Research and Implement Embedded Artificial Intelligence in Low-Power Water Meter Reading Device

Hoan Nguyen Duc
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
nguyenduchoan.space@gmail.com

Thao Nguyen Manh
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
thaonm@uit.edu.vn

Huy Trinh Le
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
huytl@uit.edu.vn

Fabien Ferrero
CNRS/LEAT
Université Cote d'Azur
Sophia Antipolis, France
fabien.ferrero@univ-cotedazur.fr

***Abstract*—This paper presents a system using artificial intelligence deployed on ESP32-Cam to conduct OCR on water meter readings. Data transmission through LoRa technology helps reduce overall system power consumption. The accuracy of digit classification tasks reaches up to 98%. The average current consumption in active and sleep mode is 107 mA and 1 uA, respectively. With these specifications, the proposed system is proved to be low-power, low-cost and has a long-lasting operating time.**

***Index Terms*—LoRaWAN, Water Meter, Deep Learning, Low Power, Image Processing In C/C++**

I. Introduction

In the past, high resource-demanding and complicated tasks such as AI algorithms are often done on high performance computing servers. With advanced development in open source software and IC design technology, more and more embedded boards with complex computing power are created. These integrated circuits are not only low-power, low-cost but also designed to adapt and deploy high accuracy signal processing techniques or complicated algorithms without the Internet connection. These advanced innovations open a new opportunity to run AI algorithms - here also known as inference - directly on embedded devices with limited resources.

There are several papers about smart water meters. In [1] and [2], reed switchs are used to gather meter data. This technique requires physical invasion into the water meter. Therefore it is hard to deploy in some areas with strict local policies. Article [3] replaces the traditional mechanical water meter (read by a human operator) with a digital one, leading to an increase in replacement cost. These disadvantages are resolved in [4]. This paper captures an image of the water meter's dial face and then sends it to the cloud to process. This solution does not require changes to the water meter's structure. Digit classification accuracy is high with the help of powerful cloud computing. However, the system requires a stable internet connection which leads to an increase in power and processing time. Cloud servers can have vulnerabilities that pose concerns to the privacy and integrity of data. Researches [5], [6] overcome above limits. However, study [5] has a low accuracy with underexposed or overexposed captured images. These studies only show results about OCR tasks without long-range data communication method or low-power design. Therefore, the overall power consumption is not optimized, and the lifetime of the system will decrease. This drawback limits the ability to deploy in widespread use.

A good water meter reading device has to balance energy consumption, image processing - AI tasks, and communication ability. Datalink also needs high stability, wide range coverage, high capacity while still low power. There are low-power wireless standards commonly used, such as BLE, Zigbee, ANT+, LoRa,... [7]. BLE is a widely-used low-power wireless standard [8]. Zigbee is designed as a range-scalable wireless standard with mesh protocol [9]. ANT+ is a low-power protocol based on a proprietary protocol which is used for monitoring application, especially for the multicast scheme [10].

LoRa is Semtech's technology for low-power, wide area networks (LPWANs). This modulation technique is known for low power consumption and wall penetration. Besides, this technology has some advantages such as high capacity potential, robustness against interference and noise,... [11]–[13]. However, LoRa has low data rates so applications using this technology has to optimize their packet size. In [14], the authors design an optical reader kit for a traditional water meter. This device takes a snapshot of the meter's consumption wheel, converts to grayscale and transmits the image to a LoRa Gateway. Although being preprocessed, a grayscale image still has a large data size. Transmitting this kind of data through

LoRa technology will lead to a significant increase in transmit time, packet dropout rates, and a decrease in the reliability and lifetime of the system.

This paper proposes utilizing some image processing algorithms and a custom convolutional neural network (CNN) for embedded devices. These techniques are deployed on ESP32-cam to conduct OCR on water meter readings before sending to LoRa Gateway through LoRaWAN protocol. ESP32-Cam plays a role as a single-end device in a LoRa-adopted low-power wide area network.

