Digitisation of Conventional Water Meters using Automated Number Recognition

Xue Jun Li
Department of Electrical and Electronic Engineering
Auckland University of Technology
Auckland, New Zealand
xuejun.li@aut.ac.nz

Yi Gao
Department of Embedded Systems
MIIVII Technology Inc.
Beijing, China
gy008608@hotmail.com

***Abstract*—It is desired to enhance traditional mechanical water meters with automatic meter reading (AMR) in the era of Internet of Things (IoT). In this paper, we propose and implement an add-on module using optical character recognition (OCR) to achieve that. The proposed module will be attached to a conventional mechanical water meter, capture and translate the register readings into plain text, which can be sent wirelessly to a database server. In addition, we propose a novel light-weight optical character recognition algorithm and test it in a prototype with embedded digital system. Experimental results show that the recognition accuracy can be as high as 98.1%, which indicates a promising candidate technology for AMR to revolutionise water meters. Finally, this technique can be applied to other mechanical meters with analogue number displays, paving the way for ubiquitous IoT.**

***Index Terms*—Automated meter reading, water meter, optical character recognition, embedded systems, computer vision.**

I. Introduction

Internet of Things (IoT) refers to a ubiquitous networking paradigm that connects various devices/sensors for massive sensing with Big Data for the applications like smart cities, including smart home [1], intelligent transportation systems (ITS) and environmental monitoring. One typical example is the automatic meter reading (AMR) system. It enables automatic collection of consumption, diagnosis, and status data from metering devices, which is then uploaded to a central database for billing, troubleshooting, and analysis [2].

Nowadays, despite the fact that many countries have adopted AMR system for electricity billing, water billing still relies on traditional mechanical water meters, which needs manual transcribing by a human operator [3], leading to low efficiency, unavoidable error and high manpower cost. How to enable AMR for traditional mechanical water meters remains an attractive research topic for both academia and industrial communities. Several pioneer works [4] were reported to replace conventional mechanical waters by electronic smart meters, whose cost unfortunately becomes the main obstacle that prevents its large-scale deployment.

Meanwhile, digital signal processing has been evolving rapidly over the past two decades, among which digital image processing has been playing an important role in many research fields. As we know, digital image processing includes multi-scale signal analysis, classification, feature extraction, projection and pattern recognition. Among these techniques, pattern recognition is often adopted to extract meaningful information from digital images, leading to its popular application in image analysis.

Enlightened by the power of digital image processing, we view the stringent requirement of connecting conventional mechanical water meters to IoT from another angle. *Is it possible to enable AMR functionality through image recognition without replacing an existing mechanical water meter?* It is desired to integrate the advantages of existing mechanical water meters, such as high reliability, good accuracy and negligible maintenance and the advantages of electronic water meters, such as AMR functionality and real-time information collection. Consequently, we propose to digitise an existing mechanical water meter by adding an electronic module, which captures the numerical display of the water meter as image and converts the image into plain text through optical character recognition (OCR).

Through literature review, we found that OCR on printed characters seems to be a perfect candidate to digitise a mechanical water meter. However, it requires costly system resources, such as large memory space and fast central processing unit (CPU) speed. To make our proposal feasible, we need to design a light-weight OCR algorithm that can achieve adequate OCR accuracy under the constraint of the corresponding hardware with limited memory space and CPU speed. Our contributions are three-folds: (i) We proposed a low-complexity OCR algorithm, namely Moment Template Matching, which combines the advantage of template matching and geometric moments. (ii) We implemented the proposed OCR algorithm in an embedded system using ATmega1284p. (iii) We compared the proposed OCR algorithm with existing algorithms. The rest of this paper is organized as follows. Section II presents related work, and Section III presents our proposed light-weight OCR algorithm. Software design is presented in Section IV. Section V presents the hardware implementation with the experimental results and Section VI concludes the paper with future work.

II. Related Work

OCR refers to the method using image processing to recognise printed or handwritten text characters in digital images of physical documents [5]. On a computer-based platform with powerful CPU and abundant memory space, recent development of OCR algorithms tend to be even more complex by taking machine learning (e.g., deep learning) into consideration.

A. Image Processing

The main steps in digital image processing include transformation, compression, enhancement, recovery, segmentation, representation, description and object recognition [6].

