

A fully AI-based system to automate water meter data collection in Morocco country

Ayman Naim^{*}, Abdessadek Aaroud, Khalid Akodadi, Chouaib El Hachimi

Lab. LAROSERI, Dept. of Computer Science, Faculty of Science, Chouaib Doukkali University, El Jadida, Morocco

ARTICLE INFO

Keywords:
 Automatic meter reading
 Telemetry
 MNIST
 CNN model
 Embedded system
 Water consumption

ABSTRACT

With the growing demand for water resources, the need for monitoring has become a necessity for rational and sustainable use of this resource. Water meter data collection is an essential step toward this goal. In Morocco, this task is performed manually at most once a month due to constraints related to the cost and time. In general, the consumption is estimated and calculated based on the average consumption recorded in the previous months. This causes many claims from customers because of higher invoices, which does not reflect reality. In this paper, we propose a fully AI-based system to automate water meter data collection, which is composed of a Recognition System (RS) and a web services platform. This framework offers multiple services for both customers and water service providers, such as consumption monitoring, detecting water leaks, visualizing water consumption, and potable water coverage in a geographic map. It also provides a powerful tool to help ensure accurate decision making with multiple reporting services. The main component of the RS is the Convolutional Neural Network model trained on a proposed MR-AMR (Moroccan Automatic Meter Reading) dataset. In the model test phase, we achieved an accuracy of 98.70%. Our system was tested and validated by experiments.

1. Introduction

Humans cannot change the amount of water available on earth. However, the growing demand for industrial, agricultural, and domestic use of this resource has transformed its regime and disrupted its natural functioning, which threatens this vital resource, especially the potable water that needs to go through a series of expensive treatments to rid it of all impurities before human consumption. So, the rationalization of the use of this resource should be carefully considered.

The growth of many technical fields such as the internet of things, artificial intelligence, and Big Data has facilitated several complex challenges in recent years, such as remote water meter data collection and the use of smart meters to track water use.

In Morocco, water meter data collection is done manually at most once a month by the National Office of Electricity and Drinking Water (ONEE). This water service provider often issues invoices based on estimates.

The estimated consumption is calculated based on the average consumption recorded in the previous months, which accumulates reading errors. The office corrects these errors during the last month of the year.

This process makes invoices subject to severe price problems due to

water consumption ranges, as shown in Table 1. The price of invoices increases, depending on which consumption range the customer belongs to.

According to some customer testimonies, water invoices have reached up to \$150 for customers who usually pay around \$20 each month.

In summary, the manual process can cause multiple issues for both customers and water meter service providers because of the high cost of these corrective operations and the lack of accurate consumption data.

Most of the works cited in the literature treat a sub-part of this problem using different techniques [2,5,6,8]. In this paper, we propose a fully AI-based recognition system constituted with hardware and software parts. The hardware part is made up of.

- Raspberry Pi 3b+,
- Camera,
- 180-degree angle lens for index zooming out,
- LED for dark vision,
- Battery for a Raspberry Pi (RPI).

The software part consists of.

* Corresponding author.

E-mail address: aymannaim10@gmail.com (A. Naim).

Table 1
Water consumption ranges.

Billing method	Monthly consumption in m ³	Ranges	Price in \$ Excluding Taxes
Progressive billing	1 ≤ Consumption ≤ 12	Range 1	from 1 to 6 m ³ 0.26
		Range 2	from 7 to 12 m ³ 0.519
Selective billing	13 ≤ Consumption ≤ 20	Range 3	from 1 to 20 m ³ 0.519
		Range 4	from 1 to 35 m ³ 0.989
	> 35 Consumption	Range 5	from 1 to total consumption 1.458

- CNN model for index recognition,
- Multi-services web platform.

As shown in Fig. 1, the system does not require renewals or replacement of existing meters and is easy to install.

The leftover portion of this paper is organized as takes after. Section 2

depicts the related works of all meter index reading technics. We introduce the proposed approach and the AI-based recognition system in sections 3 and 4. Sections 5 and 6 provide the MR-AMR dataset and the web services platform. Implementation & experimentation is presented in section 7. We conclude and present some perspectives in section 8. An annex is added at the end of this paper.

2. Related work

There are several works done in state of the art for the AMR (Automatic Meter Reading) problems. Some authors incorporate three main stages: (1) counter detection, (2) digit segmentation, and (3) digit recognition, and others focused on a single stage.

Authors in Refs. [2,7,12] took some features of the water meter images such as the counter digits area position with their colors (green background and red decimal digits) and applied a notable index detection algorithm. A major inconvenience of this technique is that it might not work on all kinds of water meters, and the WM (Water Meter) illumination changes (day or night) influence the quality of output results badly.

Authors in Refs. [6,11], have relied their works on Deep Learning approaches, which are mainly dependent on the availability of unique and large dataset. It is used to train and evaluate their systems (more than

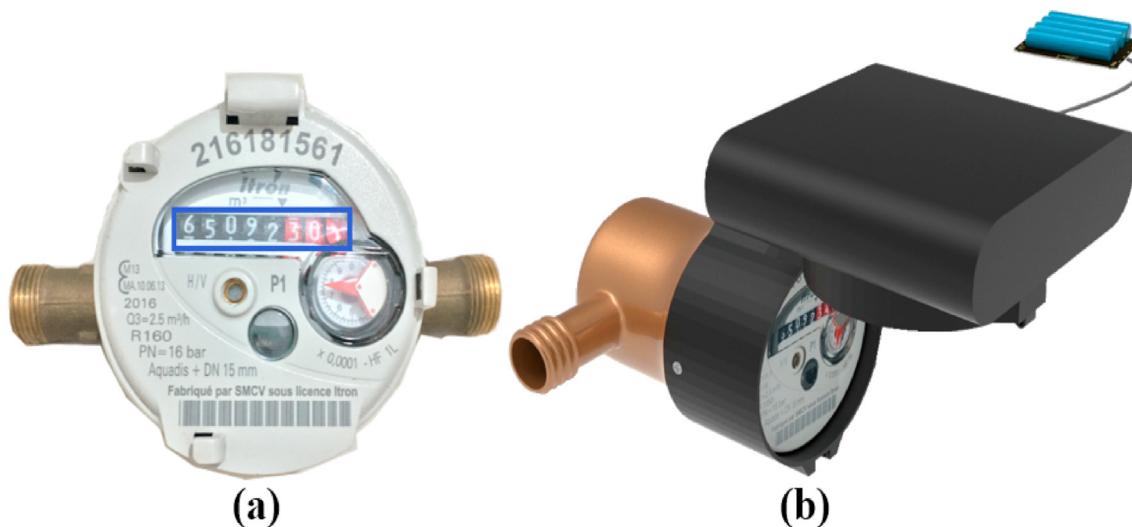


Fig. 1. (a) Water meter without embedded system (b) Water meter with an embedded system.

Table 2
Benchmarking study of most recent related work.

	[5; 7;12]	[2; 6;20]	[11; 19; 21]	[1; 26; 28; 30]	[27; 29]	Our approach
Camera's type	not specified	Mobile phone camera	mobile phone camera or not specified	not specified or embedded camera	Mobile phone camera	embedded camera
Index detection	whole meter image	whole meter image	whole meter image	area of interest or whole meter image	whole meter image	area of interest
Input digit's type	full-state image	full-state image	both full-state and mid-state images	both full-state and mid-state	dial meter	both full-state and mid-state images
Dataset size	[253; 8000; 903] images	[168958; 6150; 10] images	[222198; 2000; 23810] images	[6000; 6000; 2000; 777]	[2000; 1430]	140000 images
Embedded system (yes/no)	no	no	no	[no; no; yes; no]	no	yes
The trigger event for index acquisition	not specified	manually	not specified	not specified	no	online
Web services framework	no	[no; yes; yes]	[yes; no; no]	[no; yes; yes; not specified]	no	yes
Power energy	not specified	not specified	not specified	not specified	no	specified
Recognition in a dark vision	no	no	no	no	no	yes
Test accuracy (%)	[87; 79.52; 95]	[67.5; 81.42; 99]	[94.16; 98.30; 99.59]	[97; not specified; 97; 98.5]	[93.6; 97.65]	98.70

