Smart Agricultural Technology 5 (2023) 100236

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

An Edge-computing flow meter reading recognition algorithm optimized for agricultural IoT network

Le Liu [a], Xin Qiao [b,c,*], Wei-zhen Liang [b,c], Joseph Oboamah [c,d], Jun Wang [e], Daran R. Rudnick [b,f], Haishun Yang [g], Abia Katimbo [b,f], Yeyin Shi [b]

[a] School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, China
[b] Department of Biological Systems Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA
[c] Panhandle Research and Extension Center, University of Nebraska-Lincoln, Scottsbluff, NE, USA
[d] Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA
[e] Department of Chemical and Environmental Engineering, University of Iowa, Iowa City, IA, USA
[f] West Central Research and Extension Center, University of Nebraska-Lincoln, North Platte, NE, USA
[g] Department of Agronomy and Horticulture, University of Nebraska-Lincoln, Lincoln, NE, USA

ARTICLE INFO

Editor: Stephen Symons

Keywords:
Deep learning
Digits recognition
Internet of Things
Flow meter
Agricultural water use

ABSTRACT

Groundwater resources in Nebraska, U.S. are closely monitored by 23 Natural Resources Districts (NRDs) located across the state. Growers who use groundwater for irrigation are required to have flow meters installed at wells to monitor their water usage. However, many of these flow meters are still being read and recorded through in-person visits, which can be time-consuming and costly. Although some flow meters in Nebraska are monitored remotely by telemetry-enabled camera systems, yearly telemetry costs are high and making long-term operation financially burdensome. Using less expensive network protocol, such as Internet of Things (IoT), to transmit flow meter readings could enable new monitoring opportunities. However, there are challenges in directly transmitting flow meter images via IoT due to limited bandwidth. Therefore, in this study, we developed an algorithm using object detection deep learning techniques, i.e. You Only Look Once (YOLO) that can be programmed at an IoT node which can recognize readings from images of flow meters onsite before transmitting. The developed algorithm could significantly reduce data size and is essential for flow meter monitoring in an IoT network setting. The developed algorithm achieved 95.35% accuracy when recognizing 1,248 real-world flow meter images obtained at the courtesy of North Platte Natural Resources District (NPNRD) in western Nebraska. The framework and algorithm were also tested in a real-world scenario on a flow meter installed on a linear-move sprinkler irrigation system and showed promising results. By leveraging IoT and deep learning techniques, this research has the potential to revolutionize flow meter monitoring, reducing costs and improving efficiency in the management of groundwater resources in Nebraska, and potentially in other regions as well.

Introduction

Irrigation is critical to the success of crop production. In U.S., Nebraska is one of the most irrigated state, accounting for approximately 3.78 million ha of irrigated production land. Groundwater is vital to the well-being of Nebraskans and economic prosperity of the state. According to the 2018 Nebraska Groundwater Quality Monitoring Report [1], Nebraska has 96,593 active irrigation wells and 30,932 active domestic wells. Dynamic and real-time information on water quantity drawn from these wells is essential to water users and management agencies. Out of the 23 Natural Resources Districts (NRDs) in Nebraska, who regulates groundwater usage, 18 NRDs have mandatory requirements on installation of flow meters on all groundwater wells or at least on newly installed wells in their districts in order to monitor groundwater usage. Most of these flow meters require NRDs' staff to visit in person to record the numbers. Due to the large footprint that an NRD can cover, manually reading all flow meters can be extremely time and labor consuming, and therefore difficult to obtain frequent readings. For instance, the North Platte NRD (NPNRD) at the Nebraska Panhandle covers 1.21 million ha in 5 counties. Currently, NPNRD is remotely monitoring around 800 groundwater wells in Scotts Bluff County using a commercial flow meter monitoring system (Fig. 1). The system utilizes a

* Corresponding author.
E-mail address: xin.qiao@unl.edu (X. Qiao).

https://doi.org/10.1016/j.atech.2023.100236
Received 28 April 2022; Received in revised form 14 April 2023; Accepted 15 April 2023
Available online 16 April 2023

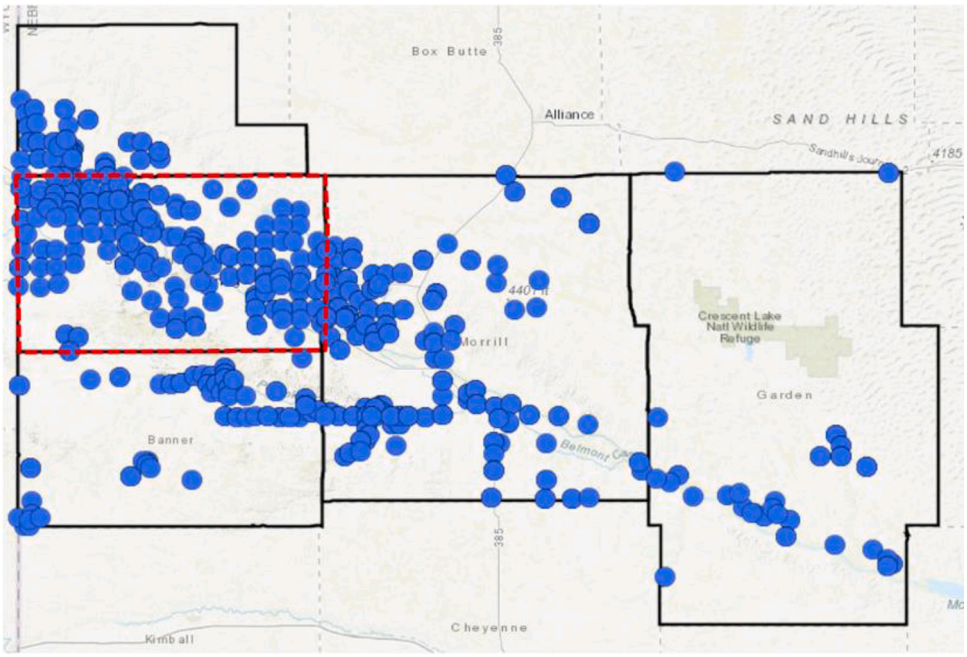

Fig. 1. Flow meters (blue circles) that North Platte Natural Resources District (NPNRD) monitor as of 2013. Red dashed line highlights Scotts Bluff County. Photo downloaded from NPNRD website.

