# IMPLEMENTING AI ON LOW-POWER EMBEDDED DEVICES FOR DIGITAL WATER METER IDENTIFICATION AND DATA TRANSFER VIA LORA NETWORK

**Thai Vu Hien***

*The University of Danang - University of Science and Technology, Vietnam*

*Corresponding author: tvhien@dut.udn.vn

(Received: March 28, 2024; Revised: May 04, 2024; Accepted: May 06, 2024)

**Abstract -** This study introduces an artificial intelligence system implemented on the ESP32-CAM platform, aimed at conducting optical character recognition (OCR) on water meters. Leveraging LoRa technology for data transmission ensures efficient energy utilization and convenient long-range communication capabilities. The system achieves an impressive accuracy rate of 98.2% in identifying water meter readings, showcasing its reliability. Proposed as a feasible solution, it offers the advantages of low energy consumption, cost-effectiveness, and flexibility in widespread deployment, particularly leveraging existing water meter infrastructure. Thus, this research presents a promising choice for various applications beyond merely reading water meter readings. Its efficient and accurate OCR functionality makes it suitable for diverse scenarios, ranging from smart city initiatives to industrial automation processes. With its ability to integrate seamlessly into existing infrastructure and deliver reliable results, this system stands poised to revolutionize OCR applications in various domains, contributing to enhanced efficiency and productivity.

**Key words -** LoRa; Machine Learning; Water Meter; Image Processing; Low Power.

## 1. Introduction

Nowaday, where advanced Internet of Things (IoT) systems often require complex machine learning operations. Traditionally, these operations run on high performance computers or servers, which are costly and energy-intensive. Transmitting data to cloud servers for processing further increases power consumption and raises concerns about privacy and data integrity. However, advancements in open-source software and IC design have led to the development of energy-efficient and cost-effective embedded boards, robust computing capabilities. These boards can execute high-precision signal processing or complex algorithms without relying on internet connectivity, offering a new opportunity to implement AI directly on embedded devices. This approach, particularly useful for applications like smart metering and monitoring, has garnered significant interest. Several papers have explored automatic water meter reading using embedded devices.

In [1-3], methods were discussed that involve physical intervention into the water meter. However, implementing these techniques can be challenging in regions with stringent regulations and may escalate costs. In contrast, the papers suggest employing cameras as a means of modifications, replacements of meters. These investigations involve capturing images of the dial face of water meters and sending them to the cloud for analysis, achieving accurate digit classification through the use of robust cloud computing resources. Nevertheless, these systems rely heavily on a stable internet connection, which elevates power consumption and processing time. Conversely, papers [4-6] outline OCR methods that do not necessitate internet connectivity, physical modifications to the meter, or meter replacements. However, this research primarily focuses on OCR tasks without incorporating long-range data collection techniques in internet-disconnected areas. Among these, [7-8] utilize devices such as mobile phones, smart glasses, or computers equipped with high-performance that are ineffective on energy-efficient and costly. Moreover, the research referenced in [9] and [10] lacks comprehensive hardware designs. Additionally, they employ more costs compared to our proposal. Notably, the system discussed in [11] exhibits a low recognition rate when dealing. Moreover, proposals in these articles overlook low-power design considerations and only provide complete digit, particularly in scenarios involving minimal water loss. Consequently, overall power consumption remains suboptimal, potentially shortening the system's lifespan and limiting widespread deployment.

A balance must be struck in an automated water meter between power consumption, image processing, communication capabilities and machine learning. Additionally, the data link must demonstrate high stability, extensive coverage, and ample capacity, all while maintaining energy efficiency. Consideration can be given to several low-power wireless standards, such as ANT+, BLE, LoRa and Zigbee [12]. BLE emerges as a widely embraced radio sensor network technology. Zigbee provides scalability in range and supports mesh networking functionalities [13]. ANT+ customized for monitoring applications, excelling particularly in multicast schemes [14]. LoRa, developed by Semtech, stands out as a technology engineered with low-power wide area networks (LPWAN). Renowned for its low power and impressive penetration capabilities, this modulation technique offers several benefits. LoRa facilitates long-range communication, exhibits robustness against noise, boasts high capacity, relatively easier to deploy compared to NB-IoT [15–17]. Nonetheless, it's important to acknowledge that LoRa has its limitations, notably its low data-rate. Therefore, it was leveraged to optimize the size of packet accordingly. Researchers in [18] design an device that captures an image of the water meter, converts to grayscale, then transmits to the gateway.

