Table 5. Average Daily Water Usage by Diameter Size

caliber(mm)	number	Aug (m³/day)	Sep (m³/day)	Oct (m³/day)	Nov (m³/day)	Dec (m³/day)	Average(m³/day)
15	3310	2.38	2.16	2.13	3.26	2.19	2.43
20	3342	1.07	1.23	1.08	1.79	1.17	1.26
25	744	2.84	2.96	3.13	2.55	3.72	3.04
80	301	130.77	157.54	149.98	164.90	163.76	153.30

5 Conclusion

In this paper, we design a fully automated, multi-functional smart water meter system comprising three layers: the IoT layer, the edge layer, and the cloud layer.

The IoT layer is responsible for image capture, compression, and uploading, with the HC32F460 chip handling these tasks efficiently. The edge layer, powered by the NCT3568 chip, deploys the YOLO-METER algorithm, which improves water meter image recognition accuracy through the FastC3k2 and MRFBlock modules, significantly enhancing feature extraction and detection accuracy while reducing model parameters.The cloud layer aggregates data from the edge, enabling users and administrators to monitor water consumption and manage the system effectively. This layered architecture ensures high efficiency, reliability, and optimized power usage.

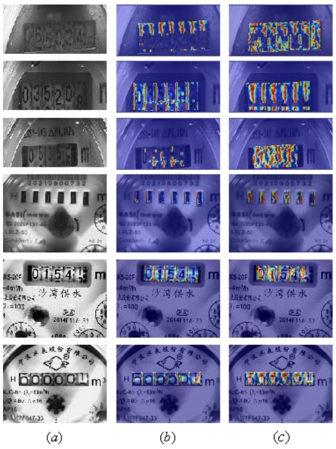


Fig. 10. (a)Original images;(b)heat maps of YOLO11n;(c)heat maps of YOLO11+FastC3k2

References

1. Katusiime, J., Schütt, B.: Integrated water resources management approaches to improve water resources governance. Water. 12, 3424 (2020).

2. Gupta, A.D., Pandey, P., Feijóo, A., Yaseen, Z.M., Bokde, N.D.: Smart water technology for efficient water resource management: A review. Energies. 13, 6268 (2020).

3. Li, C., Su, Y., Yuan, R., Chu, D., Zhu, J.: Light-Weight Spliced Convolution Network-Based Automatic Water Meter Reading in Smart City. IEEE Access. 7, 174359–174367 (2019). https://doi.org/10.1109/ACCESS.2019.2956556.

4. Chen, B., Zhou, Y., Huang, T., Ying, S., Chen, T.: Technical features and application of electromagnetic flow meter. In: 2020 International Conference on Communications, Information System and Computer Engineering (CISCE). pp. 1–5. IEEE (2020).

5. Buck, B.S., Johnson, M.C., Barfuss, S.L.: Effects of particulates on water meter accuracy through expected life. Journal-American Water Works Association. 104, E231–E242 (2012).

6. Kumar, K., Swain, T., Ekhande, C., Nanaware, R.: Effects of flow measurement on sloped pipes using ultrasonic flowmeter. In: 2015 International Conference on Industrial Instrumentation and Control (ICIC). pp. 1490–1494. IEEE (2015).

7. Brundu, F.G., Patti, E., Osello, A., Del Giudice, M., Rapetti, N., Krylovskiy, A., Jahn, M., Verda, V., Guelpa, E., Rietto, L., others: IoT software infrastructure for energy management and simulation in smart cities. IEEE Transactions on Industrial Informatics. 13, 832–840 (2016).

8. Ye, Y., Yang, Y., Zhu, L., Wang, J., Rao, D.: A LoRa-based Low-power Smart Water Metering System. In: 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE). pp. 301–305. IEEE, Guangzhou, China (2021). https://doi.org/10.1109/ICCECE51280.2021.9342327.

9. Han, D., Kim, H.: A number recognition system with memory optimized convolutional neural network for smart metering devices. In: 2018 International Conference on Electronics, Information, and Communication (ICEIC). pp. 1–4. IEEE, Honolulu, HI (2018). https://doi.org/10.23919/ELINFOCOM.2018.8330594.

10. Alvisi, S., Casellato, F., Franchini, M., Govoni, M., Luciani, C., Poltronieri, F., Riberto, G., Stefanelli, C., Tortonesi, M.: Wireless Middleware Solutions for Smart Water Metering. Sensors. 19, 1853 (2019). https://doi.org/10.3390/s19081853.

11. Shutao Zhao, Baoshu Li, Jinsha Yuan, Guiyan Cui: Research on Remote Meter Automatic Reading Based on Computer Vision. In: 2005 IEEE/PES Transmission &amp; Distribution Conference &amp; Exposition: Asia and Pacific. pp. 1–4. IEEE, Dalian, China (2005). https://doi.org/10.1109/TDC.2005.1546972.

12. Oliveira, D.M., Cruz, R.D.S., Bensebaa, K.: Automatic Numeric Characters Recognition of Kilowatt-Hour Meter. In: 2009 Fifth International Conference on Signal Image Technology and Internet Based Systems. pp. 107–111. IEEE, Marakesh, Morocco (2009). https://doi.org/10.1109/SITIS.2009.27.

13. Edward V, C.P.: Support Vector Machine based automatic electric meter reading system. In: 2013 IEEE International Conference on Computational Intelligence and Computing Research. pp. 1–5. IEEE, Enathi (2013). https://doi.org/10.1109/ICCIC.2013.6724185.

