Extraction of Text on TV Screen using Optical Character Recognition

Ivan Kastelan*, Sandra Kukolj*, Vukota Pekovic**, Vladimir Marinkovic*, Zoran Marceta**
* University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
** RT-RK Institute for Computer Based Systems, Novi Sad, Serbia

ivan.kastelan@ieee.org, sandra.kukolj@rt-rk.com, vukota.pekovic@rt-rk.com, vladimir.marinkovic@rt-rk.com, zoran.marceta@rt-rk.com

Abstract—Optical character recognition (OCR) is a very active area of research and has become very successful in pattern recognition. It is based on algorithms for machine vision and artificial intelligence and used in developing algorithms for reading text on images, e.g. reading registration plates, scanned books and documents, etc. This paper presents the system for text extraction on the image taken by grabbing the content of the TV screen. The preparation steps for OCR are developed which detect the text regions in an image. An open-source algorithm for OCR is then run to read the text regions. The system is used as a part of Black Box Testing system in order to test and functionally verify TV set operation. After reading the text regions, comparison with the expected text is performed to make a final pass/fail decision for the test case. The system successfully read the text from the TV screen and can be used in a functional verification system.

1. INTRODUCTION

Optical character recognition (OCR) is a very active area of research and has become very successful in pattern recognition. OCR is mostly used in developing algorithms for reading text on the image taken by the camera, e.g. in reading registration plates, reading scanned books and documents, etc. It is based on algorithms for machine vision and artificial intelligence, i.e. neural networks, vector machines, fuzzy classifiers, etc.

Xie [1] developed an OCR system based on image preprocessing technologies combined with least square support vector machine (LS-SVM). It uses dynamic threshold operation and robust gray value normalization to segment characters and extract features respectively, and then uses LS-SVM to classify characters based on features. The system was evaluated by carrying out recognition experiments on the optical characters of electronic components. The author claims that the proposed method has a lot of potential for robust recognition learning.

Fuzzy classifiers are a promising method for OCR, but it is a challenging task due to the complexity of definition of the membership functions, even though the size of the alphabet is small for Roman alphabet. Fonseca, Rodrigues, Mora and Ribeiro [2] presented a solution for the semi-automatic design of a Fuzzy classifier for letters and digits to be applied on the automatic recognition of cars license plates on unstructured conditions. They used Classification and Regression Tree (CART) algorithm to learn the rules that regulate the design of the different characters and to generate fuzzy rules that implement the fuzzy classifiers in a completely automatic way. A fuzzy inference engine executes the rules to obtain the characters classification. Reference [2] proposes two layers of fuzzy classifiers: one to distinguish between letters and digits and the other to classify different letters or different digits.

One of the areas of application of OCR is in automatic number plate recognition. The objective of this field is to design the efficient and reliable system for vehicle recognition based on the registration number read on the vehicle’s number plate. Qadri and Asif [3] proposed one system for automatic number plate recognition which detects the vehicle, captures the vehicle image, extracts the vehicle number plate region using image segmentation and recognizes the characters on number plate using OCR. The recognized registration number is then compared with the database and the vehicle’s owner is discovered.

In the sensor field, Taylor, Singh, Brown, Bjarnason, Hanson and Gossard [4] applied the OCR in quantifying the image quality of imaging systems capturing the image of 8x10 mm copper letters on a fiberglass substrate. OCR results were used as metric for image quality of imaging systems at different frequencies – images where OCR performed better and managed to recognize characters were of the higher quality. They quantified image quality at different detection bands (100 GHz, 400 GHz and 600 GHz) and discovered that 100 GHz system did not work well for OCR, while 600 GHz system had a fivefold better OCR result than a 400 GHz system.

This paper proposes another area of application of OCR, namely in the extraction of text on the content of TV screens. The proposed text extraction system grabs the image representing the current TV screen content, prepares it for OCR and runs OCR to detect regions of text on the image and read the content. The proposed system is a part of the Black Box Testing (BBT) system [5]-[6] used for automated testing and functional verification of digital television sets. Text extraction is used to verify the functional operation of TV sets by, for example, reading the menu options presented on the screen in order to verify if the TV opened the correct menu when presented with a given set of remote control commands.

The rest of the paper is organized as follows: section 2 gives the overview of the proposed system and algorithm on which the proposed system is based. Section 3 explains the procedure in which the grabbed image is prepared for OCR. Section 4 gives experimental results of the proposed OCR method in extracting text regions on the TV screen. Finally, section 5 gives some concluding remarks and plans for future research.
II. SYSTEM AND ALGORITHM OVERVIEW

The proposed system for text extraction using OCR, as a part of a system for testing and functional verification of digital television sets, consists of:

- Object under test (TV set)
- Grabbing device
- Algorithm which analyzes the grabbed image, consisting of:
  - Preparation for OCR
  - OCR
- Final decision making process which:
  - Compares the extracted text with the expected content
  - Decides whether the test passes or fails.

The following are the steps in the text extraction algorithm:

- Edge detection
- Detection of candidate lines for text region boundaries
- Labeling of text segments
- Adaptive thresholding
- OCR run on the detected text regions

This paper presents the procedure implemented in order to detect the text regions, which represents the image preparation for OCR. After detecting the text regions, an open-source OCR algorithm is run on these regions to extract the text.