In the article [19], the author uses a three-layer artificial neural network (ANN) to apply in recognizing digits on a driver's license. Article [20] uses OCR model to identify product serial numbers. Research [21] applied OCR in vehicle license plate recognition. Article [22] studies the application of OCR in character recognition in paper documents, serving data digitization. Research [23] is similar to [20], using OCR to identify product serial numbers, specifically foods and beverages such as alcohol. Regarding accuracy, all studies show that OCR algorithms have a high accuracy of above 95%. However, in this research, with the application of OCR in water meter digit recognition. Identification has some more difficult problems to solve. Specifically, the lighting conditions were lower, and the digits were not as fixed as in studies [19-23]. With lower light conditions, because the camera is completely hidden on the watch face, a flashlight is integrated to improve lighting when taking photos, and image processing. In addition, unlike the studies [19-23] where the OCR models all run on high-performance computers, the author implemented the OCR algorithm entirely running on the low-power embedded device ESP32-CAM. Therefore, the processing must also be optimized in terms of capacity and image size to ensure processing. In this study, image processing algorithms were proposed, along with convolutional neural network (CNN) designed specifically for ESP32-CAM. It was implemented on the ESP32-Cam platform to enable Optical Character Recognition (OCR) to read water meter value. Subsequently, the data was transmitted by LoRaWAN protocol. The ESP32-Cam served as a node within a LoRa-based low-power wide area network (LPWAN), ensuring connectivity and efficient data transmission across the network.

## 2. System design

### *2.1. System diagram*

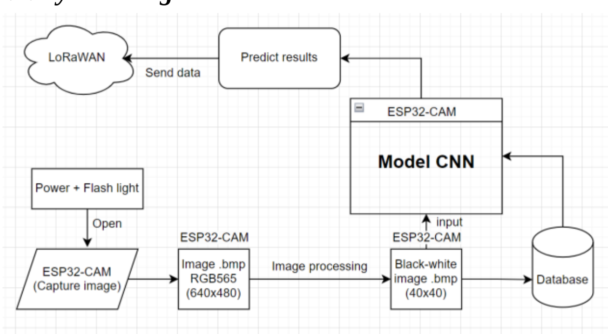


***Figure 1.*** *System design diagram*

Figure 1 shows that when the device is started with the switch, the input current will help turn on the flashlights designed on the circuit and the ESP32-CAM, these technological strides present a novel opportunity to directly implement AI algorithms on embedded devices with constrained resources in the OSI reference model's physical layer. The ESP32-CAM module is a noteworthy example of such a device. The ESP32-CAM is a versatile microcontroller module featuring the ESP32 system-on-chip (SoC) combined with a camera module, making it an ideal platform for applications requiring both wireless connectivity and image processing capabilities. The ESP32-CAM runs the program loaded on it and takes images by taking preset images in RGB565 format. After image was taken, it will be passed through an image processing function such as cropping, increasing brightness, balancing light and thresholding the image to give output image a black and white image. Then we will receive the processed image in gray image format with dimensions of 40x40 to put into the training model embedded in ESP32-CAM. After the image is processed before running and making predictions, the image will be used as input for the ESP32-CAM to run inference and receive prediction results with the input image. Then the prediction results will be sent to server via LoRaWAN wireless network.