camera and cellular telemetry equipped at each flow meter to send captured images back to be transcribed. Besides the initial cost of hardware, cellular telemetry subscription is charged per device per year (~$60, based on personal communication with NPNRD) resulting in a significant long-term expense. This also limits NRDs' ability to scale up their remote monitoring capability. Therefore, finding new ways to obtain reliable and frequent readings from flow meters remotely, yet low-cost to operate, is key to help Nebraska NRDs or other water management agencies to increase their monitoring capability and efficiency. To achieve such goals, it is key to use low-cost telemetry for real-time data transmission. Long range wide area network, or LoRaWAN, is one of the data transmission protocols that has been rapidly developing for many Internet of Things (IoT) applications. With one LoRaWAN gateway, it has the capability of connecting to large number of battery-powered sensors at low power consumption with a transmission range up to 15 km in suburban areas [2,3]. The limitation of LoRaWAN is that its data transmission rate is much lower than traditional telemetry, at a maximum speed of 27 kb/s [2] and therefore not suitable for directly transmitting large size data such as flow meter images. However, if images are pre-processed and stored onsite, and only processed results are transmitted, the size of data package would reduce from ~1000 kb to 2–3 kb which will be suitable for LoRaWAN.

Previous recognition techniques for detection of digits in meters are usually categorized into two groups: traditional methods and more recent convolutional neural network (CNN)-based methods. A traditional method utilizes computer vision and image processing techniques such as pixel projections [4], connected components [5], and matching a prior template Zhao et al. [6] to recognize readings from images. It is usually composed of multiple stages including 1) detecting and extracting a region of interest (ROI) which encloses the target digits from the input image, 2) segmenting subregions containing each individual digits from the ROI, and 3) recognizing these digits. To determine ROI, several methods have been developed to detect counters by utilizing pixel projections [4,6–8]. Specifically, a binarization of an input image is first computed, a ROI is then determined by counting the number of white pixels of the binary image along the vertical and horizontal directions, respectively. The major limitation of such method is that the accuracy of the detection highly depends on intrinsic characteristics of images. For instance, the layout and complexity of the image content, and features of the input image, such as the resolution, color contrast, distortion, as well as image qualities, etc. As these features can vary significantly in different application scenarios, pixel projection method usually only produces promising results under a determined analysis task with similar spatial layouts of digits and color characteristics. Once a ROI is determined, a traditional method subsequently segments the contours or silhouettes of individual digits from the ROI. Some techniques achieve segmentation by computing the connected components from the binarization of the ROI [4,5]. Alternatively, a support vector machine (SVM) can be used to compute the segmentation of digits [8,9]. By using this method, the binarization of the ROI is parameterized into a higher dimensional space, and then, a certain number of hyperplanes are computed by solving a predefined optimization equation to separate these hyperparameters into different classes, representing background and foreground. Unfortunately, both methods can easily fail when the input images are not from similar distribution, i.e. the characteristics of images vary significantly. The last stage of a traditional method is to recognize digits from the previously determined segmentations. Many techniques have been developed including template matching, SVM, and histogram of oriented gradient (HOG) [4,8–10]. Some of these methods produce promising results, but the accuracy relies extremely on the success of binarization and segmentation computed from the previous two stages. Therefore, they are not robust enough for more general application contexts. For this study, since flow meters are mostly installed at outdoor environment that are susceptible to dust, dew, abrasions, images taken from such complex scenes, e.g., various flowmeter types, different camera views, and messy environments, can be quite challenging to recognize using aforementioned traditional methods [11].

Beyond the traditional methods, more recently, another line of research employs the power of artificial intelligence techniques to automatically recognize readings of meters. Gao et al. [11] developed a two-stage framework by first adopting aggregated channel features to detect a ROI, and subsequently recognizing readings by combining CNN and bidirectional long short-term memory (LSTM) techniques [12]. While the author reported high accuracy using this method, the images were with high degree of clarity and similarity. In addition, it fails to

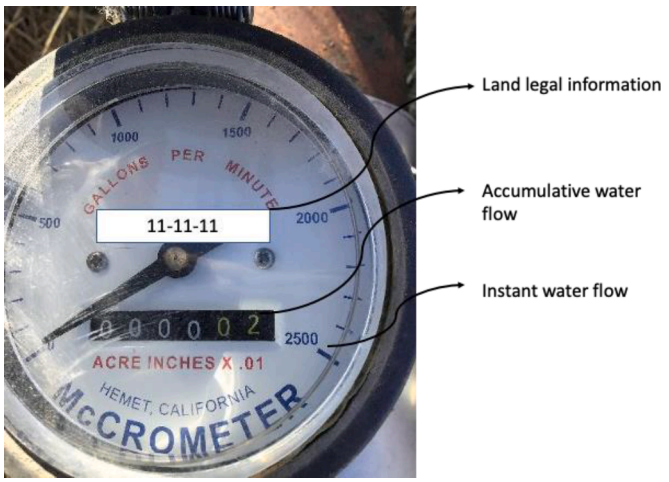

Fig. 2. An example of a McCrometer Flow Meter (McCrometer Inc., Hemet, CA, USA) installed in the North Platte Natural Resource District. Numbers on the dial include land ID at the top of the meter (pseudo land legal IDs are used here for privacy protection), accumulative water flow (ACRE INCHES X 0.01 as shown) in the lower center, and tick markers around edge of the meter which are used to read instant water flow (gallons per minute).