II. Solution

A. Hardware Design

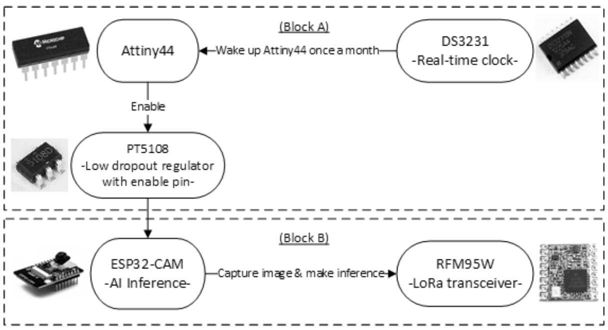


Fig. 1: Hardware design

The system in Fig. 1 is controlled by two microcontrollers. Therefore the device is devided into two function blocks.

<u>Block A:</u> Time management and power control unit

- Attiny44 uses I2C to set the alarm time in IC DS3231. DS3231 interrupts Attiny44 to wake up once a month. Then, Attiny44 sleeps in the remaining time.
- Attiny44 controls the enable pin of IC PT5108 to turn on/off the power of block B.

<u>Block B:</u> Image preprocessing and AI inference unit

- ESP32-Cam takes photos, preprocesses, and saves them to MicroSD card. This set of images is the dataset for training the CNN on Google Colab.
- A deep learning model trained on Google Colab is put into ESP32-Cam memory. ESP32-Cam takes a photo, preprocess, and uses this model to make an inference.
- ESP32-Cam uses SPI with RFM95W board to send out predicted water meter readings. A LoRa Gateway collects these data from all devices within a range of up to many kilometers and then sends them to the internet server of The Things Network (TTN).

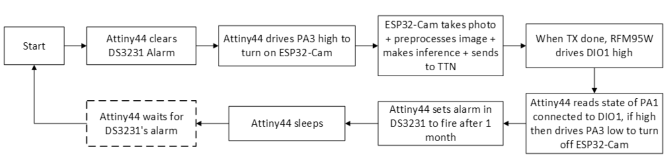


Fig. 2: Operating flowchart

Fig. 2 presents the operating flowchart of the device. The first system start-up date is used as wake-up time in the following months. Attiny44 clears the alarm flag in DS3231, if it is available. The enable pin of LDO PT5108 is driven to high state to turn on the power of block B. ESP32-Cam takes and preprocesses photo, makes an inference, and then sends to The Things Network server through LoRa Gateway.

When the transmission is done, Attiny44 then turns off PT5108's enable pin to cut down the power of block B. Attiny44 reads time stored in DS3231, sets the alarm for one following month, and goes to sleep. When there is an alarm signal from DS3231, Attiny44 wakes up, clears the alarm flag, and the loop starts again.

B. Image Preprocessing

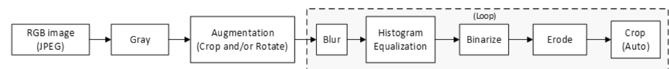


Fig. 3: Image preprocessing flowchart

Fig. 3 illustrates the operation of preprocessing image. ESP32-Cam captures an image of the water meter's dial face in JPEG format at 1024x768 size. Image is converted to grayscale. The image is cropped and rotated to the correct angle as shown in Fig. 4. Then five distinct digit images of water meter reading are cut out in the loop to preprocess.

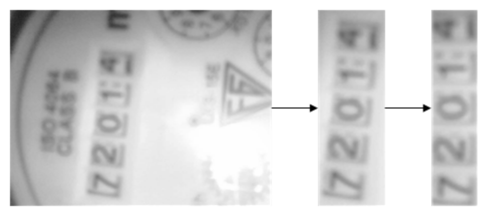
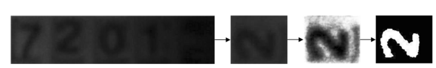
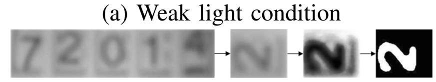


Fig. 4: Image augmentation

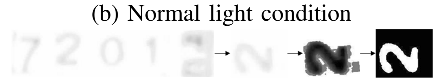
Each digit number is blurred to reduce contrast noise in the histogram equalization step. This paper resolves the limit mentioned above when classification accuracy drops to about 70-80% in different light conditions. Histogram Equalization algorithm helps balance contrast effectively [15]. Image features remain almost the same in different light conditions as shown in Fig. 5. Image is binarized after going through the histogram equalization step.