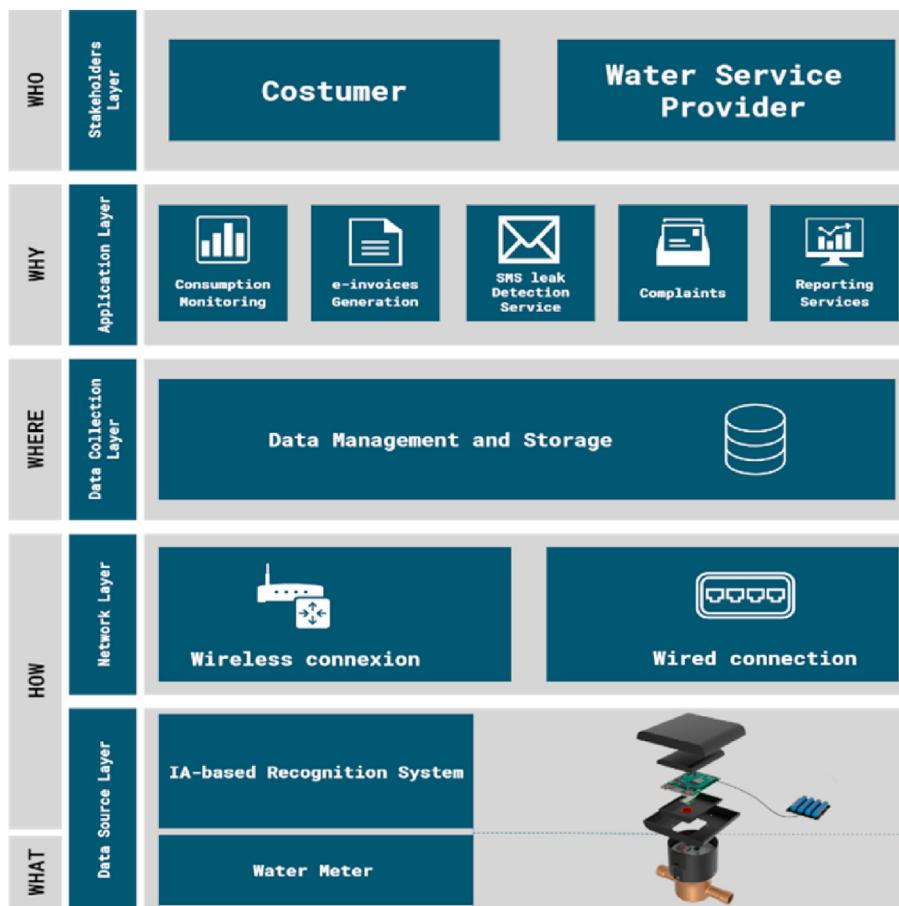


Fig. 2. General multi-layer architecture.

Table 3
Comparative study between centralized and decentralized approaches.

Factors	Centralized approach	Decentralized approach
Computational power	Higher	Lower
Storage	A massive amount of data	Fewer data storage needed
Maintenance Cost	Expensive	Inexpensive
Network debit needed	High	Low

45 000 images). However, these datasets were not adapted to all WM and not freely accessible to the public.

Elrefaei and Bajaber [20], made three primary stages of recognizing the electric meters: pre-processing which ends up with cropping the numeric reading area, segmentation of individual digits using horizontal and vertical scanning of the cropped numeric area, and recognition of the reading by comparing each segmented digit with the digit's templates. Their proposed system is implemented using Android Studio software with the OpenCV library. Although better results were attained, only 21

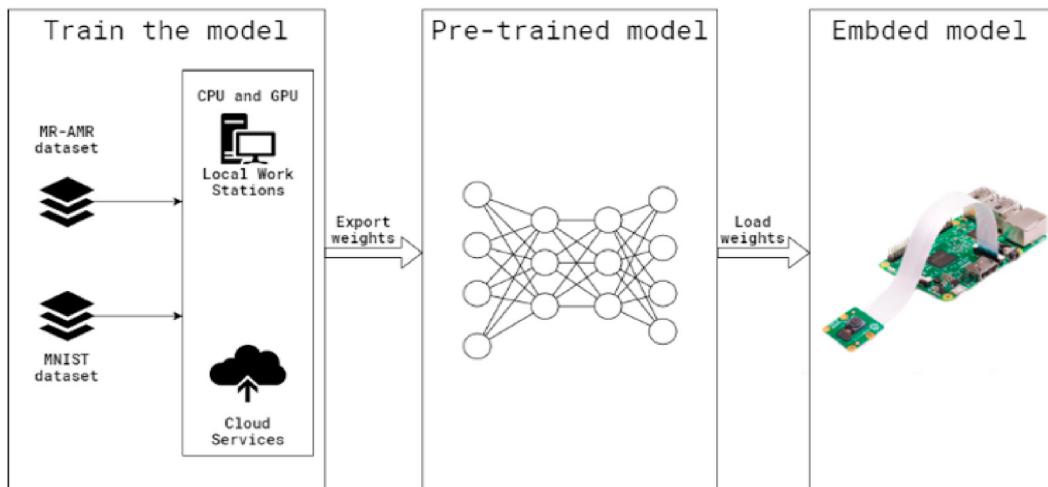


Fig. 3. Embedded model workflow.

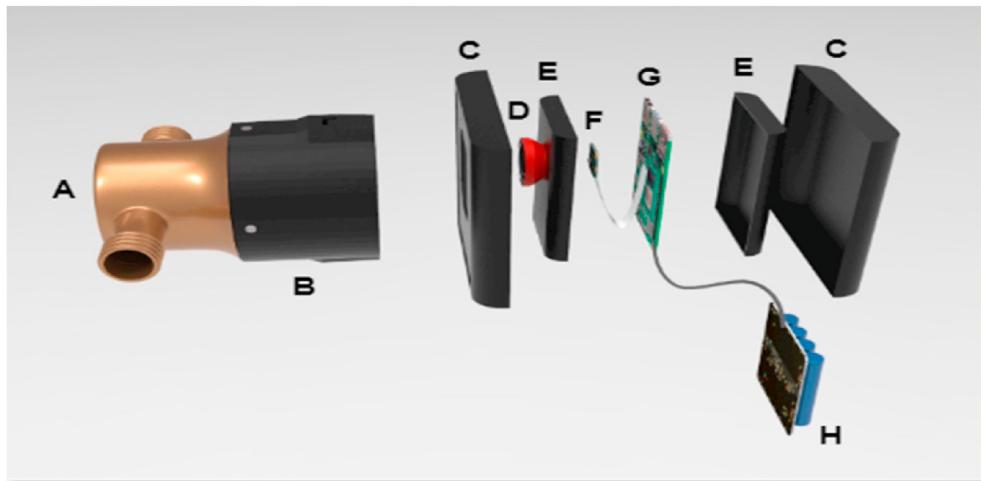


Fig. 4. AI-based Recognition System prototype.



Fig. 5. Camera connected to the RPI.

Table 4
System autonomy in the worst case.

RPI state	I_{T_d} (mA)	T_d (h)
Power consumption in stress mode		
400% CPU load (stress -cpu 4)	980	20
RS = Stress + WIFI + camera +4 led	$980+50+250+4*30=1400$	14
Power consumption optimization		
RS without USB/LAN IC	$1400-100=1300$	15h4min
RS without (USB/LAN IC + HDMI)	$1400-(100+30)=1270$	15h26min
RS without (USB/LAN IC + HDMI+ 4 LED)	$1400-(100+30+ 4*30)=1150$	17h03min
RS without (USB/LAN IC + HDMI+ 4 LED + Software)	$1400-(100+30+ 4*30+100)=1050$	18h40min

images were used to evaluate their system performance, which may not be enough representative. A Smartphone camera captured the dataset's images. Their system was not integrating into a full system that could be used by the electricity company employees.

Just references [5,12,19] made available the datasets used in their tests. Datasets in Refs. [5,12] are composed mostly of gas meter images with a resolution of 640×480 pixels, and the counter occupied a massive portion of the image, which eases its detection. Besides, both datasets are small (253 and 903 images, respectively), and the cameras used to capture them were not specified. In Ref. [19], the dataset is composed of 2000 electric meter images. It is the largest public one in the literature, but it is on command. It does not consist of many different meter image

Table 5
System autonomy in the best case.

RPI state	I_{T_d} (mA)	T_d (h)
Power consumption in idle mode		
Idle	350	56
RS=Idle + WIFI + camera +4 led	$350+50+250+4*30=770$	25h27
Power consumption optimization		
RS without USB/LAN IC	$770-100=670$	29h15min
RS without (USB/LAN IC + HDMI)	$770-(100+30)=640$	30h38min
RS without (USB/LAN IC + HDMI+ 4 LED)	$770-(100+30+ 4*30)=520$	37h42min
RS without (USB/LAN IC + HDMI+ 4 LED + Software)	$770-(100+30+ 4*30+100)=420$	46h40

types. This system cannot recognize digits in night vision and make some errors in half-digit image recognition. It was the first work to make the images used in the experiments publicly available. However, it was not done in a AMR pipeline.

There are many works focused on a single stage of the AMR pipeline [1,30], which makes it difficult to accurately evaluate the presented methods in an end-to-end manner (e.g., the results achieved by a recognition model may vary considerably depending on how accurate the counter region is detected). Another factor that makes it difficult to assess existing methods or their applicability, is that the authors commonly do not report the execution time of the proposed approaches or the hardware in which they performed their experiments.

Authors in Ref. [28] insert a new stage in the AMR pipeline, called corner detection and counter classification, which enables the counter region to be rectified before the recognition stage. They also introduced a publicly available dataset (upon request) for AMR with 12500 fully-annotated images acquired on real-world scenarios. Moreover, they don't have a methodology for simultaneous detection and recognize the cases where the counter has rotating digits. Real-world application is having issues due to some failed meter reads, and very few reading errors are tolerated in real-world applications since a single-digit recognized incorrectly can result in a large reading/billing error.