### *2.2. LoRa communication*

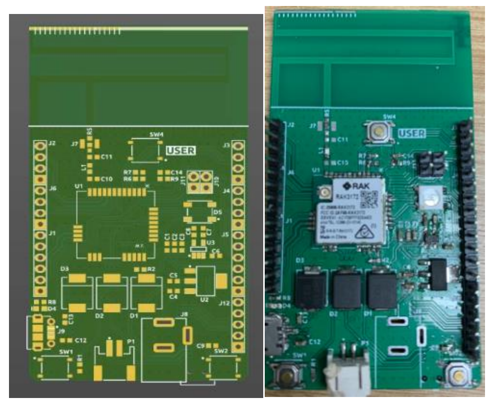

***Figure 2.*** *Design and actual image of LoRa communication device*

The decision to utilize LoRaWAN technology for this project is underpinned by its unique characteristics, which align well with the requirements of the application. LoRaWAN (Long Range Wide Area Network) is a wireless communication protocol designed for long-range communication with low power consumption, making it an attractive option for IoT (Internet of Things) applications such as the one being considered. One of the primary advantages of LoRaWAN is its exceptional range capabilities. LoRaWAN devices can communicate over distances of several kilometers in urban environments and even greater distances in rural settings, depending on factors such as antenna placement and terrain. This long-range capability is particularly beneficial for applications where devices may be deployed across wide geographic areas, such as environmental monitoring or asset tracking. The transmission delay in a LoRa network can range from a few milliseconds to several hundred milliseconds, depending on factors such as communication distance, operating frequency, and network configuration. In a standard LoRa network, the delay typically ranges from 10 milliseconds to 100 milliseconds. The packet loss rate also depends on factors such as noise, environment, and network configuration. In ideal conditions, the packet loss rate can be reduced to below 1%. However, in environments with high noise and interference, the packet loss rate can increase to several percent.

### 2.3. Image processing

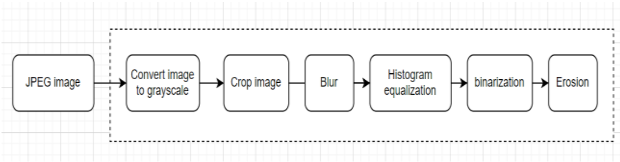


***Figure 3.*** *Illustrates the operation of preprocessing image*

Figure 3 shows the preprocessing process and data collected to train the machine learning model. For the problem of recognizing numbers, the clearest feature to distinguish them is their shape. This characteristic of the digits is retained through the process of preprocessing the image into black and white and then binary images, each digit will have an area where a different binary value "1", "0" appears, from That is fed into the machine learning model for training. The following is a flow chart of the image processing process:

First, the original captured image in the form of a 640x480 JPEG color image will be stored in the PRAM memory of the ESP32-CAM and converted to a grayscale image. Although the ESP32-CAM and OV2640 support sizes up to 1600x1200, because the JPEG image capture group uses bitmap image processing, the large image size does not have enough memory to store. For that reason, we reduced the size to match the memory of the ESP32-CAM. Very large images will first go through a cropping step to crop out the area surrounding all 5 digits.

***Figure 4.*** *Example of image processing process*

The device's source code embeds crop coordinate values to progressively segment the image into 5 distinct digit images for individual preprocessing. A crucial step in this process is Histogram Equalization. Due to the OV2640 camera's inability to focus, images often appear blurry with low contrast and overlapping gray areas, resulting in unclear digits. To mitigate this, the team employs Histogram Equalization to enhance contrast in grayscale images before converting them to binary using a team-determined threshold. This ensures effective isolation of specific number shapes in the resulting binary image. Figure 4 shows an example of image processing, the image was changed to binary image after this process.

### 2.4. Data collection

Currently, various water meter datasets are being studied, including the Water Meters Dataset by author Kutsev Roman, as illustrated in Figure 5. This dataset contains a diverse array of water meter images accompanied by corresponding OCR labels designed for reading water meter readings.

The dataset contains images with various shapes, angles, and perspectives, making it ideal for digit recognition tasks without requiring manual cropping adjustments. However, it may not align with the group's objective of capturing photos directly from water meters without unnecessary perspectives. Using the dataset for digit extraction would involve complex and time-consuming image processing.