14. Liang, Y., Liao, Y., Li, S., Wu, W., Qiu, T., Zhang, W.: Research on water meter reading recognition based on deep learning. Sci Rep. 12, 12861 (2022). https://doi.org/10.1038/s41598-022-17255-3.

15. Zhu, J., Li, M., Jiang, J., Li, J., Wang, Z., Shen, J.: Automatic wheel-type water meter digit reading recognition based on deep learning. J. Electron. Imag. 31, (2022). https://doi.org/10.1117/1.JEI.31.2.023023.

16. Martinelli, F., Mercaldo, F., Santone, A.: Water Meter Reading for Smart Grid Monitoring. Sensors. 23, 75 (2022). https://doi.org/10.3390/s23010075.

17. Zhang, J., Liu, W., Xu, S., Zhang, X.: Key point localization and recurrent neural network based water meter reading recognition. Displays. 74, 102222 (2022).

18. Zhang, Q., Bao, X., Wu, B., Tu, X., Jin, Y., Luo, Y., Zhang, N.: Water meter pointer reading recognition method based on target-key point detection. Flow Measurement and Instrumentation. 81, 102012 (2021). https://doi.org/10.1016/j.flowmeasinst.2021.102012.

19. Peng, Y., Chen, Z.: Application of Deep Residual Neural Network to Water Meter Reading Recognition. In: 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). pp. 774–777. IEEE, Dalian, China (2020). https://doi.org/10.1109/ICAICA50127.2020.9182460.

20. Yang, F., Jin, L., Lai, S., Gao, X., Li, Z.: Fully Convolutional Sequence Recognition Network for Water Meter Number Reading. IEEE Access. 7, 11679–11687 (2019). https://doi.org/10.1109/ACCESS.2019.2891767.

21. Xiu, H., He, J., Zhang, X., Wang, L., Qi, Y.: HRC-mCNNs: A Hybrid Regression and Classification Multibranch CNNs for Automatic Meter Reading With Smart Shell. IEEE Internet Things J. 9, 25752–25766 (2022). https://doi.org/10.1109/JIOT.2022.3197930.

22. Ktari, J., Frikha, T., Hamdi, M., Elmannai, H., Hmam, H.: Lightweight AI Framework for Industry 4.0 Case Study: Water Meter Recognition. BDCC. 6, 72 (2022). https://doi.org/10.3390/bdcc6030072.

23. Chen, L., Sun, W., Tang, L., Jiang, H., Li, Z.: Research on Automatic Reading Recognition of Wheel Mechanical Water Meter Based on Improved U-Net and VGG16. WSEAS TRANSACTIONS ON COMPUTERS. 21, 283–293 (2022). https://doi.org/10.37394/23205.2022.21.35.

24. Zhao, S., Lu, Q., Zhang, C., Ahn, C.K., Chen, K.: Effective recognition of word-wheel water meter readings for smart urban infrastructure. IEEE Internet of Things Journal. (2024).

25. Cherukutota, N., Jadhav, S.: Architectural framework of smart water meter reading system in IoT environment. In: 2016 International Conference on Communication and Signal Processing (ICCSP). pp. 0791–0794. IEEE, Melmaruvathur, Tamilnadu, India (2016). https://doi.org/10.1109/ICCSP.2016.7754253.

26. Amir, A., Fauzi, R., Arifin, Y.: Smart water meter for automatic meter reading. IOP Conf. Ser.: Mater. Sci. Eng. 1212, 012042 (2022). https://doi.org/10.1088/1757-899X/1212/1/012042.

27. Li, X.J., Chong, P.H.J.: Design and Implementation of a Self-Powered Smart Water Meter. Sensors. 19, 4177 (2019). https://doi.org/10.3390/s19194177.

28. Hudiono, H., Taufik, M., Yoga Perdana, R.H., Rakhmania, A.E.: Digital centralized water meter using 433 MHz LoRa. Bulletin EEI. 10, 2062–2071 (2021). https://doi.org/10.11591/eei.v10i4.2950.

29. Sushma, N., Suresh, H.N., Lakshmi, J.M., Srinivasu, P.N., Bhoi, A.K., Barsocchi, P.: A Unified Metering System Deployed for Water and Energy Monitoring in Smart City. IEEE Access. 11, 80429–80447 (2023). https://doi.org/10.1109/ACCESS.2023.3299825.

30. Hsia, S.C., Wang, S.-H., Hsu, S.-W.: Smart Water-Meter Wireless Transmission System for Smart Cities. IEEE Consumer Electron. Mag. 10, 83–88 (2021). https://doi.org/10.1109/MCE.2020.3043997.

31. Chen, J., Kao, S., He, H., Zhuo, W., Wen, S., Lee, C.-H., Chan, S.-H.G.: Run, Don’t Walk: Chasing Higher FLOPS for Faster Neural Networks, http://arxiv.org/abs/2303.03667, (2023). https://doi.org/10.48550/arXiv.2303.03667.

32. Chen, J., Kao, S., He, H., Zhuo, W., Wen, S., Lee, C.-H., Chan, S.-H.G.: Run, Don’t Walk: Chasing Higher FLOPS for Faster Neural Networks, http://arxiv.org/abs/2303.03667, (2023). https://doi.org/10.48550/arXiv.2303.03667.

33. Li, J., Liang, X., Shen, S., Xu, T., Feng, J., Yan, S.: Scale-aware fast R-CNN for pedestrian detection. IEEE transactions on Multimedia. 20, 985–996 (2017).

34. Wang, C.-Y., Yeh, I.-H., Liao, H.-Y.M.: YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information, https://arxiv.org/abs/2402.13616, (2024).

35. Hyndman, R.: Forecasting: principles and practice. OTexts (2018).