Authors in Ref. [26] focused on the problem of water meter recognition in smart city applications. The experimental results demonstrate that the proposed network can identify the water meter number accurately and simultaneously requires fewer parameters and less computation. They also have implemented the prototype system which can handle real-time databases on the distributed cloud platform. Their system sends the water meter index image and not the index via the transport layer to

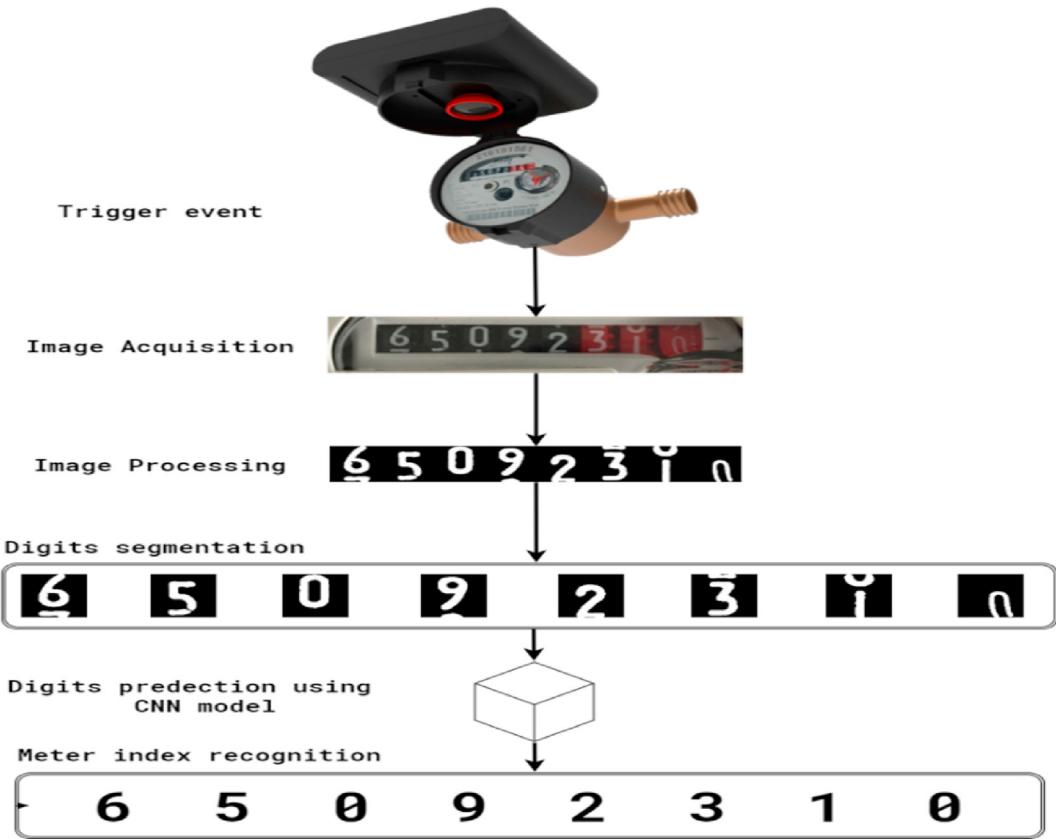


Fig. 6. Water Meter index recognition workflow.

Table 6
System autonomy in the best case.

Method	CNN	FFNN	RNN
Accuracy (%)	98.70	96.01	97.71

the function layer which increases the percentage of data storage. Furthermore, they are not transplanting the algorithm to the smart camera.

We focus on studies related to digit-based meters, even though some recent works addressed the recognition of dial meters [27,29]. Such works usually explore the angle between the pointer and the dial to perform the reading.

In addition to the points mentioned earlier, the significant researches did not make a consistent end to end full system (image recognition, dataset contribution, mobile and web service application for water and electricity companies).

In Table 2, we compare the most cited approach in the literature with our system taking into account various criteria such as the command system, dataset images, and power energy.

In the following, a full-state image refers to an image containing a digit that appears in total. In contrast, a mid-state image refers to images containing two digits that appear partially.

3. Proposed approach

We represent the proposed system with a layered model, as shown in Fig. 2. This system is intended for both Customers and Water Service Providers.

Starting from the bottom to the top, the Data Source Layer has two parts: The Water Meter and the AI-based Recognition System that retrieves the water meter index. It is achieved by taking an image,

processing it, and recognizing digits. It then sends the results and information about the customer for authentication purposes via the Network Layer to the Data Collection Layer.

The Application Layer contains the different services provided by the platform.

4. AI-based recognition system

Choosing to embed the deep learning model on a RPI system-on-chip (SoC), comes after evaluating the centralized and decentralized approaches.

4.1. Distributed vs. centralized approach

In a centralized approach, each customer's embedded system will retrieve images, send them directly to the server, and make inferences in a centralized manner.

For example, suppose we have N water meters with RPI embedded system connected to a server in a centralized approach. We assume that the size of a cropped image is 100 KB (image size in our dataset case); we should have in 24-h to store $24 \times N \times 100 \text{ KB} = 2,4 \times N \text{ MB}$. This amount of data will have a direct impact on server storage capacity, computation power, response time, and network bandwidth.

Table 3 shows the advantages of the decentralized choice that we are looking for in our case, taking into account the previous example. That is why we chose to go for distributed Recognition Systems.

The training phase was done on platforms with high computational performance (workstations, Google Cloud, and Azure Cloud), we then exported the pre-trained model architecture and weights for deployment in the Recognition System Fig. 3.

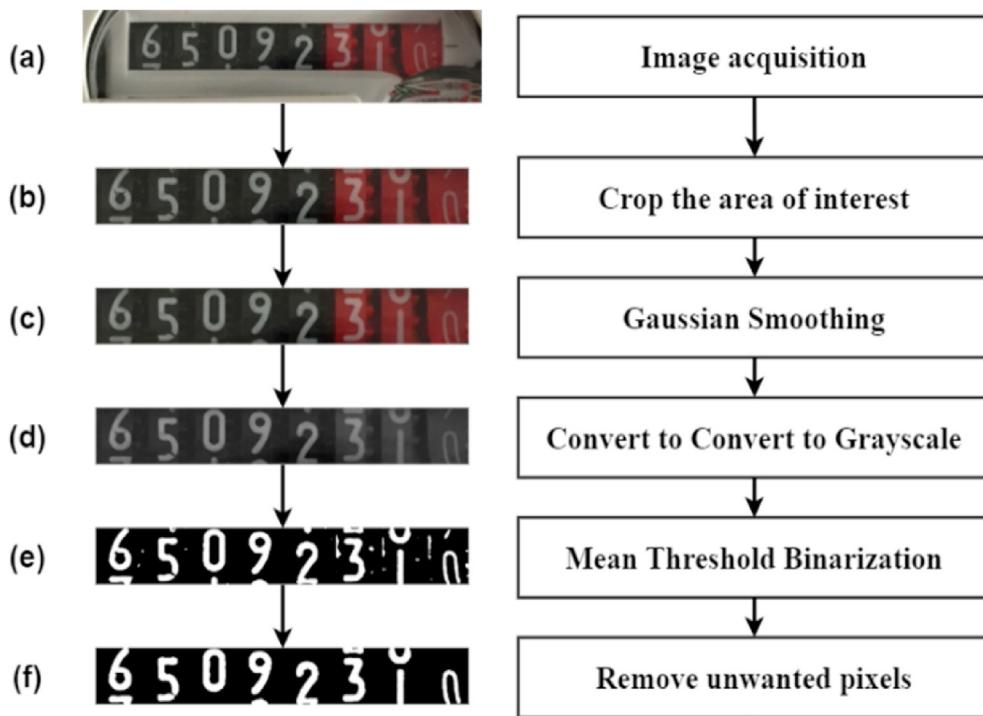


Fig. 7. Image processing steps.



Fig. 8. Digit segmentation.



Fig. 9. Digits resizing.

4.2. Components and 3D printed prototype

The Recognition System can be easily installed in the water meter (Fig. 4. A), and it is composed of seven components:

- B- Female cylinder to allow flexibility.
- C- External Security Box.
- D- 180-degree angle lens to enlarge the angle and zoom out.
- E- Internal box and four LEDs to enable night-vision.
- F- RPI Camera Module v1 (Fig. 5).
- G- Raspberry Pi (RPI) 3 b + model with 64 Go SD card.
- H- Li-ion Battery HAT, power bank management chip.

This prototype can be adapted to all Moroccan water meter types regardless of their diameter because of the nature of the elastic materials printed through it.

The advantages of using the RPI in the embedded system are:

- Extensibility
- Low cost,
- Dimension: credit-card sized,
- Useful for prototyping,
- Integrated WIFI card,
- Support all operating system OS (Windows, Linux)

- Camera access (Fig. 5)

In our case, we use Ubuntu-mate 18.04 as OS [23].

Camera, RPI, WIFI card, and LED needed electrical energy in their functions, so an energetic study is required.

4.3. Energetic study

Energy is one of the most significant challenges faced by embedded systems. Designing an energy-efficient embedded system depends on several parameters, such as its consumption and battery quality [22,23].

To deal with this challenge, we applied several techniques to optimize the consumption of our Recognition System:

- Switch the RS to an economical mode when not in use.
- Automatic power-off unused modules.

Our power component capacity is 9800 mAh for each battery. It means that the RS has an autonomy of 14 h of supply in the worst case (using the processor in stress mode, Wi-Fi, Camera, LED) and 25h27min in the best case with the same hardware configuration and without power consumption optimization.

Tables 4 and 5 show the autonomy of our Recognition System in the worst and best case, respectively, with different runtime configurations. The results can be calculated using the following formula:

$$C_{T_d} = I_{T_d} \times T_d$$

T_d : Discharge time of the battery (Autonomy in an hour).

C_{T_d} : Battery capacity associated with the T_d autonomy in ampere-hour.

I_{T_d} : Current battery discharge associated with the T_d autonomy in ampere.

The autonomy of the energy system is still limited; recharging and replacing batteries can be an annoying task for customers. Hence, we plan to design a water flow turbine generator to provide electric energy for the whole system.