(a) Weak light condition



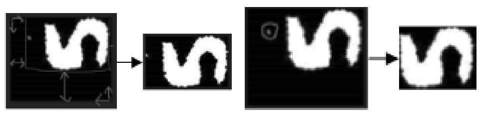
(b) Normal light condition



(c) Strong light condition

Fig. 5: Effect of histogram equalization in different light conditions

Erosion helps reduce minor white noise on the image. The black background, which is considered a redundant feature of the image is cropped out automatically in Fig. 6.



(a) Crop without erosion (b) Crop with erosion

Fig. 6: Effect of auto cropping algorithm with & without erosion

Images in this final step are used as a training dataset in Google Colab. Image captured by ESP32-Cam is preprocessed to this state before making inference.

C. Convolutional Neural Network

TABLE I: Proposed Convolutional Neural Network

Type	Num of kernels	Size/Stride/Padding	Activation
Input (width x height x channel): 40 x 30 x 1			
Convolution 1	4	4 x 3 / 1 / valid	relu
Dropout (rate = 0.1)			
Max Pooling		2 x 2 / none / valid	
Convolution 2	2	2 x 2 / 1 / valid	relu
Max Pooling		2 x 2 / none / valid	
Convolution 3	4	3 x 4 / 1 / valid	relu
Dropout (rate = 0.2)			
Max Pooling		2 x 2 / none / valid	
Flatten			
Dense (units = 80, activation = relu)			
Dense (units = 40, activation = softmax)			

The network in Tab. I consists of two convolutional layers and one max-pooling layer. Input is resized to 40 x 30 x 1 to reduce ram usage since ESP32-Cam is ram-limited. There are only three convolutional layers, so the network uses two fully connected layers (dense) to increase neural connections, which helps increase the model's performance. The network uses Dropout layers to prevent overfitting. Sparse Categorical Crossentropy loss function with Adam optimizer is used for the network. The learning rate is 0.001 and the batch size is 64. The network model consumes 99.4 KB on ESP32-Cam's flash memory.

The output of H-CNN represents 40 classes for integers (0, 1, 2, ..., 9), half-digit values between each consecutive number (0b, 1b, 2b, ..., 9b) and values between integers and half-digits (0a, 0c, 1a, 1c, ..., 9a, 9c) as shown in Fig. 7.

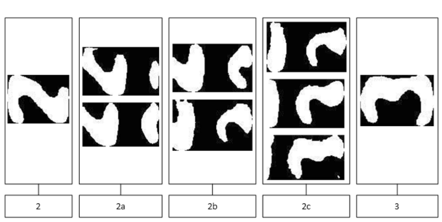


Fig. 7: Example for digit labels from 2 to 3

ESP32-Cam captures 2744 preprocessed samples. The ratio of the training/validate/test set is 70%:15%:15% per class. This dataset is imbalanced since integer images are captured more frequently than the others (e.g. half digits,...). In order to get high accuracy on this model, image augmentations (erosion, dilation, width/height shifting) are applied on the training set to balanced the number of samples per class. The final number of the three training/validate/test sets are 8680, 411, 412, respectively.

III. Result

A. Deep Learning Model Performance

The average accuracy score on the test set of the network in Tab. I is 98,204% with a standard deviation of 0.929%. Fig. 9 show the training graph and distribution box of accuracy scores with whisker plot of Tab. I.

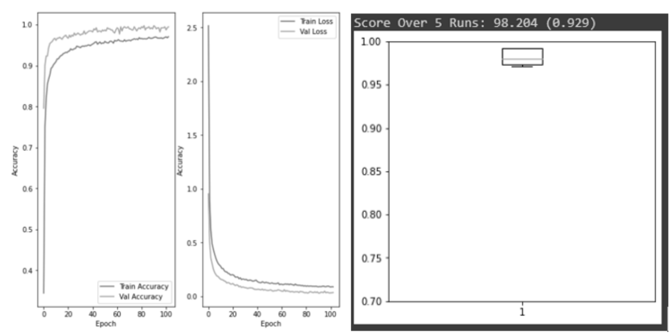


Fig. 8: Training graph, distribution of accuracy scores

B. Power Consumption

TABLE II: Average current consumption per operating mode

Operating mode	Supply voltage	Operating time	Min current	Avg current	Max current
Active	3.7 V	6.9 s	33 mA	108 mA	214 mA
	3.3 V	6.9 s	33 mA	107 mA	210 mA
Sleep	3.7 V	1 month	0.53 uA	1.42 uA	2.42 uA
	3.3 V	1 month	0.21 uA	1 uA	1.95 uA