process rotated images. Similarly, Laroca et al. [13] proposed another two-stage framework, detecting ROIs of digits using the FastYOLO and recognizing the digits in the detected ROIs using CR-NET. Even though this method eliminates the segmentation stage, reducing the source of uncertainty of accuracy, and achieving promising results, this method employs two deep learning models and does not fit the IoT application context. The reason is that a computer node in an IoT system is usually not as powerful as a desktop or cloud computer node so that minimizing computational resources is essential for on-board computation and processing. Yet, initializing the two deep learning architectures proposed in [13] and loading and executing their corresponding trained models requires much memory and computational time. Alternatively, Gomez et al. [14] reported another CNN-based method without explicitly locating target digits. However, high accuracy can only be achieved by training a large dataset, i.e., 222,198 images in their study, which is not possible in this study with much more limited training dataset. Therefore, the objective of this study was to develop an algorithm using object detection and recognition deep learning techniques that is compact enough to be able to reside at a IoT node device and can execute to process images taken from flow meters onsite to significantly reduce data size to facilitate data transmission in an IoT network setting.

Materials and methods

Flow meter data and image labeling

In this study, 3248 flow meters images were obtained from NPNRD. The images were divided into two sets, including 2000 images for training and the remaining 1248 for validation. As the images were taken from real-world scenario, they post many challenges and would require good robustness for the recognition algorithm. The challenges included: the images were taken using multiple random camera devices, inducing variations of device-related factors such as focal length, angle of view, lens distortion, and photo resolution. In addition, the images were taken manually at varied angles, heights, distances, weather and lighting conditions, which in turn added variations to the image clarity, glare, completeness of showing a flow meter, and the area ratio between the number of pixels to the background of the flow meters. Fig. 2 shows an example of an analog-styled flow meter (McCrometer Inc., Hemet, CA, U.S.A.) that are commonly deployed in the study area. The flow meter contains digits with different information. At top of the flow meter, the sequence of numbers indicates land-specific information. At bottom of the flow meter, the black block contains numbers representing accumulative water flow. Finally, the flow meter also has tick marks all around for reading instant water flow rate. In this study, the target number is the accumulated water flow, which is what NRDs use for regulation purposes. The 2000 training images were manually labeled using an open-source tool: Yolo Mark (AlexeyAB) to train the deep learning model. Using Yolo Mark, a text file was generated for each image with information of individual digits, x and y coordinates of bounding boxes of these digits, and dimensions (width and height) of the bounding boxes. The coordinates and dimensions are relative to the width and height of the input images. For cases where the meter was

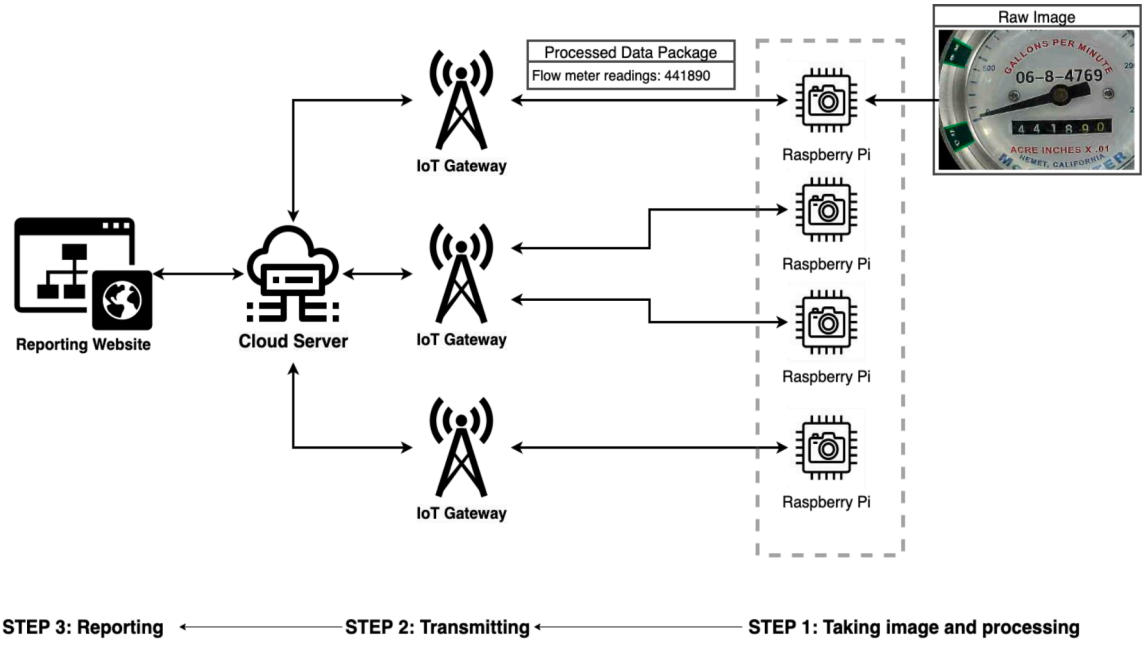

Fig. 3. . Proposed processing framework to recognize flow meter images in this study, where the Raspberry Pi-based node device will take images of flow meter, process images using developed algorithm, send data via LoRaWAN or other IoT communication protocols, and display results at a visualization platform.