1) Image Binarisation: Image binarisation significantly reduces the data size to highlight the target contour. In a colour image, each pixel can be constituted by RGB data or YUV data. R, G and B stands for red, green and blue, respectively. While Y, U and V represents luminance, chrominance and chromam, respectively [7]. Mathematically, $Y = 0.59G + 0.31R + 0.11B$, $U = R - Y$ and $V = B - Y$.

Each pixel in a grayscale image only has a luminance (Y) value between 0 and 255. To highlight the target contour [8], we can choose a threshold value and binarise the image [9]. The key challenge here is to find the appropriate threshold value to extract the target contour.

2) Noise Removal: Gaussian noise and salt and pepper noise are common in digital images [10]. The former appears during the image capture step, caused by poor lighting and/or high-temperature sensor noise [11], while the latter, also known as impulse noise, is a kind of random white or black point in the image. For noise removal, we can use typical filters like Mean Filter, Gaussian Filter and Median Filter [12], [13].

3) Image Segmentation: Image segmentation divides an image into specific areas, each with a processing target inside [14]. Common methods include threshold segmentation, region growth, edge detection, and projection.

- *Threshold Segmentation*—For a pixel in the target region, $g(i,j) = 1$, and for all other pixels, $g(i,j) = 0$. Due to its low computational complexity, the threshold segmentation method is widely used in applications that require high operation efficiency [15].
- *Region Growth*—Region growth assembles small areas that are sufficiently similar into larger areas [16].
- *Splitting and Merging*—It is almost the reverse of Region Growth method. It starts from the whole image, segments it and then combines similar areas, and finally achieves the region of interest [17].
- *Edge Segmentation*—Edge segmentation is achieved by edge detection, by identifying the mutation of greyscale, which indicates the end of a region and the beginning of another region [18]. Differential operator is commonly used for edge detection, but it is sensitive to noise and only suitable for simple images.
- *Projection*—The projection method [19] is based on eigen function. It simplifies the pixel distribution characteristics of a two-dimensional (2D) image to two one-dimensional (1D) functions. The projection method was chosen for image segmentation in this paper.

B. Recognition Algorithm

Several recognition algorithms were proposed for OCR in the literature, which are based on pixel values, pixel location, geometric moments and machine learning.

1) Template Matching: Template matching (TM) [20] attempts to find parts of an image which match a template image [21], [22]. The similarity between template T and pattern S as Euclidean distance is given by

$$D\left(i,j\right) = \sqrt{\sum_{w=0}^{W-1}\sum_{h=0}^{H-1}\left[S\left(i+w,j+h\right) - T\left(w,h\right)\right]^2} \quad (1)$$

The Euclidean distance in (1) can be simplified to a normalised correlation coefficient

$$R\left(i,j\right) = \frac{\sum\limits_{w=0}^{W-1}\sum\limits_{h=0}^{H-1} S\left(i+w,j+h\right)T(w,h)}{\sqrt{\sum\limits_{w=0}^{W-1}\sum\limits_{h=0}^{H-1} S^2\left(i+w,j+h\right)}\sqrt{\sum\limits_{w=0}^{W-1}\sum\limits_{h=0}^{H-1} T^2(w,h)}} \quad (2)$$

where $0 \leq R(i,j) \leq 1$. After a set of templates T is completely searched, the template corresponding to the maximum correlation coefficient value $R_{\max}(i_m, j_m)$ matches $S_{im,jm}$ [21], [23].

To reduce complexity, pixel difference method adopts

$$E(i,j) = \sum_{w=0}^{W-1}\sum_{h=0}^{H-1}\left|S(i+w,j+h) - T(w,h)\right| \quad (3)$$

The template associated with the minimum $E(i,j)$ value matches the input image.

2) Projection Template Matching: Based on (3), vertical/horizontal projection template matching (PTM) was proposed.

$$\begin{cases} E_v\left(i,j\right) = \sum\limits_{w=0}^{W-1}\left|\sum\limits_{h=0}^{H-1} S\left(i+w,j+h\right) - \sum\limits_{h=0}^{H-1} T\left(w,h\right)\right| \\ E_h\left(i,j\right) = \sum\limits_{h=0}^{H-1}\left|\sum\limits_{w=0}^{W-1} S\left(i+w,j+h\right) - \sum\limits_{w=0}^{W-1} T\left(w,h\right)\right| \end{cases} \quad (4)$$

As the PTM method compares the sum of pixel values in a column (vertical projection) or a row (horizontal projection), it relaxes the memory requirement and decreases the computational complexity. However, it does not work when image rotation or size change is involved.