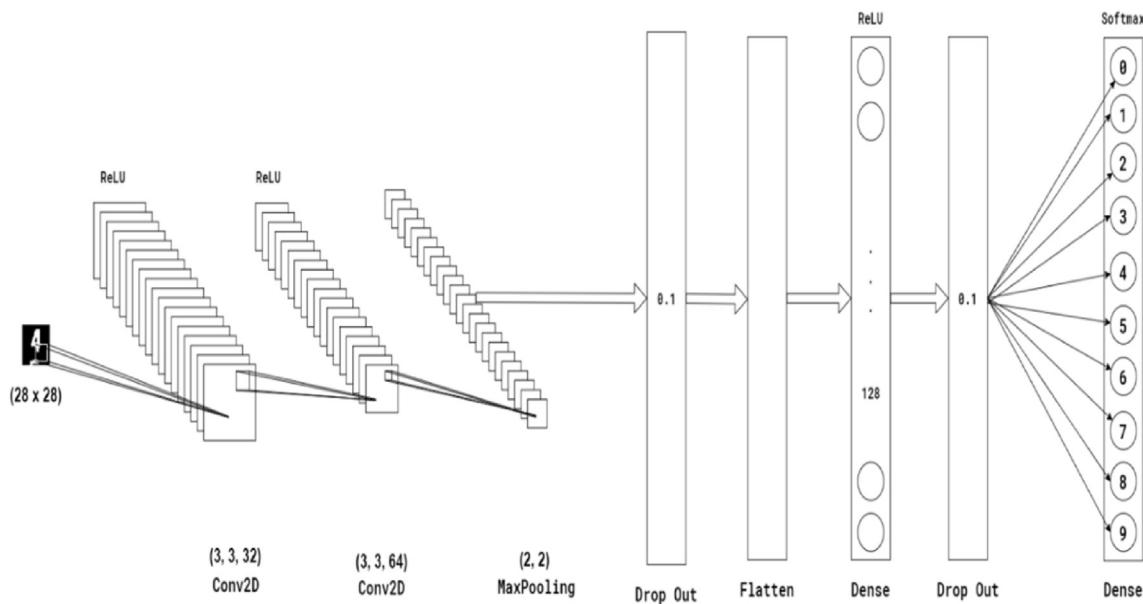


Fig. 10. CNN model for digit recognition.

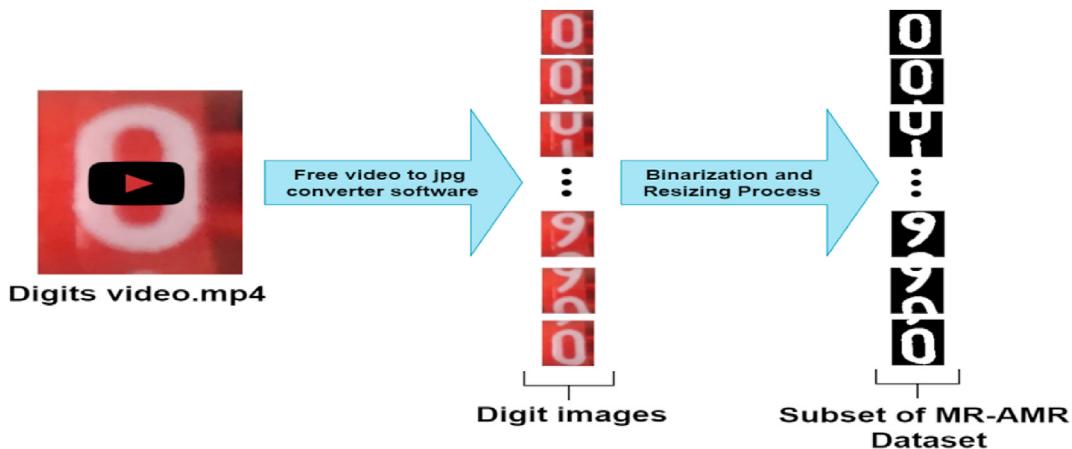


Fig. 11. Process of extracting images from video.

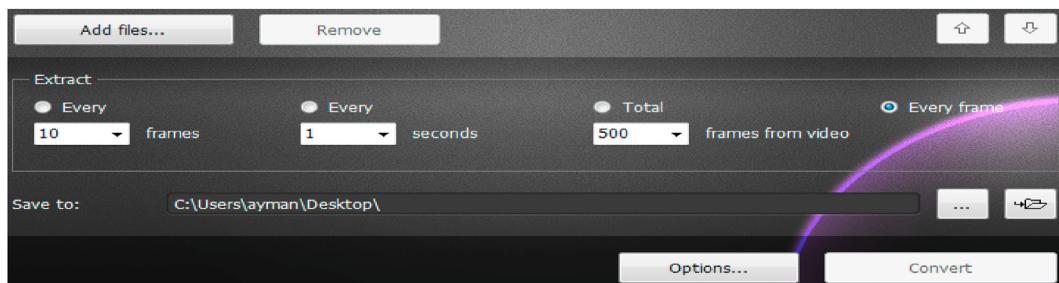


Fig. 12. Video to jpg converter interface.

4.4. Recognition system workflow

The recognition process is divided into six main steps Fig. 6.

4.4.1. Comparison of CNN with other models

To choose the best model to use during the deployment phase, we evaluated the performance of most three models used in the state of the

art with our MR-AMR dataset, as shown in Table 6.

According to the results, the CNN model is the most efficient with accuracy during the training phase of 98.70%.

4.4.2. Trigger event

The recognition process starts in response to two types of trigger events:

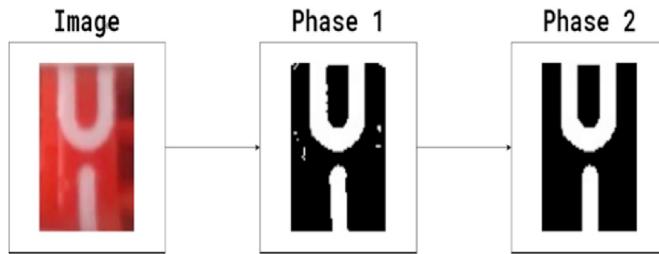


Fig. 13. Dataset preprocessing.

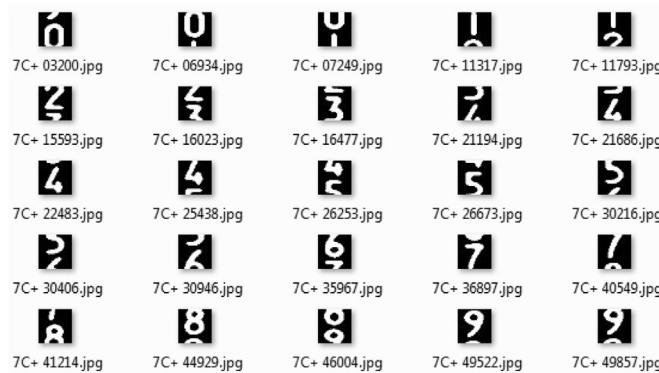


Fig. 14. Extract images from MR-AMR Dataset.

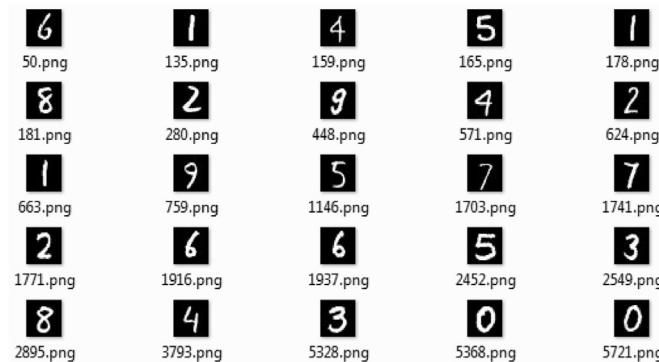


Fig. 15. Extract images from the MNIST Dataset.

Table 8

Example of image annotation.

Image	0	u	1	1	2	5	3	u	4	7
Label	0	0	1	1	2	2	3	3	4	4

- Periodically: the period is configured in the first installation of the RS onsite.
- Directly: the customer clicks on the GET MY CONSUMPTION button in his account via the platform.

4.4.3. Image processing

The first preprocessing step is reading an image received by the

Table 9
Datasets splitting.

	Total images in Training	Total images in testing	Total
MNIST	60 000	10 000	70 000
MR-AMR	120 000	20 000	140 000
MR-AMR & MNIST	180 000	30 000	210 000

We take a common hyper-parameter for all three cases.

- Epoch: When the network ends up going over the entire training dataset (i.e., once for each training instance), it completes one epoch. We train our CNN model for 10 epochs.
- Batch size: feeding the whole dataset into the CNN model represents several memory constraints. So, what is commonly done is splitting up data into subsets. In all cases, we took a batch size of 32 images.
- Learning rate: The Learning rate is the step size at each iteration while moving toward a minimum of the loss function. It decides how fast or slow we will move towards the optimal neural network weights. In all cases, we took the learning rate = 0.01.

Table 10
MNIST result.

Five-Fold cross-validation test sets	Accuracy (%)	Loss	Training time (s)
1	97.38	0.0830	4720
2	97.37	0.0850	4734
3	97.35	0.0841	4728
4	97.28	0.0842	4611
5	97.21	0.0822	4603
Average	97.31	0.0837	4679

Table 11
MR-AMR result.