***Figure 5.*** *Water Meters Dataset by author Kutsev Roman*

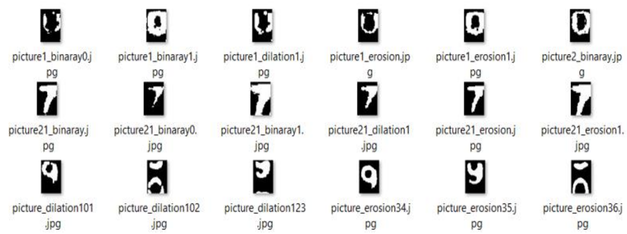

***Figure 6.*** *Dataset collected directly on ESP32-CAM*

It can be observed that images captured by the ESP32-CAM have lost some distinctive features randomly due to the low-quality camera. Even if preprocessing the images in the MR-AMR dataset to resemble those captured by the ESP32-CAM, it cannot guarantee high accuracy. Furthermore, this dataset stores images at a size of 28x28, while the group's learning model takes large input (31x43), resizing the images to increase the size will make the images blurry and no longer retain the correct digit shapes as in the original dataset. One of the objectives of this project is to create an optimized device in terms of cost. Therefore, the group proposes a solution of self-collecting training data with the goal of being able to make predictions with low-quality, low-cost hardware while still achieving high accuracy with the self-collected dataset. According to the flowchart in Figure 3, the ESP32-CAM captures images from water meter, which undergo preprocessing to produce binary digit images. A dataset is collected by simultaneously capturing images and operating a motor inside the water meter, designed by the group. This dataset trains a deep learning model in Jupiter notebooks. Once trained, the model is converted to a .cc file format for direct prediction on the ESP32-CAM without internet connection. Data collection for training involves attaching a motor mechanism inside the water meter, connected to an Arduino Uno's analog pin to generate Hash pulses controlling the motor via an L298N H-bridge circuit. After each image capture, the motor slowly rotates the dial numbers before stopping, allowing sampling of cases where only some digits rotate. Meanwhile, the ESP32-CAM performs image capture, saving images to a MicroSD card.

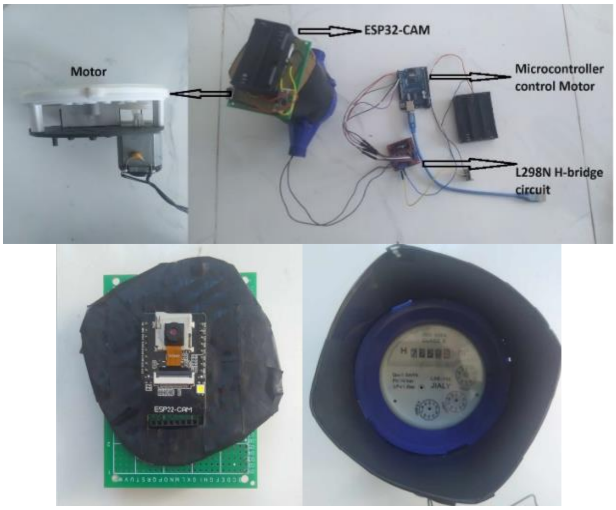


***Figure 7.*** *Data collection device*

The number of collected data samples is 2164. Below is the distribution chart of the data as shown in Figure 8.

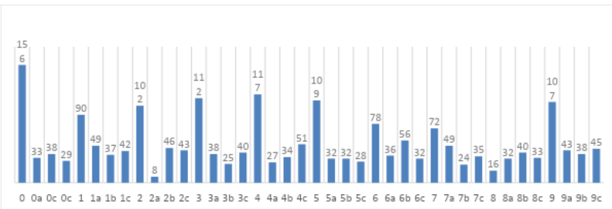


***Figure 8.*** *Distribution chart of the data*

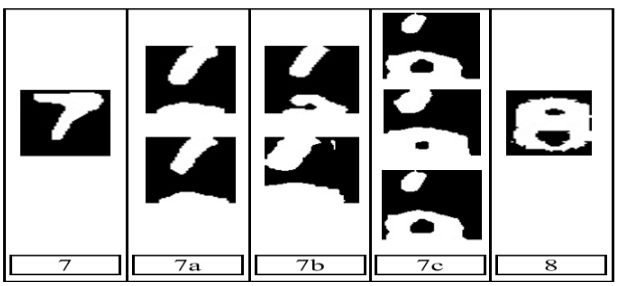


***Figure 9.*** *Illustration of some cases of numbers rotating in the middle of the data*

To improve clarity in identifying number positions on the clock face, numbers are categorized into four types, with "7" designated as the label for the clearest image. The other three labels, denoted as 7a, 7b, and 7c, represent varying distances from 7. Labels with the character 'a', for example 7a, signify cases where the number is very close to the preceding integer value (7a is close to the number 7). Labels with the character 'c', for example 7c, signify cases where the number is very close to the preceding integer value plus 1 (7c is close to 7 + 1 = the number 8). Labels with the character 'b', for example 7b, signify cases where the number is in-between (7b is between the numbers 7 and 8). The dataset is divided into training and validation sets in an 85% - 15% ratio. Initially, 95% of each class is extracted to form the training set, rounded up to ensure sufficient samples per class. However, the dataset suffers from imbalance due to easier collection of integer cases during sampling. To address this, the number of images per class in the training set is increased using image morphology transformation methods such as erosion, dilation, and random duplication. The number of images per class is adjusted to either match the largest class or exceed a certain threshold, ensuring balance. Consequently, the training dataset varies each time with 70% of the data randomly selected for this purpose.