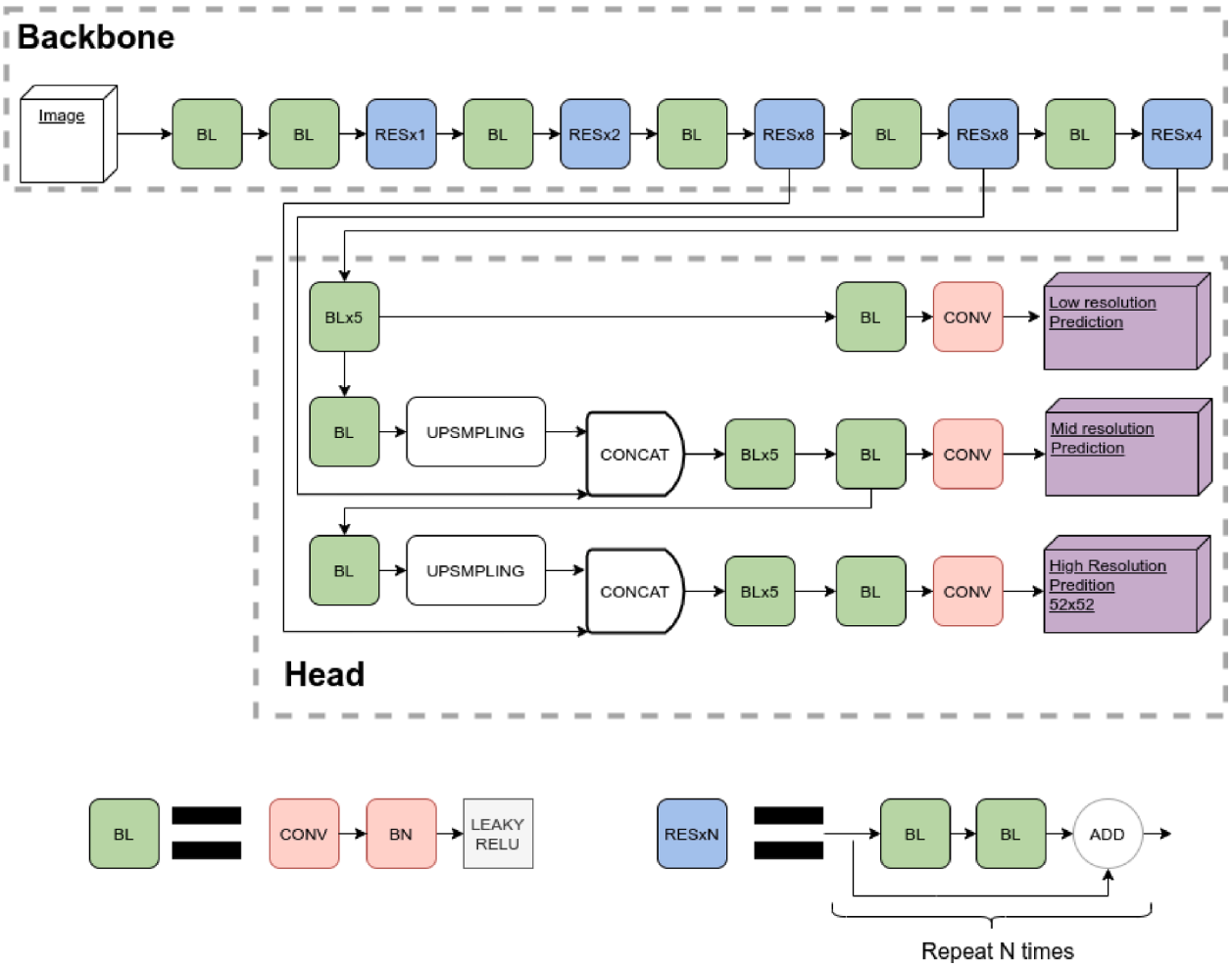

Fig. 4. . The model structure of YOLOv3. BL indicates a sequence of components including a convolutional layer, a batch normalization and a leaky ReLU activation function. RESxN indicates repeating RES combination N times, a RES represents a combination of two BLs and ADD indicates adding current results to the previous one.

rolling between two digits, two labels would be created (Fig. 6a).

The image recognition algorithm

Selection of deep learning model - You only Look Once (YOLOv3)

The image processing employed in this study has two stages: digits recognition and post-processing which includes filtering and rearrangement. The first step of the proposed algorithm is to automatically recognize all possible digits, including their actual value and locations from an input image. This was achieved by adopting the YOLOv3 deep learning model, which is one of the novel deep learning detection techniques implemented based on the Darknet framework [15]. YOLOv3 was chosen for several reasons. First, Darknet is an open-source neural network framework implemented using the C programming language. It can be easily compiled and executed on an operating system with a C compiler, which is a built-in for any Linux-based operating systems, e.g., Ubuntu, that is able to be installed on IoT node devices. Second, the Darknet is a light-weight framework. Its source code is sized about 15 Mb and it requires no additional libraries or packages for compiling and execution. Comparing to other deep learning frameworks such as the Caffe, Tensorflow, and Pytorch, which have heavy dependency on libraries, the Darknet is more flexible, resource-saving, and thus would be better suited on a IoT node device that has limited storage space and computation power. In this study, a Raspberry Pi computer was assumed to be the IoT node device to take and process flow meter images. The proposed framework is shown in Fig. 3, where the Raspberry Pi-based node device will take images of flow meter, process images using the developed algorithm, send data via LoRaWAN or other IoT communication protocols, and display results at a visualization platform. Physical constraints of the imagined IoT node devices were considered. Compared to cloud computer server or desktop computers, Raspberry Pi has less CPU cores and memories. It also lacks GPU devices that are needed by most deep learning techniques. As YOLOv3 can be executed in systems with CPU only, and for previously mentioned reasons, it was chosen as base of this recognition algorithm. Although YOLOv4 [16], a newer version of YOLO model, was released during development of this study, it was not adopted because the major improvements of YOLOv4 focus on increasing the accuracy of detecting multiple overlapping objects and it is not a concern with flow meter images.