Five-Fold cross-validation test sets	Accuracy (%)	Loss	Training time (s)
1	98.70	0.0375	9777
2	98.83	0.0351	9658
3	98.68	0.0399	9652
4	98.85	0.0341	9662
5	98.45	0.0459	9811
Average	98.70	0.0385	9712

Table 12
MR-AMR & MNIST result.

Five-Fold cross-validation test sets	Accuracy (%)	Loss	Training time (s)
1	98.10	0.0611	14477
2	98.01	0.0630	13833
3	98.05	0.0626	13989
4	98.14	0.0603	13703
5	98.07	0.0628	14228
Average	98.07	0.0620	14046

trigger event Fig. 7. A; the RPI is placed to take just the area of interest that contains the digits.

The second step is cropping the area of interest, the crop is based on parameterizing four coordinates $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$, which represents the corners of a new rectangle. Fig. 7 b. We made this process to ease the image processing in the next steps.

The third step is applying a Gaussian Smoothing to filter the cropped image and reduce the noise—Fig. 7 c.

The fourth step is converting the filtered RGB image to a Grayscale image—Fig. 7 d.

The fifth step is binarizing the Grayscale image; we use the Adaptive Threshold Mean Binarization function. Fig. 7 e.

The sixth step is removing unwanted pixels; we apply a suitable algorithm using the NumPy library to scan the image, then detect and remove all small pixels which do not represent a digit. Fig. 7 f.

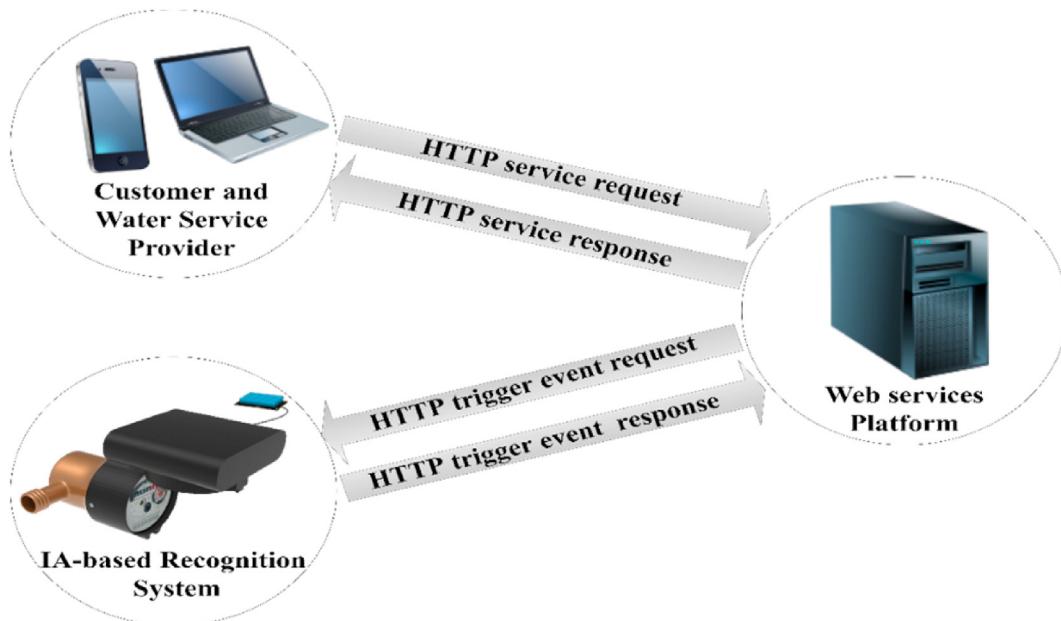


Fig. 16. Monitoring system.

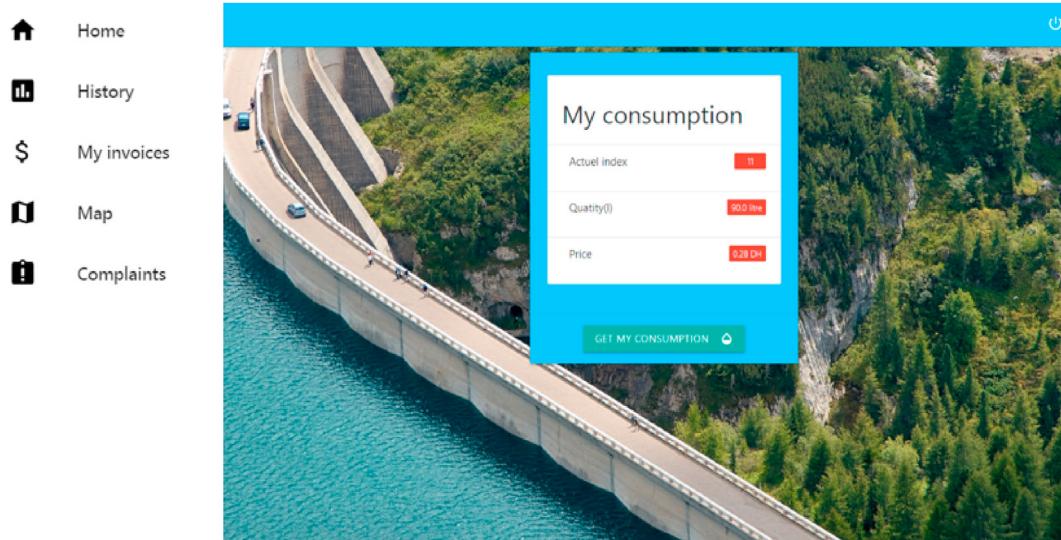


Fig. 17. Customer consumption interface.

4.4.4. Segment digits

The proposed idea behind digit segmentation is to scan all the binarized images pixel by pixel and automatically crop it into eight equal sub-images. Each sub-image contains one digit from the original image successively, as shown in Fig. 8.

After segmenting the eight digits, we resize every sub-image to 28*28 pixels, as shown in Fig. 9. The size of 28*28 is selected based on dataset conception that will be presented in section 4.

Each digit is considered as the input of a CNN model, which will be detailed in the digit recognition section.

4.4.5. Digits recognition

In this step, each box digit is passed as input for our CNN model to make predictions. The architecture of our model is shown in Fig. 10.

- The first convolutional layer, "Conv2D," contains 32 filters of (3,3). This filter slide over the input image returning 32 matrices of

activation values called feature maps that indicate where certain features are located in the image. We used Rectified Linear Unit (ReLU) as an activation function.

- The second convolutional layer, "Conv2D," contains 64 (3,3) filters with ReLu as the activation function.
- The "Max Pooling" layers reduce the number of parameters by reducing the dimensionality of each feature map, it takes the most substantial element from each one of them, and retains essential information.
- As a Regularization method, we use a Dropout layer with a ratio of 0.1 to be sure that our model does not depend on input data (overfitting).
- We flattened then our 3D matrix composed of 64 feature maps into one vector, and feed it into a fully connected layer "Dense"; which contains 128 Neuron with ReLu as the activation function.
- Another Dropout is applied with a ratio of 0.1
- Finally, the output layer is fully connected; it contains ten neurons, each neuron represents our classes (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). It uses

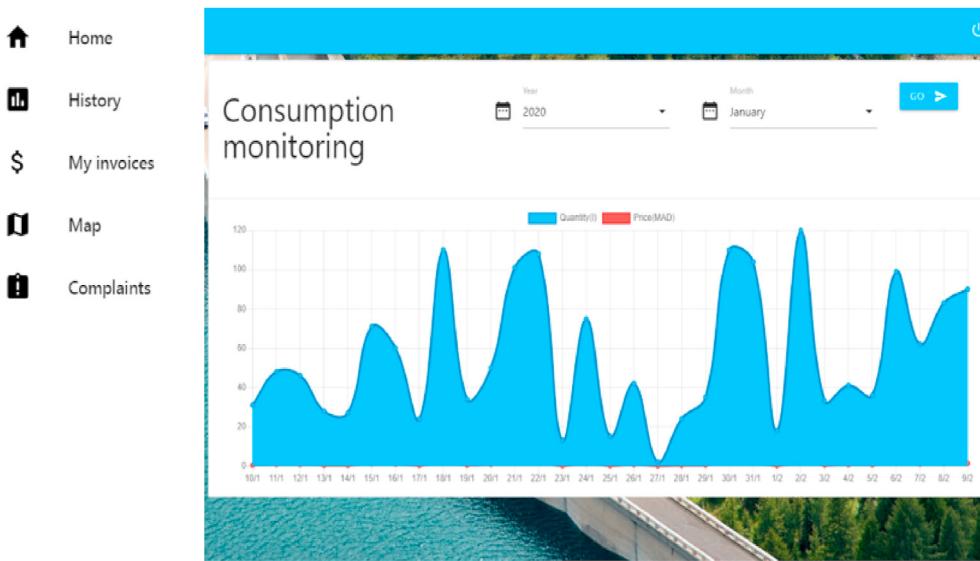


Fig. 18. Representation of daily customer consumption.

