Mobile-based text recognition from water quality devices

Shanti Dhakal, Maryam Rahnemoonfar
School of Engineering and Computing Sciences, Texas A&M University-Corpus Christi

ABSTRACT

Measuring water quality of bays, estuaries, and gulfs is a complicated and time-consuming process. YSI Sonde is an instrument used to measure water quality parameters such as pH, temperature, salinity, and dissolved oxygen. This instrument is taken to water bodies in a boat trip and researchers note down different parameters displayed by the instrument's display monitor. In this project, a mobile application is developed for Android platform that allows a user to take a picture of the YSI Sonde monitor, extract text from the image and store it in a file on the phone. The image captured by the application is first processed to remove perspective distortion. Probabilistic Hough line transform is used to identify lines in the image and the corner of the image is then obtained by determining the intersection of the detected horizontal and vertical lines. The image is warped using the perspective transformation matrix, obtained from the corner points of the source image and the destination image, hence, removing the perspective distortion. Mathematical morphology operation, black-hat is used to correct the shading of the image. The image is binarized using Otsu's binarization technique and is then passed to the Optical Character Recognition (OCR) software for character recognition. The extracted information is stored in a file on the phone and can be retrieved later for analysis. The algorithm was tested on 60 different images of YSI Sonde with different perspective features and shading. Experimental results, in comparison to ground-truth results, demonstrate the effectiveness of the proposed method.

Keywords: Document image extraction, image processing.

1. INTRODUCTION

Mobile devices have become a necessity for general population as the help us meet our communication and social networking needs. The popularity of mobile devices such as smart phones, tablets, has been growing exponentially. Today, a large number of research and development activities are targeted towards the mobile devices. Several applications are developed for the mobile platforms that are aimed at making our lives simpler and easier.

In this study, an application for a smart phone (Android phone) is developed, that processes as image and extracts information from the image. The main idea behind the application is to help the water quality researchers get quick information from the water quality device such as YSI Sonde by taking a picture of the device through the application and extracting the information displayed by the device. YSI Sonde is an instrument is used for water quality monitoring. The main use of YSI Sonde is to measure water quality parameters such as pH, temperature, salinity, and dissolved oxygen, etc. The objective of the application is to extract these water quality information from the image of the device monitor. The mobile application is developed for Android platform using Java Native interface with C++ and OpenCv functions. Image text extraction can be used in several areas such as - road sign detection and translation [1], detecting book cover and product packaging [1], text to speech conversion on mobile devices [2], and scene and billboard text reading [3]. The authors in [2] devised a system to acquire image using a web camera and detect text in the image using color and morphological information to aid visually impaired people by extracting text and translating text to speech.

Several techniques have been developed on text segmentation from images. The authors in [1], [2] use Otsu's binarization technique to binarize the image. For this purpose, the authors divide the image into multiple small windows and apply binarization techniques to each window. They use Fischer's Discriminant Rate (FDR) to locate the text area in the image and select and extract the connected components from the image. To extract and (recognize image, the connected components are fed to OCR software. The OCR finally recognizes the individual characters from the image. In [3], the authors use adaptive binarization and color clustering techniques to extract text region from the scene images.

Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2015, edited by Reiner Creutzburg, David Akopian, Proc. of SPIE-IS&T Electronic Imaging, Vol. 9411, 941105

Fuzzy c-means (FCM) clustering is used in [4] to segment image into text and no-text region and then connected component analysis is applied to extract text after binarization. The authors in [5] presented text extraction method where text is segmented from background using intensity variation and color variance.

The paper is organized as follows. Section 2 discusses the image to text extraction methodology used in this study. The results and discussion are presented in section 3. Section 4 concludes the paper and is followed by the list of references.

2. METHODOLOGY

The methodology used in this project is presented in flowchart in figure 1. The input image is first preprocessed to enhance the image. Next, the image is passed to Optical Character Recognition software for character recognition. Finally, the recognized characters are extracted from the image.

2.1 System Overview

The project is divided into three sections – Preprocessing, Optical Character recognition and Character Extraction, as shown in the figure 1.