3) Geometric Moments: Geometric moments are also known as Cartesian moments [24]. Moments describe the arrangement of image pixels, compactness, irregularity and higher-order descriptions together [25]. The first-order geometric moment of image calculates the centroid coordinates. The second-order geometric moment can handle image translation, scale and rotation. Moment of order $(p+q)$ is independent of scaling, translation, rotation, and even grey-level transformations. For discrete images, moment m_{pq} of a function $I(x,y)$ is approximated by

$$m_{pq}(I) = \sum_x\sum_y x^p y^q I(x,y) \quad (5)$$

where (x, y) are the coordinates. Seventh-order geometric moment characteristics are usually adopted to have both rotation invariant and scale-invariant property [25].

4) Machine Learning: Machine learning has been recently introduced to digital image processing [26]. However, it results high computational cost, thus not suitable to digitise conventional water meters.

C. Analysis of Existing OCR Algorithms

Preliminary study was performed to investigate the feasibility of implementing TM, PTM and GM methods in a microcontroller-based system to perform OCR on a conventional water meter display.

For the TM method, images of numbers from 0 to 9 were extracted as template images, each of which has a pixel resolution of 26 (height)×19 (width). Next, the TM formula in (3) was applied to calculate the corresponding error values between the input image with ten templates. Finally, the input image was recognised as the number in the template associated with the minimum error value.

TM method has its advantages and disadvantages. It has high recognition accuracy as the matching is performed on a pixel-by-pixel level. Nevertheless, there are several drawbacks. First, storage of these number templates occupies large memory space, e.g., 4,940 bytes are required for the ten templates. Let us consider ATmega1284P with static random-access memory (SRAM) space of 16K bytes. If we take the received image into account, the total required memory space will exceed its maxmimum memory size. Next, the computational complexity of template matching for a single number is given by $T = 3WHN_t - 1$, where N_t is the number of templates, each with a resolution of $H \times W$ pixels. With $W = 19$, $H = 26$ and $N_t = 10$, the TM method requires 14,819 steps to perform a single number recognition.

The PTM method can be used to reduce the memory requirement. With the TM method, ten number templates occupy 4940 bytes. For the PM method, one template occupies 19 bytes (by vertical projection) plus 26 bytes (by horizontal projection), i.e., a total of 45 bytes. Thus, ten number templates totally require 450 bytes, which is much less than that of the TM method.

Similarly, after segmentation, the input image will be processed using the projection method as described in (4). With projection method, computational complexity for a single number recognition is reduced to

$$\begin{cases} T = 3WN_t - 1, \text{for vertical projection} \\ T = 3HN_t - 1, \text{for horizontal projection} \end{cases} \tag{6}$$

With $W = 19$, $H = 26$ and $N_t = 10$, it requires 569 and 779 steps to perform a single number recognition using vertical projection and horizontal projection, respectively.

However, projection method has an obvious drawback of losing the pixel location information. When a number is mirror-reversed version of another number, their projection templates become the same. For example, the PTM method cannot differentiate number 6 from number 9.

The GM method can be used to overcome the drawback of the PTM method. GM is often used to recognise high-resolution images, and even those images are rotated or scaled. In this paper, the difference of the geometrical characteristics of numbers is very small, not to mention that the numbers 6 and 9 are with the same geometric moment values. For comparison study, we choose the following error formula with second-order geometric moments m_{20} and m_{02} as calculated in 5,

$$E = \sqrt{\left[m_{20}(S) - m_{20}(T)\right]^2 + \left[m_{02}(S) - m_{02}(T)\right]^2} \tag{7}$$

where S and T are the input image and template image, respectively. The computational complexity of the GM method for a single-number recognition is given by $T = (4WH + 5)N_t - 1$. With $W = 19$, $H = 26$ and $N_t = 10$, GM method requires 19,809 steps to perform a single number recognition. Therefore, it is not favourable to use the GM method.