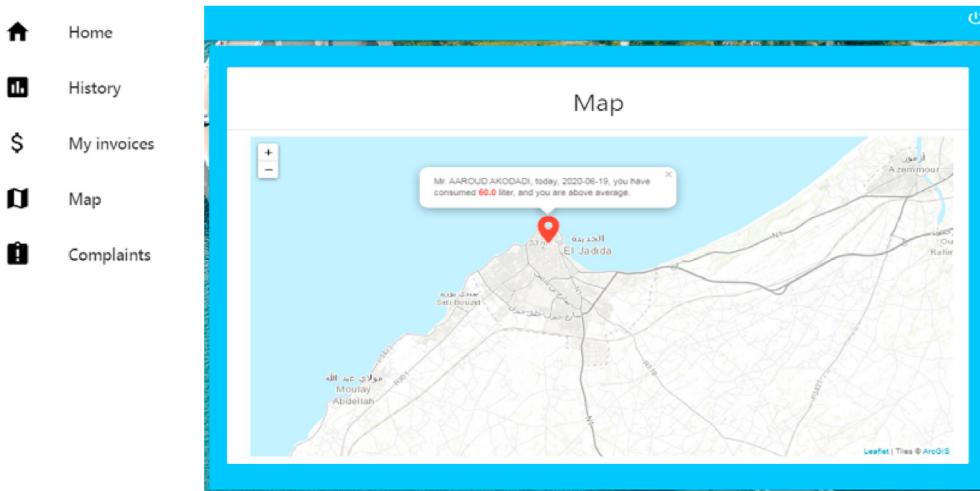


Fig. 19. State of customer consumption.

the SoftMax activation function to classify the digits with probabilistic values between 0 and 1.

5. MR-AMR dataset

5.1. Design methodology

MR-AMR (Fig. 14) is the largest digit meter dataset, created from different Moroccan meters. The MR-AMR dataset contains approximately 140000 high-quality digit meter images. The images in this unique dataset is captured by different types of Canon digital cameras in RGB format with the resolutions of 5184 x 3456 pixels and then converted to a binarised images. The collected image dataset is undoubtedly a precious resource for computer science researchers. In order to construct the MR-AMR digit dataset, only we made records in the water and electricity providers company. These records contain different types of water and electricity meters images. The process is shown in Figs. 11 and 12.

In the absence of an international standard for annotating datasets, we opted for a semi-automatic annotation that respects the following best practices:

- Ensure proper distribution of images by a label, to avoid situations such as where a label can have 5000 images while another has 400 images.
- Include a wide variety of images containing all possible digit states for each label.

Our annotation follows two phases, as shown in Fig. 13:

- Phase 1 (automatic): In this step, we apply an image processing algorithm to obtain a binary image.
- Phase 2 (semi-automatic): In this step, we check if there is an unwanted pixel in phase 1 results. Afterward, we process them either with an algorithm that eliminates these pixels (automatically) or processes this image manually, otherwise using the GNU Image Manipulation Program (GIMP) (manually).

The resulting images are resized to 28x28 pixels, which represent the same characteristics of the MNIST dataset (Fig. 15).

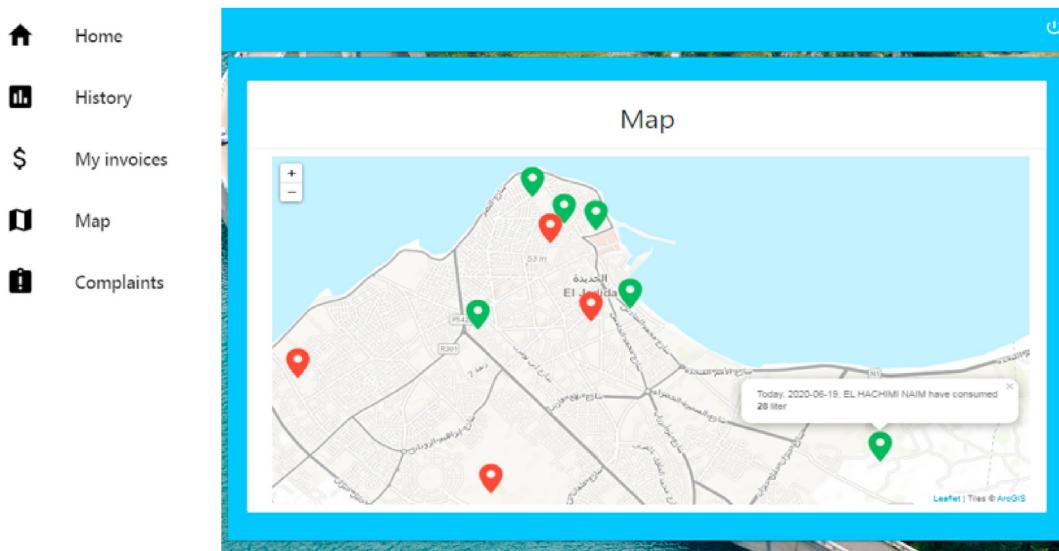


Fig. 20. Provider map of customers consumption.

Table 13

Materials and environment tools.

Front end	Back end	Embedded System
HTML5	Django Framework	OpenCV
CSS3	MySQL database	TensorFlow
JS jQuery		Keras
Material Design		Scikit-learn
		NumPy

5.2. Dataset annotation method

Images annotation in supervised machine learning is critical; it has a direct effect on the predicted result. So, images should be well-annotated, that machines can quickly and accurately recognize the digits. For this purpose and after dataset collecting, we classify images into ten classes

(0, 1, 2, 3, 4, 5, 6, 7, 8, 9) respecting the following logic (Table 8):

- Create ten folders with the name of the wanted class.
- An image containing a full-state digit is annotated with the class of this digit.
- A mid-state image is annotated with the class that appears partly in the top.

6. Performance evaluation of CNN model using different datasets

In this section, we evaluate the performance of our CNN model using the following datasets:

- MNIST (a public dataset of digits) [10],
- MR-AMR,
- MR-AMR combined with MNIST.

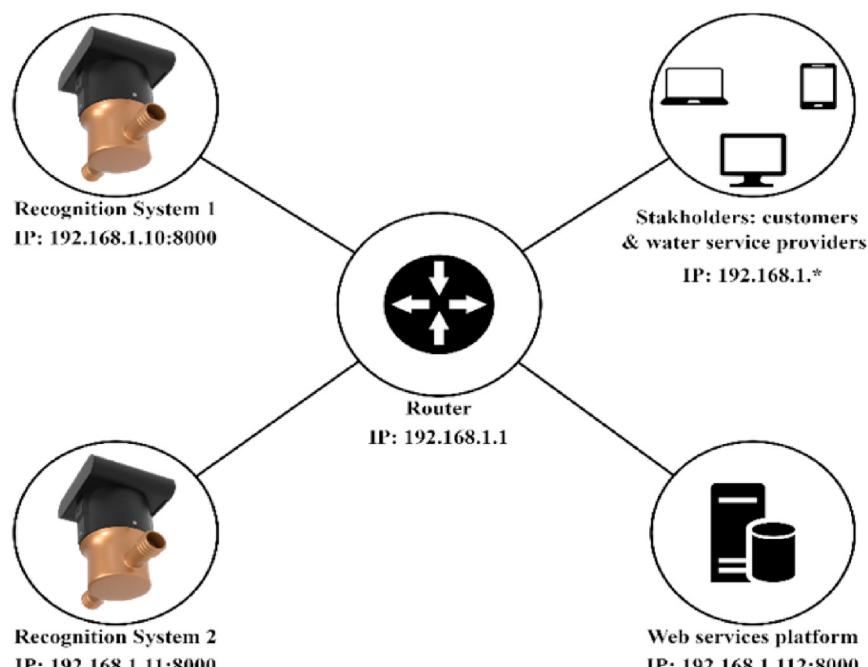


Fig. 21. Experiment architecture.



Fig. 22. Installation of RS in the household water meter.



Fig. 23. Installation of RS in UCD university water meter.

The dataset is divided into 85.8% for the training part and 14.2% for the testing. We used the same dataset percentage splitting in MNIST to go for a good comparison ([Table 9](#)).

[Tables 10, 11 and 12](#) show results for the three models.

To evaluate the performance of our model, we used the cross-validation method where we:

1. Shuffle the dataset randomly,
2. Split it into five groups,
3. For each unique group: take the group as a holdout or test dataset,
4. Take the remaining groups as a training dataset,
5. Fit the CNN model on the training set and evaluate it on the test set,
6. Retain the evaluation score and discard the model,
7. Evaluate the model by calculating the average accuracy, loss of all groups.

It is clear that with the MNIST dataset, the model cannot recognize mid-state images. Even if our model has the highest average accuracy ([Table 11](#)), it shows some errors with full-state images in the experimental context. Moreover, our dataset contains a more mid-state image than a full-state image. For this reason, we decide to use the combination of the two datasets (MNIST & MR-AMR), to get the straighten points of

both datasets. The result was auspicious in Training, Testing, and experiments, as detailed in section [7](#).

Although this promising result, we believe that an error of 1.93% may cause a problem either for customers or for water service providers, such as getting some incoherence states. In this sense, we implement an algorithm in the solution back-end to minimize and to prevent this kind of error from occurring. The principal idea is doing some verification before updating the actual water meter index, so we cannot accept an index that is less than the previous one.

7. Web services platform

The monitoring service allows the customer to track his consumption in real-time. In addition to displaying the actual water meter index ([Fig. 17](#)), this service also permits him to visualize the consumption curve (quantity in liters and the actual bill price) ([Fig. 18](#)).

Water meter data collection can be done in two different ways ([Fig. 16](#)):

- The customer sends a trigger event to start the recognition process via the platform (click on the button GET MY CONSUMPTION).
- The recognition process happens periodically.

In the experiment section, the period used is $T = 1\text{h}$, i.e., at each hour, the RS sends a request containing a customer identifier and the current water meter index.