Setup of backbone and head for YOLOv3

Like any modern deep learning-based object detectors, YOLOv3 has two components: a backbone and a head. A backbone is a specific design of deep neural networks pre-trained on ImageNet [17], which is a large image database. A backbone is responsible for extracting low level features from images such as lines and edges. The structure of the backbone of YOLOv3 is described in Fig. 4. where a pink box with CONV represents a convolutional layer. YOLOv3 designs a convolutional layer by utilizing Leaky ReLu [18] activation function and batch normalization [19] technique to improve the performance of the network. A convolutional layer, together with batch normalization and Leaky ReLu, form the base layers of the YOLOv3 architecture, which are represented

Fig. 5. a. a linear-move sprinkler irrigation system at Panhandle Research and Extension Center, University of Nebraska-Lincoln, Scottsbluff, NE, USA. b. an analog-style vertical flow meter. c. an AI-FlowCAM mounted on top of the flow meter. d. an image taken by AI-FlowCAM.

using green boxes in Fig. 3. Blue boxes, denoted as RSExN, represent residual blocks with two base layers that have been repeated N times [20]. As this backbone has 53 convolutional layers, it is referred to as Darknet-53 by its original author [21].

Subsequent to a backbone, several additional layers are designed to predict different object classes. In this study, the object classes include digits of the flow meter readings and their bounding boxes. These additional layers are usually referred as a head of a neural network (Fig. 4). The head of YOLOv3 is enclosed by a gray dashed box in Fig. 4. While most alternative heads predict classes (digits in this case) and bounding boxes through two separate routines, YOLOv3 predicts them in a single process, significantly speeding up the prediction. Specifically, it simultaneously predicts b_x, b_y, b_w, b_h, t_o, and t_c, where b_x, b_y, b_w, and b_h are the four coordinate offsets of a bounding box relative to the top left corner of the input image, t_o indicates the objectness, i.e. the possibility of existences of an object prediction of this bounding box, and $t_c$indicates the class prediction if there is an object detected in this bounding box. In addition, to increase the accuracy of detections, YOLOv3 predicts the detections across three different resolution scales, combining both the global context and fine-grained features of the images. As shown in Fig. 3, the low-resolution prediction only uses information from the last layer of the backbone, while the mid and high-resolution predictions integrate coarse-grained information computed from early layers. For each individual scale, it also predicts three bounding boxes with different sizes. Therefore, the output of each scale is a 3d tensor with a depth of $[3x(4 + 1 + N)]$, where 3 represents three bounding boxes, 4 represents the four offsets of a bounding box, 1 indicates the objectness prediction, and N represents the number of possible classes. In this implementation, as it has 10 different digits, i.e., 10 different classes, N equals 10, and thus, the depth of the 3d tensor is 45. Another important configuration of utilizing YOLOv3 is to determine the bounding box priors at the three resolution scales. This study started with nine priors computed via the k-means clustering method proposed in [15] by using the distance metric relative to the intersection of union of bounding box centroids, and adjusted these priors based preliminary experimental results. The final priors are (10,13), (16,30), and (33,23) for a low-resolution scale, (30,61), (62,45), and (59,119) for a middle-resolution scale, and (116,90), (156,198), and (373,326) for a high-resolution scale, each pair represents the width and height priors of a bounding box.

Post-processing: Filtering and rearrangement

As mentioned earlier, the digits that are most interested to this project are accumulative water flow. A filtering process was created to filter out non-necessary digits. First, the filtering process looped through all the raw detections and merged close ones in terms of the intersection over union (IoU) of their bounding boxes. Specifically, given two bounding boxes B0 and B1, the IoU between them is defined as $IoU = \frac{area(B_0 \cap B_1)}{area(B_0 \cup B_1)}$. For each pair of bounding boxes, if their IoU was greater than a predefined threshold, which is 0.1 in this implementation, they were merged using the one with a higher prediction score. Subsequently, the merged predictions were spatially sorted from left to right along the x direction to form the accumulative flow meter reading.

Implementation in real-world

To assess the performance of the framework and algorithm in real-world conditions, an analog-style flow meter (McCrometer Inc.,

Table 1
. Recognition results of individual digits. The first column indicates the class of digits. Accuracy refers to the fraction between the number of correct recognitions of a digit to the total number of recognitions of this digit. Recall is defined as the fraction between the number of correct recognitions of a digit and the total frequency of this digit in the dataset. The fourth column indicates average accuracy which is computed using the area under curve approach with IoU of 0.5 [22]. The last column indicates the frequency of a digit in the dataset.

Digit	Accuracy (%)	Recall (%)	Average Accuracy (%)	Frequency
0	95.45	98.64	98.71	1107
1	91.36	94.75	94.41	781
2	95.30	96.28	96.72	779
3	91.35	96.20	96.12	736
4	95.69	96.96	97.79	756
5	92.04	97.67	97.58	687
6	94.18	95.75	97.28	659
7	94.78	95.66	96.52	645
8	95.21	96.57	97.56	700
9	94.41	97.91	98.76	621
Average	93.78	96.64	97.15	