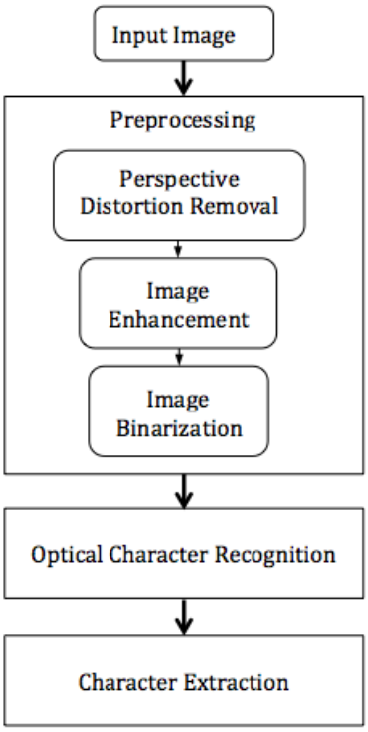


Figure 1. Flowchart of the system.

A. Preprocessing

The image captured by the application is first preprocessed using several image processing techniques to enhance the image for character recognition. Preprocessing is carried out before passing the image to the OCR for character extraction process. The preprocessing step consists of the following steps.

1). Perspective Distortion Removal

Perspective distortion removal is very important for the OCR to correctly recognize the characters. To remove the perspective distortion following steps are carried out. First, the edge of image is detected using canny edge detection algorithm [6]. The detected edge is dilated to make the edges appear stronger. Dilation is a morphological operation, which is used for filling in the gaps in the objects [7]. The dilation of set A by set B is defined as the set of all displacements, z, such that B and A overlap by at least one element. The dilation of A by B can be defined as:

$$A \oplus B = \left\{ z \middle| \left[\left(\hat{B} \right)_z \cap A \right] \subseteq A \right\} \qquad (1)$$

where, $\hat{B}$ represents the reflection of set B. Set B is referred to as the structuring element.

Next step is to find the corner of the image. For this purpose, Hough line transform [8] is used to find the straight lines in the image. The intersection point of the lines is computed. The polygon formed by four intersecting lines is checked for four vertices. Next, the corners of the polygon are determined as top-left, bottom-left, top-right and bottom-right corner. For this purpose, the center of the image is calculated. Points that have lower y-axis than the center point are determined as the top points and the points that are higher than center point are bottom points (assuming the origin lies in the top left corner). Between two top points, the point with higher x –axis value is top right corner and the point with lower x-axis value is top left corner. Similarly for bottom points, the point with higher x –axis value is bottom right corner and the point with lower x-axis value is bottom left corner. After computing the corner points in the image, the perspective transformation matrix is obtained from the corner points of the source image and the corner points of the destination image. Finally, the image is warped using the perspective transformation matrix, hence, removing the perspective distortion from the image.

2). Image Enhancement

In this step, the perspective rectified image is enhanced using piecewise linear transform such as contrast stretching technique. The main idea of contrast stretching is to increase the range of gray level values [9].

In next step, morphological operation - Black-hat transform is carried out for shading correction. The black-hat operation produces the image with the objects that are darker than the surrounding objects. Since the characters in the image are darker than the background, this operation is very desirable. Black-hat transform (h) is defined as follows [10]:

$$h = (f \bullet b) - f \qquad (2)$$

where, $(f \bullet b)$ represents closing of f by b, f is input image and b is the structuring element.

3). Image Binarization

Finally, Otsu's binarization technique is used to binarize the image. Binarization is obtained through thresholding. Thresholding is used to segment image from its background by assigning an intensity value called threshold value for each pixel in the image, such that each pixel is classified as pixel in background or the foreground. The Otsu's technique operates on the gray level bimodal histogram and automatically calculates a threshold value [11]. The binarized image then can be fed to the OCR.

B. Optical Character Recognition

For optical character recognition, an open source OCR engine - Tessaract [12] was used. Tessaract is developed by HP Labs and is maintained by Google Inc. Tessaract takes in binary image as an input and extracts the text from the input image. For the android application, tess-two library is used. Tess-two library is a part of Tessaract engine that can be used for Android development. The training files for tess-two, available online was used to train the OCR [13].

C. Character Extraction

After the text is recognized using OCR, the text should then be extracted. For this purpose, Java regular expression is used to recognize and extract only the most significant information from the image [14]. The android app extracts only the numeric text from the image. However, the app can be modified to extract alphanumeric text from the image.

3. RESULT AND DISCUSSION

3.1 Dataset and Application Development

The primary source of image for the application is the picture of a device known as YSI sonde. YSI Sonde is an instrument used in research fields to measure water quality parameters such as pH, temperature, salinity, and dissolved oxygen, etc. The app allows the user to take a picture of Sonde monitor and extract the information from the image. The user is also able to select an image from the phone's internal memory. The app was tested with at least 60 different images of YSI Sonde with different perspective angles and shading.