From (5), geometric moment algorithm actually depends on the position of each pixel in the whole image. Consider the first order geometric moment with $p = 0$ and $q = 1$:

$$m_{01} = \sum_x \sum_y x^0 y^1 I(x, y) \tag{8}$$

Consequently, m_{01} includes information about pixel quality and location information. However, information loss happens at the summation step. Without the last summation step, one can use projection to conserve the location information. This motivates us to propose moment template matching (MTM), which combines the template matching algorithm and geometric moment algorithm so that high recognition accuracy can be achieved with low computational complexity and relaxed memory space requirement.

III. Proposed Moment Template Matching

Consider an image I of size $H \times W$, we can use the quasi-moment function in (9) to convert it into a $1 \times W$ array, which can be treated as 2D to 1D transformation. Then, recognition was performed based on the similarity between an input image and a number template in the transformed domain.

$$I^P(w) = \sum_{h=0}^{H-1} hI(w, h), \quad w \in [0, W-1] \tag{9}$$

where $I(x, y)$ is the pixel value at location $(x,\ y)$. I^P is a 1D array, each of whose elements is the sum of the product of the corresponding pixel values in that column with the y-coordinate values. Note that this is similar to the vertical projection method, but I^P contains information about both pixel quality and pixel location.

$$E = \sum_{w=0}^{W-1} \left|S^P(w) - T^P(w)\right| \tag{10}$$

With (10), we can find out the similarity between the input image and each of ten number image templates in

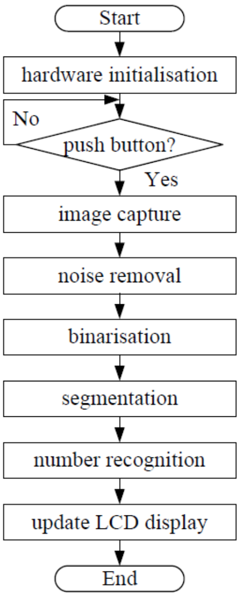


Fig. 1. Program Flowchart

the transformed domain. Mathematically, the error distance function for the proposed MTM algorithm is given by

$$E_k(i,j) = \sum_{w=0}^{W-1} \left| \sum_{h=0}^{H-1} hS(i+w, j+h) - \sum_{h=0}^{H-1} hT_k(w,h) \right| \tag{11}$$

where S is the input image, T_k is the k^{th} template array. E_k is the error distance between the input image and the k^{th} template. The template associated with the smallest error distance value gives the recognised number of the input image.

Consider N_t templates with size of $H \times W$, the required number of steps to recognise a number using the proposed MTM algorithm is $T = N_t(2WH - W + 1) - 1$. With $W = 19, H = 26$ and $N_t = 10$, the proposed MTM method requires 9,699 steps to perform a single number recognition.

IV. Software Design

We use ATmega1284p with Atmel in-circuit emulator (ICE) for the prototype. Fig. 1 illustrates the program flowchart. Upon powering up, the program initialises the input and output serial port, I2C protocol, camera and liquid crystal display (LCD) setup.

Three types of data are sent through I2C protocol to the peripheral components, which include peripheral component slave address, register address and register data. The camera OV7620 operates at 27MHz, while the microcontroller's clock speed is 20MHz. We need to use a prescaler to reduce the camera clock speed. The default resolution of OV7620 is 640×480 pixels, but we are only interested in the water meter reading number display. Through preliminary tests, we found the best image size is 151×41 pixels. Smaller image size can reduce data collection time as well as the required SRAM size.

To capture the image through OV7620, an event-triggered while loop is designed to wait for the vertical sync pulse (VSYN) signal, which indicates the starting bit of an image. Next, the horizontal valid data output window signal (HREF) is checked. Transition of the HREF signal from low to high indicates the starting bit of a new row. Finally, we check the PCLK signal. Once the PCLK signal goes from high to low, the microcontroller accepts the pixel data from the data bus.

It takes time for the microcontroller to receive image data and store them in memory. In addition, the arrival of next image is random. The next image may arrive while the current image is being saved. To solve this issue and avoid data corruption, the code is designed to accept the column data instead of row data. When a new row starts, the microcontroller only accepts one row of data, then waits for the next row. Consequently, the microcontroller has enough time to store the data of a complete row to its memory. The first frame only accepts the first column of data, and the second frame only accepts the second column of data, and so on. Thus for an image of 151×41 pixels, it requires 151 frames. In particular, as compared to the row data collection, the column data collection method can effectively avoid data loss.