### *2.5. Machine Learning model*

***Table 1.*** *Proposed machine learnings model*

| Class | Kernel | Size/Stride/Padding | Activation |
|---|---|---|---|
| Input: 48x48x3 | | | |
| Convolution-1 | 4 | 3x3 /1/valid | relu |
| Drop-out rate=0.1 | | | |
| Max-Pooling | | 2x2 /None/valid | |
| Convolution-2 | 8 | 3 x 3 /1/valid | relu |
| Max-Pooling | | 2x2 /None /valid | |
| Convolution-3 | 16 | 3 x 3/1/valid | relu |
| Drop-out rate=0.2 | | | |
| Max-Pooling | | 2x2 /None/valid | |
| Flatten | | | |
| Dense: Activation: relu | | | |
| Dense: Activation: softmax, units: 40 | | | |

The experiment considered models like YOLO, but its size exceeds ESP32-Cam's memory capacity. Today's deep learning models are typically large and demand high processing power, beyond what ESP32-Cam offers. Existing models often have different designs and modifying them may not ensure the same performance. Moreover, applying complex models to simple binary image data isn't efficient in terms of computational resources. The model described in Table 1 comprises two convolutional layers and one max-pooling layer. To mitigate RAM constraints, the input is resized to 48x48x3. With only three convolutional layers, the network employs two fully connected layers (dense) to augment neural connections, thereby improving classification accuracy. Dropout layers are incorporated to mitigate overfitting. The network employs the Sparse Categorical Crossentropy loss function along with the Adam optimizer. The learning rate is set to 0.001, and the batch size is 64. and we found that it achieved the desired performance, so we decided not to continue experimenting with other published deep learning models. Before the training process begins, it is necessary to set hyperparameters and configurations appropriately as they directly affect the training process.

## 3. Results

Figures 10 and 11 illustrate that the loss value on the validation set is lower than that on the training set, while conversely, the accuracy value is higher. This indicates that the model is being trained in the right direction, despite experiencing slight underfitting. Underfitting refers to the phenomenon where the model has not fully generalized to the entirety of the data. The distribution table of the model's accuracy values on the test set is statistically analyzed based on a normal distribution through five random data sampling for training. This image shows that the average accuracy value achieved is

98.2%. This implies that the accuracy could increase or decrease by 0.9%. The achieved results surpass those of other applications utilizing ESP32-Cam, such as in [19] where the accuracy was 95.7%, or in [20] where the accuracy was 88.85%. After training, the machine learning model is converted into array data in the C language to be loaded into the memory of ESP32-Cam. The model occupies 154 KB in the memory of ESP32-Cam and reduces to only 65.2 KB if quantization is applied. The result for the case when the model runs directly on the ESP32-CAM as was shown in Figure 12.