The application is developed in android environment. Eclipse IDE is used for development. Java native interface is used to call C++ and OpenCV functions from Java program. Training files for OCR is copied to a folder in the sd card of the Android phone. OpenCV and Tess-two library are also copied to a folder in the sd card of the phone. The application requires Android version 2.3 or higher. The preferred version of Android is 4.3. The application requires permissions to read and write to sd card and to use internal camera.

A user is supplied with two buttons on the app – one to capture an image using phone's camera and another to load an image from the sd card of the phone. Selecting the button to load image allows the user to open up gallery and choose the image from the gallery. Figure 2 shows the screen shots of the application.

3.2 Text Extraction from Image

The primary source of image is the picture of a device known as YSI sonde. Figure 3 shows a sample input image to the application.

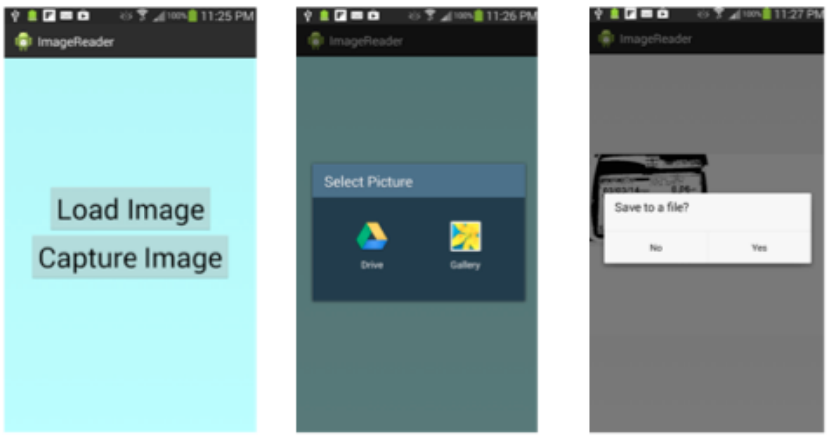


Figure 2. Screenshots of the application. a) home screen with two buttons, b) selecting image from gallery, c) saving the extracted information to a file.

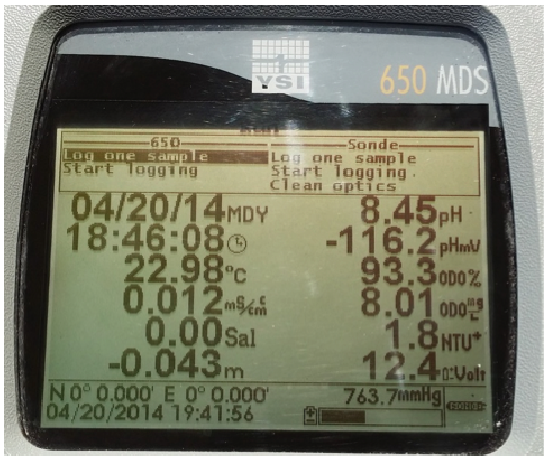
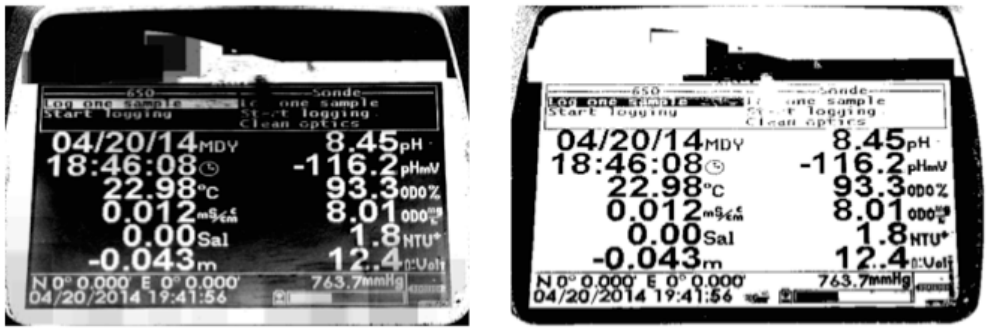


Figure 3. Sample input image of YSI Sonde.

The original image in figure 3 is preprocessed to detect and dialate edges as shown in figure 4. The figure 5 shows the computed four corners of the image as described in section 2, perspective distortion removal.

a) b)

Figure 4. Preprocessing of image. a) Edge detection and dialation, b) selecting four corners of the image.



a) b)

Figure 5. Preprocessing of image. a) bottom-hat operation with rectangular structuring element, b) binarization of image.