Hemet, CA, USA) was installed on a linear-move sprinkler irrigation system (Lindsay Corporation, Omaha, NE, USA) at the Panhandle Research and Extension Center of the University of Nebraska-Lincoln. The research site is located in Scottsbluff, NE, USA, with an average elevation of 1189 m and GPS coordinates of 41°53′34.93″N, 103°41′2.04″W. The irrigation system travels in the direction of south to north. To capture data from the flow meter, an edge-computing camera unit, named AI-FlowCAM, was programmed with the developed algorithm and installed atop the flow meter using a customized enclosure and 3D-printed mounts on the west side of the irrigation system where water inlet is located. The AI-FlowCAM was comprised of a Raspberry Pi 4 computer (Raspberry Pi Foundation, U.K.), a battery, a PiJuice HAT power management board (Pi Supply, U.K.) for managing sleep and wake schedules, a solar panel, a Raspberry Pi 8 megapixel RGB camera (2592 × 1944 pixels), Raspberry Pi LoRa node pHAT (Pi Supply, U.K.), and an external 915 MHz LoRa antenna (Laird Connectivity, Akron, OH, USA). A Sentrius RG1xx LoRaWAN-enabled gateway (Laird Connectivity, Akron, OH, USA) was set up on the rooftop of a nearby office building as the receiving gateway. Flow meter readings captured by the AI-FlowCAM, as well as LoRa transmission parameters such as received signal strength indicator (RSSI), signal-to-noise ratio (SNR), and airtime, were recorded and stored on a cloud server. The flowmeter readings recognized by the AI-FlowCAM were also verified manually by human inspection. Fig. 5 shows the setup and an image taken by AI-FlowCAM (Fig. 5d).

Results and discussion

Model training results

The proposed model was trained and evaluated on the computer cluster at Holland Computing Center (HCC) at the University of Nebraska-Lincoln. The training was completed on a computer node with 32 CPU cores, each node has 4 GB memory. To speed up the training process a NVIDIA V100 GPU with 32 GB memory was utilized. The complete training process took about 14 h. The recall of training was 0.92, and the mean average precision, with IoU threshold setting at 50%, is 93.59%, the average lose was 0.288924. To maximize the performance, the batch size was set to 64, image resolution was set to 416 × 416, and learning rate was set to 0.01. When training a detection model, YOLO first normalizes all training images of different resolutions into a unified resolution. It is noticed that the accuracy of the model was decided relative to this unified resolution. Setting this parameter to an inappropriately small number will downscale the training data significantly, losing too much information and resulting in a low recognition accuracy. On the other hand, as the study dataset has a wide variation in image resolutions, setting this parameter to a very large number will induce interpolation errors when up-sampling small images and increase the uncertainty of the model. This study finally selected 416 × 416 for the unified resolution, which led to the best accuracy.

Table 2
. Recognition results of complete accumulative water flow readings.

	Percentage (%)	Number of cases
Correct recognitions	95.35	1190
Incorrect recognitions		
Missed 1 digit	3.69	46
Missed multiple digits	0.96	12
Total	100	1248

Model validation results - Quantitative evaluation

A correct recognition case is considered only when all digits of accumulative water flow reading are correctly recognized. The validation was done on using the AI-FlowCAM at a lab setting. Processing each individual image took 6–10 s. This system was first evaluated by examining the performance in terms of recognizing individual digits. The results are summarized in Table 1, where the digit indicates the class of digits; the precision is defined as the fraction between the number of correct recognitions of a digit and the total number of recognitions of this digit; the recall is defined as the fraction between the number of correct recognitions of a digit and the total frequency of this digit in the dataset; the average precision (AP) is computed using the area under curve approach with IoU of 0.5 [22] and the frequency indicates the frequency of a digit in the dataset. The last row reports the mean of the precision, recall and AP, respectively.

As reported in Table 1, the high accuracy suggests that this algorithm is very likely to make a valid prediction of the target digits; while the high recall rate indicates that these recognitions cover the ground truth of individual digits with a high degree of completeness. The AP and the mean of AP (mAP) are widely adopted as an evaluation metric in the community of objects detecting, and the results indicate that this recognition system obtains a superior balance between validity and completeness. The algorithm was then tested on the 1248 validation flow meter images in terms of detecting accumulative flow meter readings (all digits). The results are shown in Table 2. It successfully recognized readings of 1190 images, resulting in an accuracy of 95.35%. Among the 58 failed cases, 46 of them missed 1 digit while 12 of them missed multiple digits.

Model validation results - Qualitative evaluation

In addition to evaluating the system using quantitative metrics, a qualitative analysis was also conducted. This system was able to successfully process images that have wide variations in many aspects, making it more robust in both the real-world application this work is investigating and other potential generalized scenarios. Such robustness is demonstrated in Fig. 6, where each image presents a difficult situation to be processed by alternative techniques or even read by a human-being but are accurately recognized using this system. Because of the data privacy restrictions, the device IDs have been masked using gray boxes. The robustness of this algorithm is reflected in different aspects. First, the proposed algorithm was able to successfully process images with incomplete information. As shown in Fig. 6a, the last digit is between 4 and 5 and these two numbers are only partially seen, but the system developed here recognized both two digits correctly. Fig. 6d shows another difficult case where the surface of a flow meter is worn and the area containing the target digits is blurred. Second, this system is robust in recognizing images with different foreground-background ratio, i.e. the ratio between the number of pixels of a flow meter relative to the background. For instance, Fig. 6a, 6d, and 6f have larger foreground-

Fig. 6. . Examples of successfully recognized accumulative water flow (digits in black box on the dial) in the flow meter images. Caption under each image shows recognized readings from the proposed algorithm.