After receiving the whole image, it performs noise removal before image binarisation. We use the median filter to remove the salt and pepper noise. The noise removal function is composed of two parts. First, we set a removable window on the image. The window size is 3×3. Then, we sort the nine numbers using the center value of that window.

The critical task of binarisation is to find the correct binarisation threshold. To find the optimal threshold, we use Matlab to process the image after noise removal. The image is downloaded to a personal computer (PC) via the Universal Asynchronous Receiver/Transmitter (UART) port. Matlab can display the data as an image, which allows us to observe whether the processing is going as expected.

With the optimal binarisation threshold from Matlab, we segment the image and recognize the numbers one by one. Therefore, segmentation (using horizontal/vertical projection) is required to extract each number out of the whole image.

After image segmentation, we can recognize a number by using the proposed MTM algorithm in Section III. In brief, we transform a 2D image into a 1D array using (9), and then compare it with a template, which is the predetermined data for numbers from 0 to 9. The template associated with the smallest error distance corresponds to the recognised number. By looping the recognition step four times, the significant digits in the display of the considered water meter are recognised.

V. Results and Discussions

In the implemented prototype using ATmega1284p, upon the push-button signal, it will activate camera control, receive the image data and display the recognised reading numbers on an LCD. Fig. 2 shows the block diagram of the implemented prototype. Fig. 3 shows the photo of the prototype. For the hardware setup, the prototype is attached to the water meter.

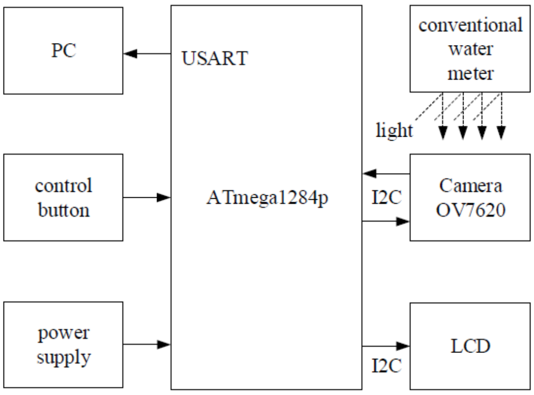


Fig. 2. Hardware Components

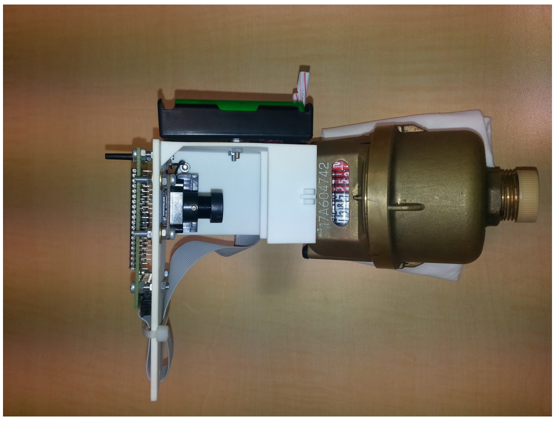


Fig. 3. Implemented Prototype with Water Meter

The camera faces towards the water meter display, and their relative position is fixed.

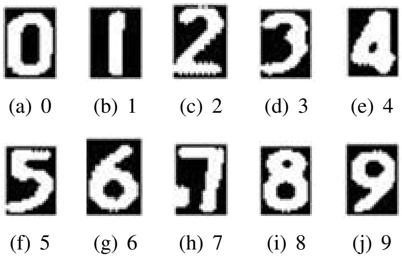
From the prototype, we can extract the number templates as shown in Fig. 4. Matlab simulation is performed to compare the proposed MTM method with TM, PTM and GM methods, using TM algorithm as the benchmark.

There are four significant digits of the number displayed on the water meter, ranging from 0 to 9. Each number is chosen three times for the simulation and a total of 2000 sets of numbers are used. For OCR systems, the recognition accuracy (RA) is calculated as

$$RA = \frac{N_c}{N} \times 100\% \tag{12}$$

where N_c and N denote the number of correctly recognised digits and the number of total image inputs, respectively.

Next, we compare algorithm run time[1] to check the computational complexity of each algorithm. The results are obtained with a PC with Intel® Core™ i7-8700 Dual CPU @3.2GHz @3.19GHz, 32GB of RAM, 64-bit Windows 10 Operating System, Matlab R2018b (Version 9.5.0.944444). Table I summarises the performance comparison results.



