DOI:10.5604/20830157.1109372

OPTICAL CHARACTER RECOGNITION USING ARTIFICIAL INTELLIGENCE TECHNOLOGIES

Adam Musiał, Piotr Szczepaniak
Lodz University of Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics

Abstract. *The article represents results of the research of an Optical Character Recognition system. Proposed OCR system is able to convert a raster image into the text string, which represents the text shown on the input image. The main innovation is the fact that the system was created without following any strict rules. It was more an innovative research rather than simple programming using ready guidelines.*

Keywords: character recognition, artificial intelligence, feature extraction, clustering algorithms

OPTYCZNE ROZPOZNAWANIE ZNAKÓW Z UŻYCIEM SZTUCZNEJ INTELIGENCJI

Streszczenie. *Celem projektu opisywanego w artykule było przygotowanie działającego systemu do optycznego rozpoznawania znaków, tj. zdolnego przekształcić rastrowy obraz wejściowy w łańcuch znaków odpowiadający zapisanemu tekstowi na obrazie. Nowością jest m.in. fakt wykonania tego systemu bez podążania za z góry znaną architekturą aplikacji, a przygotowanie go w sposób bardziej doświadczalny, czyli wykorzystując podejście nowatorskie.*

Słowa kluczowe: rozpoznawanie znaków, sztuczna inteligencja, ekstrakcja cech, algorytmy klastrowania

Introduction

The aim of this article is to describe one of the complex method of implementing a working Optical Character Recognition System. Specified system was the source project for the master's thesis [2] and was thoughtfully described. It uses some of the Artificial Intelligence Technologies, Fourier Transform, a lot of own graphic methods for image pre-processing purposes and even some heuristic approaches.

Another advantage of presenting such system is the fact of showing all components of a working system together on one diagram. Because of that, the presented article may be treated as an overview or an introduction to such systems. Because of the limited volume this article will be an overview rather than full specification. That may even help readers in creating similar working systems using abstract data processing tools. Those systems may be original and not a copy.

1. Approach

The main program was composed using many functional layers. Every layer contains different algorithms and is responsible for another processing stage.

Whole system may be described as a black box which receives True-Color bitmap image on the input and produces text string on the output (Figure 1).

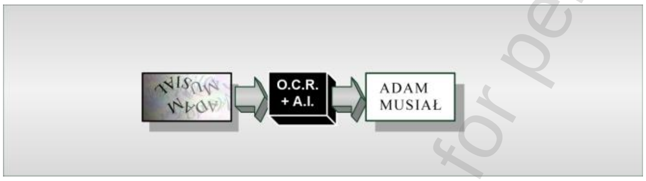

Fig. 1. OCR system in form of black-box model

Several different methods were analyzed. The important assumption was that the module should only analyze boundaries of the letters instead of analyzing their complete image. Another assumption was that the only serif font will be used and that only a subset of uppercase letters, numbers and some special symbols will be used. To make the program more useful recognition of Polish national letters was marked as the demanded feature. It was a quite interesting part because in Polish language a dot symbol may be a standalone symbol of period, may be a half of the colon symbol or even a part of Polish national letter "Ż".

At the beginning of the processing procedure the input image is filtered and preprocessed. Next step is to trace boundaries and to extract all features. Another layer is responsible for classifying shapes. After that, a lot of geometric algorithms and custom expert system is used to join letter shapes into letters. Recognized letters are merged into text lines and text lines are joined into whole page. To clarify the description, each module will be discussed in its own chapter. After that a short summary will be presented.

2. Image preprocessing

The first application layer in this project was the image pre-processing layer. That layer is responsible for converting True-Color bitmap image into an equivalent monochromatic image representation. There was the assumption that every letter contour should be accurately represented on the monochromatic image in form of black pixels. Every background pixel should be represented in form of white pixels. At that time, the system assumed that text is generally darker than the background.

There were two feature requirements. The first was that the program should be as noise-proof as possible. The second was that the program never knows what the source of the image is. During experimental tests two main sources of test images were used including scanned images.

First step was to maximize the contrast of image. To do that task a simple histogram analysis was used. Thanks to that even images with low contrast, with dark background or light text were optimized.

Next step was to convert the True-Color image with three color channels into grayscale. There were three versions of the conversion subroutine. First version was a simple equation with sum and division by the number of color channels. Rest two versions were using various equations of luminosity calculating purposes.