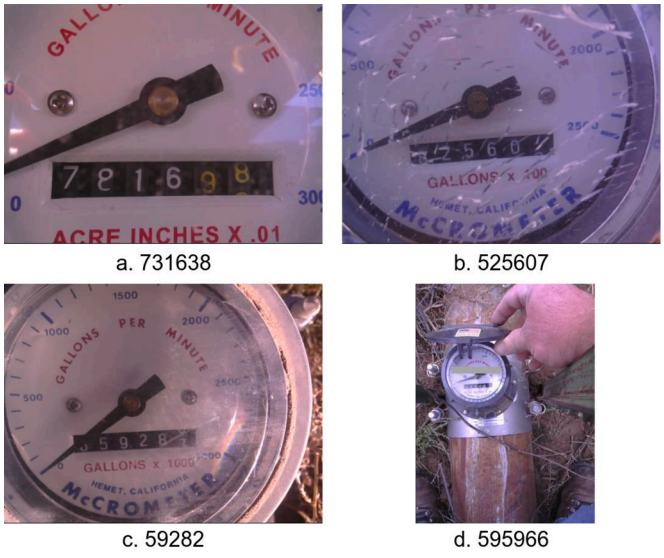

Fig. 7. . Examples of failed recognized accumulative water flow (digits in black box on the dial) in the flow meter images. Caption under each image shows recognized readings from the proposed algorithm.

background ratios, while in Fig. 6b, 6c, and 6e have smaller foreground-background ratios. In addition, the flow meter images were taken under different lighting conditions so that there were variations in color hues of the images. For example, Fig. 6a and 6b appear to have a yellow tint while Fig. 6c had a blue tint and 6d and 6f had white tint. Furthermore, this algorithm was able to extract numbers from blurry images, which are frequently encountered in a real-world application, and make accurate predictions. This was shown in Fig. 6e, where the image had low resolution, and the digits of the reading were hard to read even if it is enlarged to be read by human. Lastly, the proposed system was accurate when recognizing images taken from a rotated angle (Fig. 6f). Alternative techniques usually process such case by first extracting long horizontal or vertical lines from the image, and subsequently use them as references to rotate the image by a degree so that the digits block is parallel to the truly horizontal line (x axis). Accuracy of this approach is sensitive to the a priori assumption of the references and can easily fail when contents of images are complicated. In contrast, the proposed approach in this study was able to directly make the correct recognition without computing the rotation angle, suggesting the generality and efficiency of this approach. The success of predicting these images indicates that the proposed algorithm is not only capable in a specific controlled environment but robust in a wide spectrum of real-world application scenarios.

Discussion of failed cases

Beyond the successful cases, this work also analyzed the incorrect recognitions and summarized them into four categories. The first is shown in Fig. 7a. Although this image is in good resolution and the contents are seen clearly, some of the digits were covered by dirt so that they appear to be incomplete or confused with digits. Such noises mislead the algorithm to make false predictions, e.g., the second digit 8 and fifth digit 9 were both recognized as 3. The second category is that the flow meter was shattered heavily where the cracks were on top of the digits, making it extremely difficult to distinguish them visually (Fig. 7b). In this example, the first digits 8, which is white color, are covered by scratches which also appear to be white in the image. This information is mixed in such a way that the features of this area extracted by the deep neural network may not match the features of other areas that may contain digit 8, and lead to incorrect reading as 5. The third category is that a large portion of the first digit was blocked by the meter pointer needle, as shown in Fig. 7c, in which case even a human being is not able to recognize it correctly. However, in such cases the first digit could be estimated according to the previous reading in real applications so it would not be a major concern. The last case is that the digits are too small in the image to be recognized, as shown in Fig. 7d. In this situation the information of the area containing these digits may quickly be eliminated or diluted when the tensor encoding this image is convoluted in the first couple of convolutional layers. The

Table 3
Performance summary of AI-FlowCAM during the 10-day real-world testing period.

	Percentage (%)	Number of cases
Correct recognitions	73.8	79
Incorrect recognitions	26.2	28
Other performance parameters		
Average runtime (seconds)	9.16	
Average RSSI	−51	
Maximum SNR	12	
Minimum SNR	6	
Average airtime (seconds)	0.051	

loss of information results in a high degree of uncertainty to make the recognition. These challenges could be resolved or mitigated by having fixed camera type, lighting condition, and mounting location to achieve standardized images.

Real-world evalulation results

During a 10-day evaluation on a linear irrigation system, AI-FlowCAM was installed and tested. The camera was programmed to capture images and send data at hourly intervals, resulting in a total of 262 images collected. Out of these images, 107 were deemed valid as they had distinctive flow meter readings during irrigation, while 155 were considered invalid due to repetitive readings. Repetitive images with the same flow meter reading were not counted towards the final performance statistics. All image readings were successfully sent and received by the gateway. The average processing time for each image was 9.16 s using the configuration of AI-FlowCAM. The average airtime to send the processed image data from AI-FlowCAM to the gateway was 0.051 s, with an average RSSI of −51, and a maximum and minimum SNR of 12 and 6, respectively. Out of the 107 valid images, 79 images were recognized correctly, accounting for 73.8% of the total images, while 28 images were not recognized correctly, representing 26.2% of the total images. Among the 28 images, 24 images were able to recognize the first 4 digits but were missing the last 2 digits, while 4 images reported no detection even though images were taken by the camera. The issue of missing digits could be attributed to the fact that the colors of the testing flow meter were different from the colors of the training images. The training images had flow meter with white background with a black box containing the digits, whereas the test flow meter had a deep blue background, causing difficulties for the algorithm to locate the correct area containing the digits, especially when the irrigation was running, and the digits were rolling on the flow meter. The reason for the 4 images reporting zero reading despite successfully capturing images is unclear and could possibly be due to a busy CPU when processing the images. Table 3 provides a summary of the performance of AI-FlowCAM during the testing period.

The testing period for the proposed framework was limited to only 10 days due to freezing temperatures. However, during this period, it was demonstrated that the framework has the potential to work effectively in real-world scenarios. It's worth noting that no graphical user interface (GUI) was designed to visualize the readings during this brief testing period. For a similar setup involving the design and testing of an edge-computing camera sensor for capturing crop canopy cover images, processing the images onsite, and transmitting the processed images using LoRa to a website-based GUI, readers can refer to Liang et al. [23] (Fig. 8).