(a) 0 (b) 1 (c) 2 (d) 3 (e) 4

(f) 5 (g) 6 (h) 7 (i) 8 (j) 9

Fig. 4. Number Template Images

TABLE I
PERFORMANCE COMPARISON

Algorithm	RA	Run Time (s)	Template Size (bytes)
TM	99.9%	0.00574	4940
PTM	95.3%	0.00232	450
GM	33.6%	0.00765	80
This Work-MTM	98.1%	0.00375	760

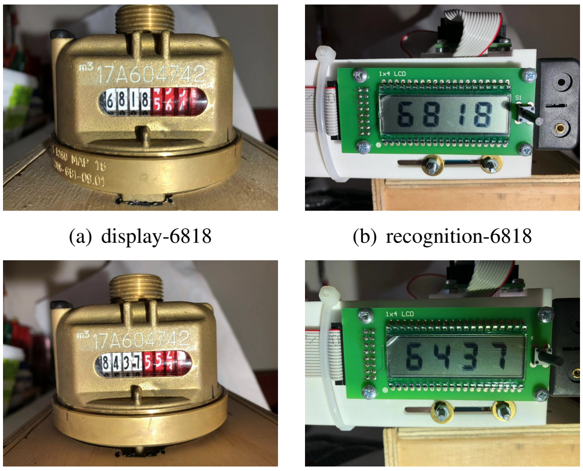
In Table I, the TM algorithm provides the highest recognition accuracy at the cost of high computational complexity and high memory resources, preventing us to adopt this algorithm in microntroller-based applications. Although GM algorithm has translation, scale and rotation invariant features, it results in the lowest recognition accuracy. Thus, it is not suitable for our considered application. The PTM algorithm compresses a 2D image to a 1D array using vertical/horizontal projection, which leads to lost pixel and coordinate information. Consequently, its accuracy is lower than that of the TM algorithm.

For the proposed MTM algorithm, it combines the geometric moment and vertical projection, conserving the coordinate information during the transformation step. It achieves a higher RA than the PTM algorithm, while requiring less memory than the TM algorithm.

Fig. 5 shows the examples of water meter reader results and Fig. 5(d) illustrates a recognition error. The error happened during the first number recognition by mistaking number 8 as number 6.

The experimental results show that the MTM algorithm can work properly to recognise the numbers showing on a conventional water meter display in a resource-limited microcontroller-based OCR system. Nevertheless, we noticed the recognition result using the implemented prototype with ATmega1284p is not as good as using a PC running Matlab. The possible reasons are: (i) Conversion of the MTM algorithm from the Matlab code to the microcontroller-based code results possible discrepancy. (ii) The fixture between meter and reader is not rigid enough. If one presses the button too hard, the whole prototype will shake, which will result in subsequent number recognition error.

[1] It refers to the average time required for a single digit recognition in Matlab simulation.



(a) display-6818 (b) recognition-6818

(c) display-8437 (d) recognition-6437

Fig. 5. Recognition Examples

VI. Conclusion

This paper proposed a lightweight optical character recognition algorithm, namely Moment Template Matching, and presented its implementation in a microcontroller based prototype to digitise a conventional water meter for automated meter reading in the context of IoT. For the image recognition, we first use an camera to capture an image of the register display on the mechanical water, and then obtain the recognition target by noise removal, binarisation, and segmentation. In the Matlab simulation on a PC, the proposed algorithm can achieve an accuracy of 98.1% at a run time of 0.00375 seconds per number character.

For future work, we consider the following items:

1) In the experiment, we found the case was out of shape slightly, which would affect the recognition accuracy. Rigid materials can be used, and the joints between the water meter and the OCR prototype can be redesigned to improve the stability.
2) We can optimise the program logic, and improve the noise reduction performance.
3) Camera OV7620 has many parameters for exposure settings. Different combinations of them could be tuned to improve the output image quality, thus the recognition accuracy.

Acknowledgment

The authors would like to thank Jian Huang for his help in the PCB design.

References

[1] W. Li, T. Logenthiran, V. Phan, and W. L. Woo, “A novel smart energy theft system (sets) for IOT-based smart home,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5531–5539, June 2019.