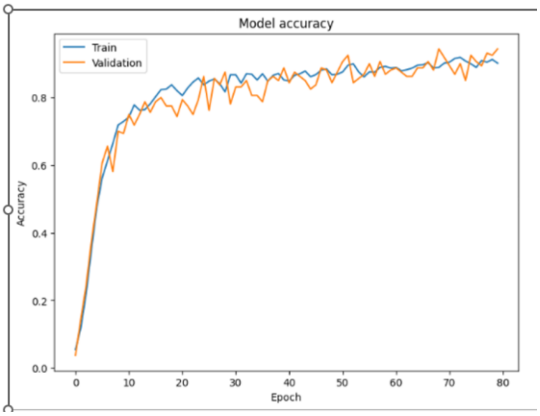


***Figure 10.*** *Model accuracy*

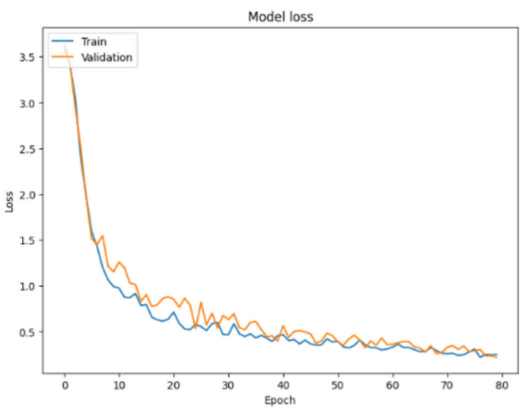


***Figure 11.*** *Model loss*

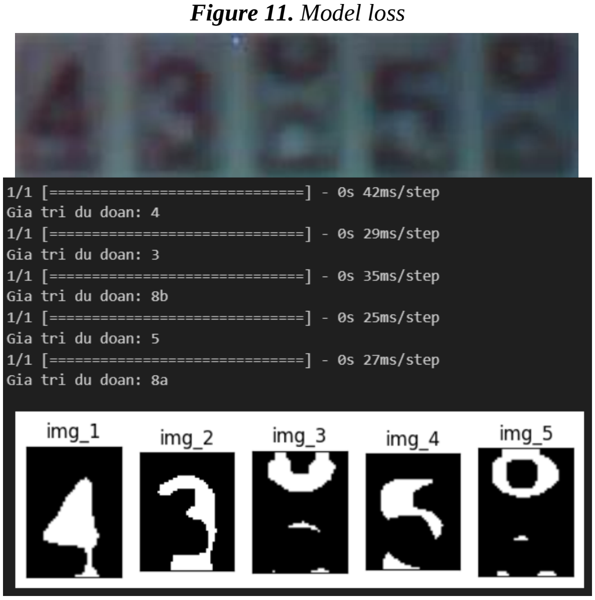


***Figure 12.*** *Example of predicting digits on a water meter*

## 4. Conclusion

This paper presents a design that holds promise for various applications and can be widely utilized due to its precision, minimal power consumption, compact size, utilization of open-source AI libraries, and the use of cost-effective commercially available components like the ESP32-Cam. The system demonstrates remarkable accuracy, reaching as high as 98.2% in classifying segmented digits, this is the accuracy of the model run on ESP32-CAM, the machine learning model is converted into array data in the C language to be loaded into the memory of ESP32-Cam, highlighting its effectiveness. These characteristics combined make the suggested system a viable option known for its minimal energy usage, cost-effectiveness, long-lasting performance, and suitability for widespread implementation. Consequently, it becomes an attractive option for various applications where efficient and accurate optical character recognition (OCR) and a prolonged operational life are crucial factors to consider.

**Acknowledgement:** This research is funded by The Murata Science Foundation and The University of Danang - University of Science and Technology, code number of Project: T2023-02- 10MSF.

## REFERENCES

[1] M. Suresh, U. Muthukumar, and J. Chandapillai, “A novel smart watermeter based on IoT and smartphone app for city distribution management”, in *IEEE Region 10 Symposium (TENSYMP)*, 2017, pp. 1-5.

[2] M. Mudumbe and A. M. Abu-Mahfouz, “Smart water meter system for user-centric consumption measurement”, *in IEEE 13th international conference on industrial informatics (INDIN), 2015*, pp. 993-998.

[3] N. Cherukutota and S. Jadhav, “Architectural framework of smart water meter reading system in IoT environment”, *in International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 0791-0794.

[4] F. Yang, L. Jin, S. Lai, X. Gao, and Z. Li, “Fully Convolutional Sequence Recognition Network for Water Meter Number Reading”, *IEEE Access*, vol. 7, pp. 11679 – 11687, 2019.

[5] K. Eurviriyanukul, K. Phiewluang, S. Yawichai, and S. Chaichana, “Evaluation of Recognition of Water-meter Digits with Application Programs, APIs, and Machine Learning Algorithms”, *in 8th International Electrical Engineering Congress (iEECON), 2020*, pp. 1-4.

[6] C. Li, Y. Su, R. Yuan, D. Chu, and J. Zhu, “LightWeight Spliced Convolution Network-Based Automatic Water Meter Reading in Smart City”, *IEEE Access*, vol. 7, pp. 174359 – 174367, 2019.