On the server-side, if this information concerns the same day, the system updates the current consumption; otherwise, a new one is created. Therefore, the system stores the consumption index of the last hour of each day (instead of having 24 records per day).

In the case of a leak or abnormal consumption, an alert SMS will be sent to the customer on his mobile phone; this will help him to interact as quickly as possible. This preventive detection is mostly based on a comparison of current consumption to those preceding it.

As a future service, we plan to use a machine learning model and the historical data generated by the platform to be more accurate when it comes to understanding customer behavior.

The customer can also identify his state concerning his district or his city; by showing his water meter position on the map in green if it is less or equal to the average consumption or in red if he exceeds ([Fig. 19](#)). The customer can also follow the status of complaints, and generate invoices automatically and precisely. The water service provider, in turn, will be able to identify and visualize on the map the distribution of consumption across the covered territory.

Daily monitoring of consumption will provide useful insights and indicators about the consumption (Averages, Peak times) for better decision making ([Fig. 20](#)).

8. Implementation & experimentation

8.1. Programming environment description

All experiments were performed on an HP Z6 G4 workstation with an Intel ® Xeon ® Silver 4112 CPU 2.6 GHz, 16 GB of RAM, and an NVIDIA Quadro P400 GPU (12 GB of RAM).

The implementation was done with Python 3.6 using suitable libraries and frameworks, as shown in [Table 13](#).

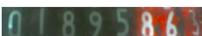
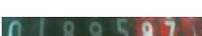
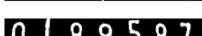
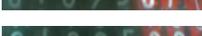
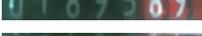
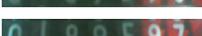
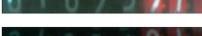
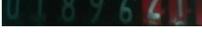
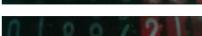
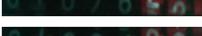
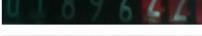
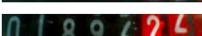
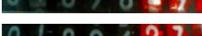
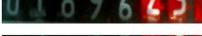
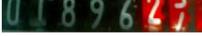
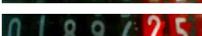
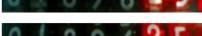
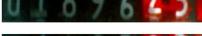
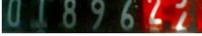
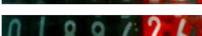
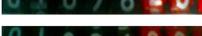
8.2. Experimentation

Before moving to a production environment, we deployed the system in a Local Area Network to check the functionality and interconnection between different modules following the architecture shown in [Fig. 21](#).

We first installed our prototype on water meters. ([Figs. 22 and 23](#)). The Recognition System listens permanently on port 8000 for any trigger event coming from customers and sends periodically (at each hour) the

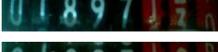
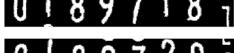
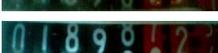
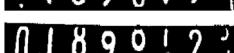
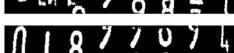
Table 14

Experimental results.

Line	Trigger event source	Date and time	Real water meter image index	Image Procession	RS predicted index	Platform index	Results State
1	Automatically	01-06-2020/08:00			01895857	01895857	Success
2	User	01-06-2020/8:25			01895863	01895863	Success
3	Automatically	01-06-2020/9:00			01895871	01895871	Success
4	User	01-06-2020/9:20			01895891	01895891	Success
5	Automatically	01-06-2020/10:00			01895921	01895921	Success
6	Automatically	01-06-2020/11:00			01895931	01895931	Success
7	User	01-06-2020/11:40			01895961	01895961	Success
8	Automatically	01-06-2020/12:00			01895974	01895974	Success
9	Automatically	02-06-2020/14:00			01896212	01896212	Success
10	User	02-06-2020/14:10			01896214	01896214	Success
11	Automatically	02-06-2020/15:00			01896217	01896217	Success
12	User	02-06-2020/15:15			01896219	01896219	Success
13	Automatically	02-06-2020/16:00			01896224	01896224	Success
14	User	02-06-2020/16:08			01896225	01896225	Success
15	Automatically	02-06-2020/17:00			01896228	01896228	Success
16	User	02-06-2020/17:02			01896229	01896229	Success
17	Automatically	02-06-2020/18:00			01896238	01896238	Success
18	Automatically	02-06-2020/19:00			01896239	01896239	Success
19	User	02-06-2020/19:45			01896247	01896247	Success
20	Automatically	02-06-2020/20:00			01896254	01896254	Success
21	User	02-06-2020/20:35			01896256	01896256	Success
22	Automatically	02-06-2020/21:00			01896239	01896256	Failed
23	User	02-06-2020/21:08			01896263	01896263	Success
24	Automatically	02-06-2020/22:00			01896266	01896266	Success
25	User	02-06-2020/22:52			01896888	01896888	Success
26	Automatically	02-06-2020/23:00			01896890	01896890	Success
27	User	02-06-2020/23:53			01896908	01896908	Success
28	Automatically	03-06-2020/00:00			01896909	01896909	Success
29	User	03-06-2020/00:30			01896913	01896909	Success
30	Automatically	03-06-2020/01:00			01896932	01896932	Success
31	User	03-06-2020/01:24			01896968	01896968	Success

(continued on next page)

Table 14 (continued)

Line	Trigger event source	Date and time	Real water meter image index	Image Procession	RS predicted index	Platform index	Results State
32	Automatically	03-06-2020/02:00			01896977	01896977	Success
33	Automatically	03-06-2020/12:00			01896999	01896999	Success
34	Automatically	03-06-2020/13:00			01897065	01897065	Success
35	Automatically	03-06-2020/14:00			01897129	01897129	Success
36	Automatically	04-06-2020/16:00			01897186	01897186	Success
37	Automatically	04-06-2020/17:00			01897203	01897203	Success
38	Automatically	04-06-2020/18:00			01897589	01897589	Success
39	Automatically	04-06-2020/19:00			01897998	01897998	Success
40	User	05-06-2020/20:04			01898725	01898725	Success
41	Automatically	05-06-2020/21:00			01899894	01899894	Success

collected water meter consumption data. The stakeholders can use all services provided by the platform server at IP address "192.168.1.112".

Table 14 shows the results of the experiment:

8.3. Discussion

In this experiment, we initialize the water meter index manually, and we test the system on several circumstances:

8.3.1. Camera resolution

We used two types of camera resolution:

- Normal Mode 640x480 pixels (from line 1 to 15 in Table 9)
- Full Mode 2592x1944 pixels (from line 16 to 42)

The RS is not affected by changing resolution and recognizes both types without any problems.

8.3.2. Light and brightness

The recognition process was performed during different dayparts (Morning, Afternoon, Night) to test the impact of light and brightness on recognition results.

- High brightness (from line 16 to 25 in Table 9)
- Low brightness (from line 9 to 15)

We remark that any level of light and brightness does not influence RS.

8.3.3. Trigger event

We tested the system using two possible modes of triggering events (manually or periodically). The system showed the same performances for both modes.

In line 22, the RS fails to recognize the last digit of the index (3 instead of 5). In this case, the RS does not update the index. It keeps the previous record 01896256 (line 21), in respect of a correct index algorithm that takes into account previous consumption before any update.

9. Conclusion & perspectives

In this paper, we have proposed a fully AI-based system to automate water meter data collection in Morocco country. The solution includes a Recognition System that consists of an embedded system with a Convolutional Neural Network using our proposed MR-AMR dataset, and a web services platform with several services such as consumption monitoring, detecting water leaks, visualizing water consumptions in a geographic map, reporting services, and potable water coverage. This solution helps save water and lead to sustainable use of this vital resource. The reliability of all systems, as demonstrated by experience, is 98.70%.

In future work, and to face the limits related to energy, we will focus on designing a turbine-based electricity generator to ensure the permanent system autonomy.

Furthermore, we have covered neither the system security nor transparency in this paper. Thus, we plan to use blockchain technology to deal with these issues.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Yang F, Jin L, Lai S, Gao X, Li Z. Fully convolutional sequence recognition network for water meter number reading. IEEE Access 2019. <https://doi.org/10.1109/ACCESS.2019.2891767>.
- [2] Gallo I, Zamberletti A, Noce L. "Robust Angle Invariant GAS Meter Reading. 2015. <https://doi.org/10.1109/DICTA.2015.7371300>.
- [5] Vanetti M, Gallo I, Nodari A. GAS meter reading from real world images using a multi-net system. Pattern Recogn Lett 2013. <https://doi.org/10.1016/j.patrec.2012.11.014>.
- [6] Cerman M, Shalunts G, Albertini D. A mobile recognition system for analog energy meter scanning. 2016. https://doi.org/10.1007/978-3-319-50835-1_23.
- [7] Quintanilha DBP, et al. Automatic consumption reading on electromechanical meters using HoG and SVM. 2018. <https://doi.org/10.1049/ic.2017.0036>.
- [8] Edward VCP. Support Vector Machine based automatic electric meter reading system. 2013. <https://doi.org/10.1109/ICCIC.2013.6724185>.
- [10] Lecun Y, Bengio Y, Hinton G. Deep learning. Nature 2015. <https://doi.org/10.1038/nature14539>.