The device is powered by an input voltage of between approximately 3V and 4V. Therefore measures in Tab. 2 are taken between these voltage levels. At 3.7V, the device consumes 108 mA in active mode. The lowest current is 0.53 uA in sleeping mode, and the highest is 2.42 uA, with the overall average current consumption is 1.42 uA. At 3.3V, the device consumes 107 mA in active mode. The lowest current is 0.21 uA in sleeping mode, and the highest is 1.95 uA, with the overall average current consumption is 1 uA. The above measurement is taken multiple times separately to calculate the total average values. The Tab. 3 and Fig. 9 below indicate power per task at one-time measurement only.

TABLE III: Average current consumption per task

Number	Task	Time	Avg current
0	Boot device	1.15 s	60 mA
1	Initialize TensorFlow and flashlight driver	10 ms	63.9 mA
2	Initialize camera driver	620 ms	85 mA
3	Capture image	140ms	126 mA
4	Convert image frame to RGB888	1.67 s	129 mA
5	Convert image to grayscale	390 ms	128 mA
6	Crop out zone around digit area	110 ms	125 mA
7	Rotate image	360 ms	126 mA
8	Crop out image of 1^{st} digit	10 ms	129 mA
9	Blur 1^{st} digit	100 ms	129 mA
10	Equalize histogram, binarize, erode 1^{st} digit	30 ms	129 mA
11	Auto crop out redundant black background of 1^{st} digit	10 ms	129 mA
12	Make inference for 1^{st} digit	40 ms	129 mA
13	Crop out image of 2^{nd} digit, preprocess the same steps as 1^{st} digit, make inference	190 ms	128.4 mA
14	Crop out image of 3^{rd} digit, preprocess the same steps as 1^{st} digit, make inference	190 ms	128.25 mA
15	Crop out image of 4^{th} digit, preprocess the same steps as 1^{st} digit, make inference	190 ms	128.2 mA
16	Crop out image of 5^{th} digit, preprocess the same steps as 1^{st} digit, make inference	190 ms	127.8 mA
17	Save time slots to MicroSD card	90 ms	129 mA
18	Initialize LoRaWAN LMIC driver	30 ms	127 mA
19	Send LoRaWAN packet to gateway in ABP	1.38 s	97.8 mA
20	Sleep	1 month	1.1 uA

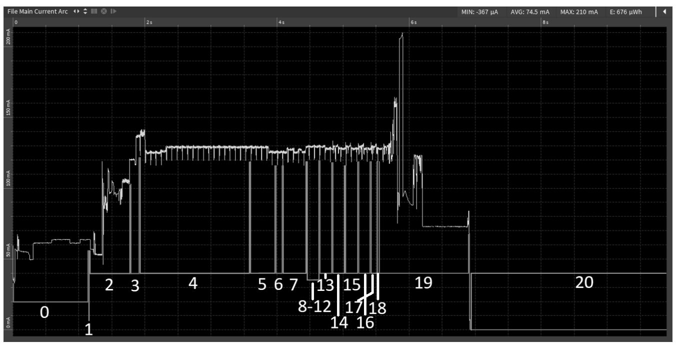


Fig. 9: Power flowchart per task

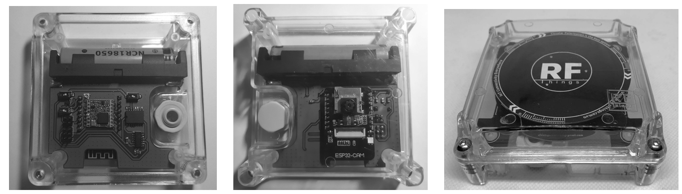
C. Monitor Website

The Things Network (TTN) provides APIs, which allows a local website to send an HTTP request to this server to get water meter values. Fig. 10 shows a monitor website created with Bootstrap, AJAX.

Fig. 10: Monitor website

D. Final Device

Fig. 11 below shows the device is protected inside an IP65 water-proof plastic case. The case is well-designed for minimal inference to the antenna.



(a) Front PCB (b) Back PCB (c) Antenna & Case

Fig. 11: Final Device

IV. Conclusion

This paper presents a design with great application potential and can be deployed in widespread use because of its high accuracy, low power, small footprint, using open-source AI libraries and low cost commercial off-the-shelf component (ESP32-Cam). However, the device is trained only for classifying Vietnam's water meter digit type, and the meter's face has to be relatively clean. The deep learning model can not be completely accurate as using sensors, reed switch, or digital water meter.