[2] Q. Liu, N. Linge, and V. Lynch, “Implementation of automatic gas monitoring in a domestic energy management system,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 3, pp. 781–786, 2012.

[3] Watercare, “Meter readings - get information on estimated, final and special readings,” 2019. [Online]. Available: https://www.watercare.co.nz/Help-and-advice/Help-with-your-account/Meter-readings.

[4] X. J. Li and P. H. J. Chong, “Design and Implementation of a Self-Powered Smart Water Meter,” *Sensors*, vol. 19, no. 19, 26 September 2019.

[5] J. Song, Z. Li, M. R. Lyu, and S. Cai, “Recognition of merged characters based on forepart prediction, necessity-sufficiency matching, and character-adaptive masking,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 1, pp. 2–11, 2005.

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2^{nd} Ed. Prentice Hall, 2002.

[7] R. J. Patan, M. Chakraborty, and S. S. Devi, “Blind color image de-convolution in YUV space,” in *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2016, pp. 1551–1555.

[8] K. Shin, I. Jang, and N. Kim, “Block adaptive binarisation of ill-conditioned business card images acquired in a PDA using a modified quadratic filter,” *IET Image Processing*, vol. 1, no. 1, pp. 56–66, 2007.

[9] T. Sund and K. Eilertsen, “An algorithm for fast adaptive image binarization with applications in radiotherapy imaging,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 1, pp. 22–28, 2003.

[10] Y.-M. Huang, M. K. Ng, and Y.-W. Wen, “Fast image restoration methods for impulse and gaussian noises removal,” *IEEE Signal Processing Letters*, vol. 16, no. 6, pp. 457–460, 2009.

[11] A. H. Sable and K. Jondhale, “Modified double bilateral filter for sharpness enhancement and noise removal,” in *2010 International Conference on Advances in Computer Engineering*. IEEE, 2010, pp. 295–297.

[12] Ursina Caluori, Klaus Simon,“DETEXTIVE optical character recognition with pattern matching on-the-fly,”*Pattern Recognition*, vol. 48, No. 3, pp. 827–836, 2015.

[13] N. Singh, T. Thilagavathy, R. T. Lakshmipriya, and O. Umamaheswari, “Some studies on detection and filtering algorithms for the removal of random valued impulse noise,” *IET Image Processing*, vol. 11, no. 11, pp. 953–963, 2017.

[14] M. A. U. Ekram, A. Chaudhary, A. Yadav, J. Khanal, and S. Asian, “Book organization checking algorithm using image segmentation and ocr,” in *2017 IEEE* 60^{th} *International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2017, pp. 196–199.

[15] I.-H. Lee and M. T. Mahmood, “Robust registration of cloudy satellite images using two-step segmentation,” *IEEE Geoscience And Remote Sensing Letters*, vol. 12, no. 5, pp. 1121–1125, 2015.

[16] L. Shen and R. M. Rangayyan, “A segmentation-based lossless image coding method for high-resolution medical image compression,” *IEEE Transactions on Medical Imaging*, vol. 16, no. 3, pp. 301–307, 1997.

[17] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos, “Hybrid image segmentation using watersheds and fast region merging,” *IEEE Transactions on Image Processing*, vol. 7, no. 12, pp. 1684–1699, 1998.

[18] A. Pratondo, C.-K. Chui, and S.-H. Ong, “Robust edge-stop functions for edge-based active contour models in medical image segmentation,” *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 222–226, 2015.

[19] C.-N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, “License plate recognition from still images and video sequences: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 377–391, 2008.

[20] S. Mori, C. Y. Suen, and K. Yamamoto, “Historical review of ocr research and development,” *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, 1992.

[21] B. Roberto, *Template Matching Techniques in Computer Vision: Theory and Practice*. John Wiley & Sons., 2009.

[22] I. Talmi, R. Mechrez, and L. Zelnik-Manor, “Template matching with deformable diversity similarity,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 175–183.

[23] M. Sonka, V. Hlavac, and R. Boyle., *Image Processing, Analysis, and Machine Vision.* Cengage Learning,2014.

[24] Y. Wang, “The geometric moment and its invariants,” *Journal of Shanghai Dianji University*, vol. 9, no. 2, pp. 7–10, 2006.

[25] M. Nixon and A. S. Aguado, *Feature extraction and image processing for computer vision*. Academic Press, 2012.

[26] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–20, 2019.