[7] V. P. Fernoaga, G. Stelea, A. Balan, and F. Sandu, “OCR-based Solution for The Integration of Legacy And-Or Non-Electric Counters in Cloud Smart Grids”, *in IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2018, pp. 398-403.

[8] A. Sharma and K. K. Kim, “Lightweight CNN based Meter Digit Recognition”, *Journal of Sensor Science and Technology*, vol. 30, pp. 15 - 19, 2021.

[9] Y. Liu, Y. Han and Y. Zhang, “Image type water meter character recognition based on embedded DSP”, *Computer Science and Information Technology*, vol. 5, 2015.

[10] G. Jin, K. Bai, and H. He, “A Smart Water Metering System Based on Image Recognition and Narrowband Internet of Things”, *Rev. d’Intelligence Artif*, vol. 33, pp. 293-298, 2019.

[11] A. Nikoukar, S. Raza, A. Poole, M. Gunes, and B. Dezfouli, “Low-

Power Wireless for the Internet of Things: Standards and Applications", *IEEE Access*, vol. 6, pp. 67893 - 67926, 2018.

[12] R. Tei, H. Yamazawa, and T. Shimizu, "BLE power consumption estimation and its applications to smart manufacturing", *in 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2015, pp. 148-153.

[13] J. M. Castillo-Secilla, P. C. Aranda, F. J. Outeirino, and J. Olivares, "Experimental Procedure for the Char-˜ acterization and Optimization of the Power Consumption and Reliability in ZigBee Mesh Networks", *in Third International Conference on Advances in Mesh Networks*, 2016, pp. 13-16.

[14] N. Q. Mehmood and R. Culmone, "An ANT+ Protocol Based Health Care System", *in IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, 2015, pp. 193-198.

[15] L. H. Trinh, M. T. Nguyen, and F. Ferrero, "Impact of Miniaturization on a UHF tri-fillar antenna for IoT communication from satellite", *in IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*, 2020, pp. 403-404.

[16] L. H. Trinh, V. X. Bui, F. Ferrero, T. Q. K. Nguyen, and M. H. Le, "Signal propagation of LoRa technology using for smart building applications", *in IEEE Conference on Antenna Measurements & Applications (CAMA)*, 2017, pp. 381-384.

[17] Turčinović *et al.*, "Analysis of LoRa parameters in real-world communication.", *in International Symposium ELMAR*, 2020, pp. 87-90.

[18] S. Alvisi *et al.*, "Wireless Middleware Solutions for Smart Water Metering", *Sensors*, vol. 19, no. 8, pp. 1853, 2019.

[19] P. Shah, S. Karamchandani, T. Nadkar, N. Gulechha, K. Koli, and K. Lad, "OCR-based chassis-number recognition using artificial neural networks", *2009 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Pune, India, 2009, pp. 31-34.

[20] M. M. Hsu, M. -H. Wu, Y. -C. Cheng, and C. -Y. Lin, "An Efficient Industrial Product Serial Number Recognition Framework*", 2022 IEEE International Conference on Consumer Electronics - Taiwan, Taipei*, 2022, pp. 263-264.

[21] S. Ranjan *et al.*, "OCR based Automated Number Plate Text Detection and Extraction", *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2022, pp. 621-627.

[22] R. Avyodri, S. Lukas, and H. Tjahyadi, "Optical Character Recognition (OCR) for Text Recognition and its Post-Processing Method: A Literature Review", *2022 1st International Conference on Technology Innovation and Its Applications (ICTIIA)*, 2022, pp. 1-6.

[23] S. Čakić, T. Popović, S. Šandi, S. Krčo, and A. Gazivoda, "The Use of Tesseract OCR Number Recognition for Food Tracking and Tracing", *2020 24th International Conference on Information Technology (IT)*, Zabljak, Montenegro, 2020, pp. 1-4.

[24] H. Bangkit *et al,*. "Automatic Water Meter Reading Development Based On CNN and LoRaWAN.", *in International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, 2023, pp. 212-215.

[25] Slyusar *et al.*, "Segmentation of analogue meter readings using neural networks", *in 4th International Workshop on Modern Machine Learning Technologies and Data Science MOMLET&DS*, 2022, pp. 165-175.