V. Acknowledgment

The author is grateful to University of Information Technology (UIT), RFThings company and L.E.A.T laboratory for their advice, supporting tools and facilities for this research.

References

[1] Mduduzi John Mudumbe, Adnan M. Abu-Mahfouz, ”Smart water meter system for user-centric consumption measurement”, in 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), doi: 10.1109/INDIN.2015.7281870.

[2] M Suresh, U. Muthukumar, Jacob Chandapillai, “A novel smart watermeter based on IoT and smartphone app for city distribution management”, in 2017 IEEE Region 10 Symposium (TENSYMP), doi: 10.1109/TENCONSpring.2017.8070088.

[3] Neeharika Cherukutota, Shraddha Jadhav, ”Architectural framework of smart water meter reading system in IoT environment”, in 2016 International Conference on Communication and Signal Processing (ICCSP), doi: 10.1109/ICCSP.2016.7754253.

[4] Vlad Paul FERNOAGA, George-Alex STELEA, Alexandra BALAN, Florin SANDU, “OCR-based Solution for The Integration of Legacy And-Or Non-Electric Counters in Cloud Smart Grids”, in 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME), doi: 10.1109/SIITME.2018.8599200.

[5] Dasol Han, HyungWon Kim, “A number recognition system with memory optimized convolutional neural network for smart metering devices”, in 2018 International Conference on Electronics, Information, and Communication (ICEIC), doi: 10.23919/ELINFOCOM.2018.8330594.

[6] Y. Liu, Yan-bin Han, Y. Zhang, “Image type water meter character recognition based on embedded DSP”, doi: 10.5121/csit.2015.51407.

[7] Ali Nikoukar, Saleem Raza, Angelina Poole, Mesut Güneş, Behnam Dezfouli, “Low-Power Wireless for the Internet of Things: Standards and Applications”, in IEEE Access (Volume: 6), doi: 10.1109/ACCESS.2018.2879189.

[8] Ritsu Tei, Hiroyuki Yamazawa, Takao Shimizu, “BLE power consumption estimation and its applications to smart manufacturing”, in 2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), doi: 10.1109/SICE.2015.7285303.

[9] Jose Maria Castillo-Secilla, Pedro Canales Aranda, Francisco José Bellido Outeiriño, Joaquin Olivares, “Experimental Procedure for the Characterization and Optimization of the Power Consumption and Reliability in ZigBee Mesh Networks”, in 2010 Third International Conference on Advances in Mesh Networks, doi: 10.1109/MESH.2010.16.

[10] Nadeem Qaisar Mehmood, Rosario Culmone, “An ANT+ Protocol Based Health Care System”, in 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, doi: 10.1109/WAINA.2015.45.

[11] Le Huy Trinh, Manh Thao Nguyen, Fabien Ferrero, “Impact of Miniaturization on a UHF tri-fillar antenna for IoT communication from satellite”, in 2020 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting, doi: 10.1109/IEEECONF35879.2020.9329452.

[12] L. H. Trinh, V. X. Bui, F. Ferrero, T. Q. K. Nguyen, M. H. Le, “Signal propagation of LoRa technology using for smart building applications” , in 2017 IEEE Conference on Antenna Measurements & Applications (CAMA), doi: 10.1109/CAMA.2017.8273458.

[13] Filip Turčinović, Josip Vuković, Slaven Božo, Gordan Šišul, “Analysis of LoRa Parameters in Real-World Communication”, in 2020 International Symposium ELMAR, doi: 10.1109/ELMAR49956.2020.9219028.

[14] Stefano Alvisi, Francesco Casellato et al., “Wireless Middleware Solutions for Smart Water Metering”, in 2019 Middleware Solutions for Wireless Internet of Things - ISBN 978-3-03921-037-4, doi: https://doi.org/10.3390/s19081853.

[15] Pangestu, P., Gunawan, D., Hansun, S., “ Histogram Equalization Implementation in the Preprocessing Phase on Optical Character Recognition”, in International Journal of Technology. Volume 8(5), pp. 947-956, doi: https://doi.org/10.14716/ijtech.v8i5.877.