- [11] Gomez L, Rusinol M, Karatzas D. "Cutting Sayre's Knot: Reading Scene Text without Segmentation. Application to Utility Meters; 2018. <https://doi.org/10.1109/DAS.2018.23>.
- [12] Gonçalves JC. "Reconhecimento de dígitos em imagens de medidores de consumo de gás natural utilizando técnicas de visão computacional," Master's thesis. Universidade Tecnológica Federal do Paraná – UTFPR; 2016.
- [19] Laroca R, Barroso V, Diniz MA, Gonçalves GR, Schwartz WR, Menotti D. Convolutional neural networks for automatic meter reading. J Electron Imag 2019. <https://doi.org/10.1117/1.jei.28.1.013023>.
- [20] Elrefaei IA, Bajaber A, Natheir S, Abusnab N, Bazi M. Automatic electricity meter reading based on image processing. 2015. <https://doi.org/10.1109/AEECT.2015.7360571>.
- [22] Rao R, Vrudhula S, Chang N. Battery optimization vs energy optimization: which to choose and when?. 2005. <https://doi.org/10.1109/ICCAD.2005.1560108>.
- [23] Kaup F, Gottschling P, Hausheer D, PowerPi ". Measuring and modeling the power consumption of the Raspberry Pi. 2014. <https://doi.org/10.1109/LCN.2014.6925777>.
- [26] Li C, Su Y, Yuan R, Chu D, Zhu J. Light-weight spliced convolution network-based automatic water meter reading in smart city. IEEE Access 2019. <https://doi.org/10.1109/ACCESS.2019.2956556>.
- [27] G. Salomon, R. Laroca, and D. Menotti, "Deep Learning for Image-based Automatic Dial Meter Reading: Dataset and Baselines," arXiv. 2020.
- [28] Laroca R, Araujo A, Zanlorensi L, de Almeida E, Menotti D. Towards Image-based Automatic Meter Reading in Unconstrained Scenarios: A Robust and Efficient Approach. 2020. " arXiv.
- [29] Zuo L, He P, Zhang C, Zhang Z. A robust approach to reading recognition of pointer meters based on improved mask-RCNN. 2020. <https://doi.org/10.1016/j.neucom.2020.01.032>. Neurocomputing.
- [30] Tsai CM, Shou TD, Chen SC, Hsieh JW. Use SSD to Detect the Digital Region in Electricity Meter. 2019. <https://doi.org/10.1109/ICMLC48188.2019.8949195>.
- [3] Gao Y, et al. Automatic water meter digit recognition on mobile devices. In: Int. Conf. Internet multimedia computing and service. Singapore: Springer; 2018. p. 87–95.
- [4] Kabalcı Y. A survey on smart metering and smart grid communication. Renew Sustain Energy Rev 2016. <https://doi.org/10.1016/j.rser.2015.12.114>.
- [9] Zhang Y, Yang S, Su X, Shi E, Zhang H. Automatic reading of domestic electric meter: an intelligent device based on image processing and ZigBee/Ethernet communication. J. Real-Time Image Process 2016. <https://doi.org/10.1007/s11554-013-0361-2>.
- [14] Jawas N, Indrianto. Image based automatic water meter reader. 2018. <https://doi.org/10.1088/1742-6596/953/1/012027>.
- [13] Nodari A, Gallo I. A multi-neural network approach to image detection and segmentation of gas meter counter. 2011.
- [15] Cai Z, Wei C, Yuan Y. An efficient method for electric meter readings automatic location and recognition. 2011. <https://doi.org/10.1016/j.proeng.2011.11.2548>.
- [16] Chouiten M, Schaeffer P. Vision-based mobile gas-meter reading machine learning method and application on real cases. In: Proceedings of scientific cooperations international workshops on electrical and computer engineering subfields; 2014.
- [17] Chouiten M, Schaeffer P. Vision-based mobile gas-meter reading machine learning method and application on real cases. In: Proceedings of scientific cooperations international workshops on electrical and computer engineering subfields; 2014.
- [18] Puttnies H, Altmann V, Golatowski F, Timmermann D. Cost-efficient universal approach for remote meter reading using web services and computer vision. 2015. <https://doi.org/10.1109/WoWMoM.2015.7158205>.
- [21] Son C, Park S, Lee JM, Paik J. Deep learning-based number detection and recognition for gas meter reading. IEIE Trans. Smart Process. Comput. 2019. <https://doi.org/10.5573/IEIESPC.2019.8.5.367>.
- [24] Shu D, Ma S, Jing C. Study of the automatic reading of watt meter based on image processing technology. 2007. <https://doi.org/10.1109/ICIEA.2007.4318804>.
- [25] Naim Ayman, Aaroud Abdessadek, Akodadi Khalid, El Hachimi, Chouaib. MR-AMR Dataset. Mendeley Data; 2020. p. V1. <https://doi.org/10.17632/8xjhrk9rx.1>.

Further reading

- [11] Gomez L, Rusinol M, Karatzas D. "Cutting Sayre's Knot: Reading Scene Text without Segmentation. Application to Utility Meters; 2018. <https://doi.org/10.1109/DAS.2018.23>.
- [12] Gonçalves JC. "Reconhecimento de dígitos em imagens de medidores de consumo de gás natural utilizando técnicas de visão computacional," Master's thesis. Universidade Tecnológica Federal do Paraná – UTFPR; 2016.
- [19] Laroca R, Barroso V, Diniz MA, Gonçalves GR, Schwartz WR, Menotti D. Convolutional neural networks for automatic meter reading. J Electron Imag 2019. <https://doi.org/10.1117/1.jei.28.1.013023>.
- [20] Elrefaei IA, Bajaber A, Natheir S, Abusnab N, Bazi M. Automatic electricity meter reading based on image processing. 2015. <https://doi.org/10.1109/AEECT.2015.7360571>.
- [22] Rao R, Vrudhula S, Chang N. Battery optimization vs energy optimization: which to choose and when?. 2005. <https://doi.org/10.1109/ICCAD.2005.1560108>.
- [23] Kaup F, Gottschling P, Hausheer D, PowerPi ". Measuring and modeling the power consumption of the Raspberry Pi. 2014. <https://doi.org/10.1109/LCN.2014.6925777>.
- [26] Li C, Su Y, Yuan R, Chu D, Zhu J. Light-weight spliced convolution network-based automatic water meter reading in smart city. IEEE Access 2019. <https://doi.org/10.1109/ACCESS.2019.2956556>.
- [27] G. Salomon, R. Laroca, and D. Menotti, "Deep Learning for Image-based Automatic Dial Meter Reading: Dataset and Baselines," arXiv. 2020.
- [28] Laroca R, Araujo A, Zanlorensi L, de Almeida E, Menotti D. Towards Image-based Automatic Meter Reading in Unconstrained Scenarios: A Robust and Efficient Approach. 2020. " arXiv.
- [29] Zuo L, He P, Zhang C, Zhang Z. A robust approach to reading recognition of pointer meters based on improved mask-RCNN. 2020. <https://doi.org/10.1016/j.neucom.2020.01.032>. Neurocomputing.
- [30] Tsai CM, Shou TD, Chen SC, Hsieh JW. Use SSD to Detect the Digital Region in Electricity Meter. 2019. <https://doi.org/10.1109/ICMLC48188.2019.8949195>.
- [3] Gao Y, et al. Automatic water meter digit recognition on mobile devices. In: Int. Conf. Internet multimedia computing and service. Singapore: Springer; 2018. p. 87–95.
- [4] Kabalcı Y. A survey on smart metering and smart grid communication. Renew Sustain Energy Rev 2016. <https://doi.org/10.1016/j.rser.2015.12.114>.
- [9] Zhang Y, Yang S, Su X, Shi E, Zhang H. Automatic reading of domestic electric meter: an intelligent device based on image processing and ZigBee/Ethernet communication. J. Real-Time Image Process 2016. <https://doi.org/10.1007/s11554-013-0361-2>.
- [14] Jawas N, Indrianto. Image based automatic water meter reader. 2018. <https://doi.org/10.1088/1742-6596/953/1/012027>.
- [13] Nodari A, Gallo I. A multi-neural network approach to image detection and segmentation of gas meter counter. 2011.
- [15] Cai Z, Wei C, Yuan Y. An efficient method for electric meter readings automatic location and recognition. 2011. <https://doi.org/10.1016/j.proeng.2011.11.2548>.
- [16] Chouiten M, Schaeffer P. Vision-based mobile gas-meter reading machine learning method and application on real cases. In: Proceedings of scientific cooperations international workshops on electrical and computer engineering subfields; 2014.
- [17] Chouiten M, Schaeffer P. Vision-based mobile gas-meter reading machine learning method and application on real cases. In: Proceedings of scientific cooperations international workshops on electrical and computer engineering subfields; 2014.
- [18] Puttnies H, Altmann V, Golatowski F, Timmermann D. Cost-efficient universal approach for remote meter reading using web services and computer vision. 2015. <https://doi.org/10.1109/WoWMoM.2015.7158205>.
- [21] Son C, Park S, Lee JM, Paik J. Deep learning-based number detection and recognition for gas meter reading. IEIE Trans. Smart Process. Comput. 2019. <https://doi.org/10.5573/IEIESPC.2019.8.5.367>.
- [24] Shu D, Ma S, Jing C. Study of the automatic reading of watt meter based on image processing technology. 2007. <https://doi.org/10.1109/ICIEA.2007.4318804>.
- [25] Naim Ayman, Aaroud Abdessadek, Akodadi Khalid, El Hachimi, Chouaib. MR-AMR Dataset. Mendeley Data; 2020. p. V1. <https://doi.org/10.17632/8xjhrk9rx.1>.