Conclusion

This paper presents an approach for automatic recognition of accumulative water flow readings for analog-style commercial flow meter, specifically for irrigation water usage monitoring within an agricultural

Fig. 8. An example of an edge-computing camera node using framework proposed in this study that was used to capture image, process image, and transmit processed results in Liang et al. [23]. Figure shows the camera node was able to capture canopy cover development at a dry edible bean field during 2-month testing period. This figure and sensor reading can be accessed at: https://phrec-irrigation.com/#/f/58/sensors/236.

IoT application scenario. The approach involves training a YOLOv3-based deep learning model to recognize potential digits and their bounding boxes in input images. Subsequently, a process is developed to filter out unnecessary digits obtained from the deep learning model and rearrange the remaining digits to form the correct reading. The system achieves a recognition accuracy of 95.35% using 1248 real-world flow meter images, while remaining lightweight and efficient, making it suitable for deployment and execution on IoT node devices, such as Raspberry Pi. Qualitative evaluation indicates high robustness of this approach, although there may be cases where it is not able to recognize numbers due to dirt, water, or meter pointer obstructions in real-world applications. The framework and algorithm were also tested in a real-world scenario on a flow meter installed on a linear-move sprinkler irrigation system, showing promising results. However, future improvements may be needed to address issues related to differences in colors between the flow meter in real-world scenarios and training images. Overall, the proposed IoT and edge-computing-based approach has the potential to significantly reduce monitoring costs for irrigators and government agencies, while increasing their monitoring capacity. Furthermore, it can be integrated with other sensors to inform irrigation management decisions.

Declaration of competing interest

None.

Data availability

The authors do not have permission to share data.

Acknowledgement

We appreciate the support from North Platte Natural Resources District for providing the flow meter images. We also would like to acknowledge the funding support from USDA with Award No. 2019–67021–29227.

References

[1] Nebraska Groundwater Quality Monitoring Report, Nebraska Department of Environmental Quality, 2018. https://nebraskalegislature.gov/FloorDocs/106/PDF/Agencies/Environment_and_Energy__Nebraska_Department_of_/702_20201130-150859.pdf. Accessed on March 9th, 2022.
[2] Lora Alliance, 2015. https://lora-alliance.org/resource_hub/what-is-lorawan/.
[3] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, T. Watteyne, Understanding the limits of LoRaWAN, IEEE Commun. Mag. 55 (9) (2017) 34–40.
[4] A. Anis, M. Khaliluzzaman, M. Yakub, N. Chakraborty, K. Deb, Digital electric meter reading recognition based on horizontal and vertical binary pattern, in: 3rd International Conference on Electrical Information and Communication Technology, 2017, pp. 1–6.
[5] M. Cerman, G. Shalunts, D. Albertini, A mobile recognition system for analog energy meter scanning, in: International Symposium on Visual Computing, 2016, pp. 247–256.
[6] S. Zhao, B. Li, J. Yuan, G. Cui, Research on remote meter automatic reading based on computer vision, in: IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific, 2005, pp. 1–4.
[7] D. Shu, S. Ma, C. Jing, Study of the automatic reading of watt meter based on image processing technology, in: 2nd IEEE Conference on Industrial Electronics and Applications, 2007, pp. 2214–2217.
[8] V. Edward, C. Paul, Support vector machine based automatic electric meter reading system, in: IEEE International Conference on Computational Intelligence and Computing Research, 2013, pp. 1–5.
[9] I. Gallo, A. Zamberletti, L. Noce, Robust angle invariant GAS meter reading, in: International Conference on Digital Image Computing: Techniques and Applications, 2015, pp. 1–7.
[10] L.A. Elrefaei, A. Bajaber, S. Natheir, N. AbuSanab, M. Bazi, Automatic electricity meter reading based on image processing, in: IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, 2015, pp. 1–5.
[11] Y. Gao, C. Zhao, J. Wang, H. Lu, Automatic watermeter digit recognition on mobile devices, in: International Conference on Internet Multimedia Computing and Service, 2017, pp. 87–95.
[12] H. Sak, A.W. Senior, F. Beaufays, Long short-term memory based recurrent neural network architectures for large covabulary speech recognition, arXiv preprint (2014). https://arxiv.org/abs/1402.1128.
[13] R. Laroca, V. Barroso, M.A. Diniz, G.R. Gonçalves, W.R. Schwartz, D. Menotti, Convolutional neural networks for automatic meter reading, J. Electron. Imag. 28 (1) (2019) 13–23.
[14] L. Gómez, M. Rusinol, D. Karatzas, Cutting Sayre's Knot: Reading Scene Text Without Segmentation. Application to Utility Meters, 13th IAPR International Workshop on Document Analysis Systems, 2018, pp. 97–102.
[15] Redmon, J., and Farhadi, A., 2018. YOLO: real-time object detection. https://pjreddie.com/darknet/yolo/.
[16] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, YOLOv4: optimal speed and accuracy of object detection, arXiv preprint (2020). https://arxiv.org/abs/2004.10934v1.
[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Wang, Imagenet: a large-scale hierarchical image database, in: IEEE conference on computer vision and pattern recognition, 2009, pp. 248–255.
[18] B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network, arXiv preprint (2015). https://arxiv.org/abs/1505.00853.
[19] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, 2015.
[20] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 *IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[21] Redmon, J., 2013. Darknet: open source neural network in C. https://pjreddie.com/darknet/.
[22] M. Everingham, L. Van Gool, C.K.I. Williams, The PASCAL visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2010) 303–338, https://doi.org/10.1007/s11263-009-0275-4.
[23] W. Liang, J. Oboamah, X. Qiao, Y. Ge, B. Harveson, D.R. Rudnick, J. Wang, H. Yang, A. Gradiz, CanopyCAM – an edge-computing sensing unit for continuous measurement of canopy cover percentage of dry edible beans, Comput. Electron. Agric. 204 (2023) 2023.