III. IMAGE PREPARATION FOR OCR

Before running the OCR algorithm, some preparation steps are performed in order to prepare the image for OCR. As shown in section 4, this step improves OCR performance because it enhances regions with text and suppresses other regions, allowing the OCR algorithm to better recognize the text written on the image.

A. Edge Detection

The first step in processing the image from the grabber is the edge detection. It is performed in two steps:

- Gaussian noise reduction
- Scharr edge detection

Given the image $A$, the noise is reduced using the Gaussian method [7], i.e. convolution (1).

$$A' = \frac{1}{159} \begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix}$$

The next step in the algorithm is the general edge detection using the Scharr operator [8]. This operator is said to have improvements over the widely-used Sobel operator. Given the image $A$, the intensity and the angle of the edges are calculated using (2)-(5).

$$G_x = A^* \begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}$$

$$G_y = A^* \begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \arctan \frac{G_y}{G_x}$$

After calculating the intensity and angle of the edges, threshold is applied on both values. Only edges with the high-enough intensity are kept for the future steps. The threshold used for edge detection is static and determined empirically, after several experimental trials.
B. Detection of Candidate Text Border Lines

Candidate text border lines are detected from the edge image, by calculating the histogram of horizontal and vertical lines. The candidate text regions are lines which are adjacent to the sufficient number of lines with a similar number of edge pixels. Therefore, in this step, only the lines satisfying these criteria are kept as candidates of text segments. This will be further explained with an example in section 4.

C. Labeling of Text Segments

After detecting the text border lines, their coordinates are stored for future use. The vertices of rectangles are kept to be the input to OCR showing it where to search for text and perform reading.

The area which does not belong to text segments is eliminated from the edge image, leaving only the edges which represent the text. Text segments are further labeled for better recognition by OCR method.

The labeling process is perfected in order to get rectangles containing text only. It is done by adding more lines in the neighborhood of the detected text. Those gaps are filled only if the current line in the observed segment is surrounded with text pixels positioned on the following sides: lower right, lower left, upper right and upper left.

D. Adaptive Thresholding

The threshold used to decide whether the horizontal or vertical line represents the text region or not is dynamically adapted during the algorithm operation. Adaptive thresholding is performed using the Sauvola thresholding [9].

In order to determine the threshold \( t(x,y) \) for the pixel \((x,y)\), the window of the area \( w \times w \) around the pixel is considered and its mean and standard deviation are calculated as \( m(x,y) \) and \( s(x,y) \) respectively. Let \( R \) be the maximum value of the standard deviation and \( k \) be a parameter, the threshold is computed using (6).

\[
t(x,y) = m(x,y) \left[ 1 + k \left( \frac{s(x,y)}{R} - 1 \right) \right] \quad (6)
\]

In the proposed algorithm, the value of the maximum standard deviation used is \( R = 128 \) (as suggested by Sauvola for grayscale images) and the value of the parameter used is \( k = 0.5 \).

E. OCR

Finally, the prepared image is fed to an open-source OCR method, Tesseract [10], which processes the image and reads the text on it. The text is then compared to the expected text, e.g. expected menu content. If the extracted text matches the expected one, the test passes, on the contrary it fails. This allows automatic testing of the functionality of TV sets and their response to remote control commands, e.g. whether they open the correct menu selections or not.

IV. EXPERIMENTAL RESULTS

In order to present the results of the algorithm, it was run on an image grabbed from TV, Fig. 2.

The results of edge detection are presented in Fig. 4. It can be seen that the edge detection resulted in detection of the text edges as well as other edges on the image. Further algorithm steps are needed, as explained in section 3, to select only the text edges. Horizontal and vertical lines from Fig. 4 are analyzed and their histograms are calculated and presented in Fig. 3.

![Figure 2. Grabbed image](image2.jpg)

![Figure 3. Histograms of horizontal lines (upper) and vertical lines (lower). Horizontal axis represents pixels and the vertical axis represents the number of pixels in each line](image3.jpg)
As seen from histograms in Fig. 3 and the actual image in Fig. 2, it is not possible to detect text regions simply by looking at lines having some number of pixels larger or smaller than some defined threshold. In order to represent text region, the line must be surrounded by sufficient number of lines with the similar number of edges, corresponding to the font size of the text. Therefore, looking at the horizontal histogram, the lines with the largest number of edge pixels can be discarded because they are surrounded by lines with no edge pixels. The actual text regions are, for example, the three line regions around pixels 400-500. Adaptive thresholding is used to select only these lines and reject lines which are “alone”, like the lines with the largest number of edge pixels representing the rectangle seen in the right image region.

Finally, Fig. 5 presents marked text regions in the grabbed image. It can be seen that the algorithm successfully eliminated most of the non-text regions, but some small regions around the map and logos remained detected. Further research needs to be done to improve the algorithm to remove these regions and leave only text regions.

V. CONCLUSIONS

This paper presented a system for text extraction based on the open-source OCR algorithm. Preparation steps for OCR were developed which detected text regions in the image, and OCR was run on detected regions to read the text. The system is used for functional verification of TV sets. Text on the grabbed image is read in order to verify whether the TV responds to remote control commands successfully or not, i.e. whether it opens the correct menu or not.

In the future research, algorithm will be further improved to detect only text regions and reduce detection of small non-text regions like some regions around the map and logos on Fig. 5.

ACKNOWLEDGMENT

This work was partially supported by the Ministry of Education and Science of the Republic of Serbia, under project No. 44009, year 2011.

REFERENCES