Figure 6 shows the screenshot of the information extracted from the image. The application is designed to extract the numeric data along with some special symbols such as '-', ':', '/', and '.'. The extracted information is stored on a file and the file is named as the current date and time so that it will be easier to retrieve later from the phone.

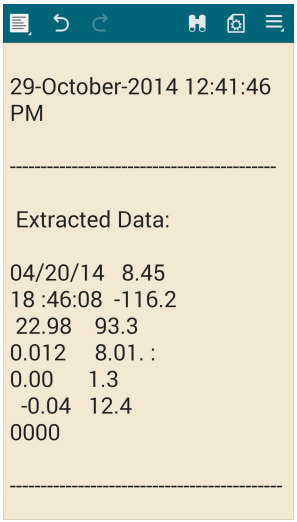


Figure 6. Screenshot of the information extraced from the image.

The app was successful to extract text from the image. The app was designed to only extract numeric data. However, alpha-numeric data can also be extracted by making a slight change. The perspective distortion removal phase sometimes produces erroneous results. However, if the corner computation is very accurate, the accuracy of the app also increases.

The algorithm was tested with 60 images. For each image, the total number of characters correctly recognized by the application was obtained. To measure the accuracy of the algorithm, the average of the total characters correctly recognized was computed. The accuracy of the algorithm thus, resulted to 91%. Since, the application was designed to extract only the numeric data and few special symbols such as '-', ':', '/', and '.', extra alphabets and other special characters were not take into account while measuring accuracy.

4. CONCLUSION

In this paper, a novel algorithm and a mobile application is developed for Android platform using Java Native interface to call C++ and OpenCv functions from Java program. The application allows a user to take a picture of the YSI Sonde monitor, extract text from the captured image and store it on a file. The application helps the user to get information quickly from the image and reduce human error while typing in the information to a sheet of paper. Also, since the extracted information is stored on a file, there is no fear of loosing the information, as opposed to writing down the information on a piece of paper. The algorithm was tested on 60 different images of YSI Sonde with different perspective features and shading. Experimental results, demonstrate the effectiveness of the proposed method. In future, a better algorithm for perspective removal can be used to improve the application.

REFERENCES

[1] Wai-Lin Chan and Chi-Man Pun, "Robust Character Recognition Using Connected-Component Extraction," *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on*, 310-313 (2011). □

[2] Ezaki, N., Kiyota, K., Minh, B.T., Bulacu, M., and Schomaker, L., "Improved text-detection methods for a camera-based text reading system for blind persons," *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on,* 257-261 (2005). □

[3] Lee, S., Seok, J., Min, K., and Kim, J., "Scene Text Extraction Using Image Intensity and Color Information," *Pattern Recognition, 2009,*1-5 (2009).

[4] Park, J., Lee, G., Lai, A. N., Kim, E., Lim, J., Kim, S., and Oh, S., “Automatic detection and recognition of shop name in outdoor signboard images”. In *Signal Processing and Information Technology, 2008. ISSPIT 2008. IEEE International Symposium on*, 111-116 (2008).

[5] Song, Y. J., Kim, K. C., Choi, Y. W., Byun, H. R., Kim, S. H., Chi, S. Y., and Chung, Y., “Text region extraction and text segmentation on camera-captured document style images”, In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, 172-176 (2005).

[6] Canny, J., “A computational approach to edge detection”, *in Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 679-698 (1986).

[7] Haralick, R.M.; Sternberg, Stanley R.; Zhuang, Xinhua, "Image Analysis Using Mathematical Morphology," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.PAMI-9, no.4, pp.532,550, July 1987

[8] Kiryati, N., Eldar, Y., and Bruckstein, A. M., “A probabilistic Hough transform”, *Pattern recognition*, *24*(4), 303-316, (1991).

[9] Al-amri, S. S., Kalyankar, N. V., & Khamitkar, S. D. (2010). Linear and non-linear contrast enhancement image. *Int. J. Comput. Sci. Network Security*, *10*(2), 139-143.

[10] Gonzalez, R. C., and Woods, R. E., & Eddins, S. L., [Digital image processing], Prentice Hall, 665-678 (2007).

[11] Otsu, N., “A threshold selection method from gray-level histograms”, *Automatica*, *11*(285-296), 23-27 (1975).

[12] Making a simple OCR Android app using Tessaract < http://gaut.am/making-an-ocr-android- app-using-tesseract/ □> (October, 2014).

[13] Tessaract OCR training files, < https://code.google.com/p/tesseract-ocr/downloads/list □> (May, 2014).

[14] Java Regex tutorial, < http://www.vogella.com/tutorials/JavaRegularExpressions/article.html □> (December 2014).