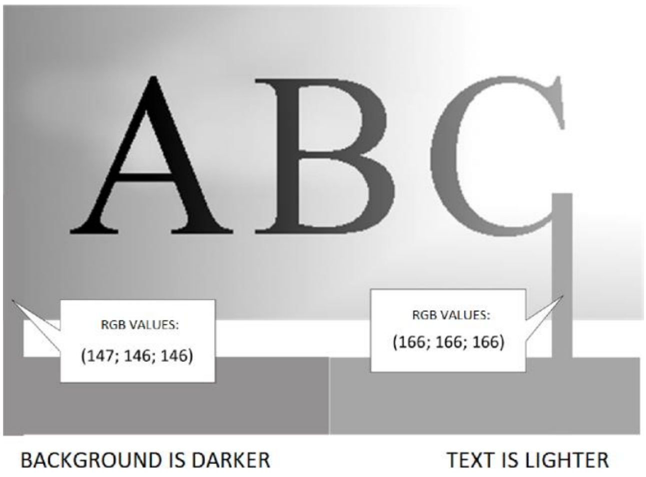

Fig. 2. Non trivial example image

The last step was to decompose the grayscale image into a black and white equivalent image. The simplest and the quickestmethod was to use simple binarisation technique. Such method may be enough for most high-quality input images. Another method is to use adaptive binarisation technique which searches for the optimal threshold level value.

Interesting example of non-trivial image is shown on the Figure 2. The main difference between usual trivial images is that some parts of text are lighter than some parts of the background. That thing is visible when analyzing the histogram because both classes overlaps themselves.

It is impossible to successfully preprocess that non-trivial image using plain nor the adaptive binarisation. The main problem is that the optimal value of the threshold simply does not exist (Figure 3).

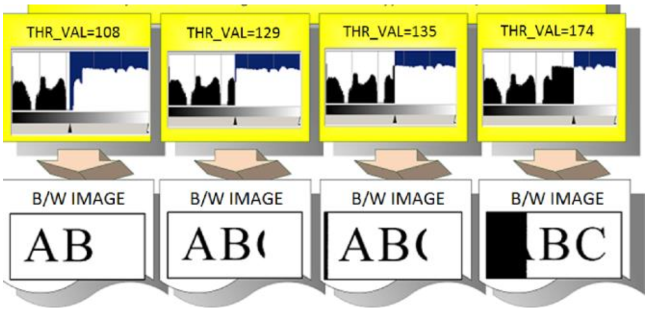

Fig. 3. Results of trying to process non-trivial image using optimized binarisation technique

To overcome that problem, own algorithm was developed. It combines custom histogram analysis and custom region growing method. It uses two indexes that are going together starting from the boundaries. Every time a small amount of the histogram values is processed. Usually the text index selects a part of the letter and the background index selects a part of the background. At this stage we assume that the text index does not mix text with the background, but on the other hand only a part of the text is selected. To complete the selection custom version of the region growing procedure is used. It selects all pixels with similar luminosity value. Usually the letter is represented by a strong pattern, so we may assume that the whole part of the letter will be selected even when only one pixel of that letter will be selected by the text index. After running some iterations of the proposed algorithm, whole image is successfully processed (Figure 4).

Very promising approach is to transform image into another color model that is less sensitive to noises and to cluster pixels with color-aware algorithm. Such approach may be very useful when trying to process colored text on colored background. Such approach was used in [1] on similar field of science with very good results.

3. Boundary tracing

At the beginning of the research, several edge filters were analyzed. Their main disadvantage was the possible ambiguity of represented shape. Even morphological operations were not enough to ensure that the boundary may be deterministically traced. Sometimes boundaries could not be closed and that was the signal of failure.

After many tests own deterministic algorithm was developed. It may be shortly described as a deterministic boundary tracer because it analyses every boundary as a path. In case of arriving the ambiguous cross it guesses the proper direction by following the preceding direction.

Another advantage is that it is possible to extract additional information about the shape. Proposed algorithm is able to determine if the traced boundary is the external or internal boundary. This is the essential information when trying to differentiate between for ex. period and internal parts of the letter. Moreover, it is possible to classify external shapes and internal shapes separately.

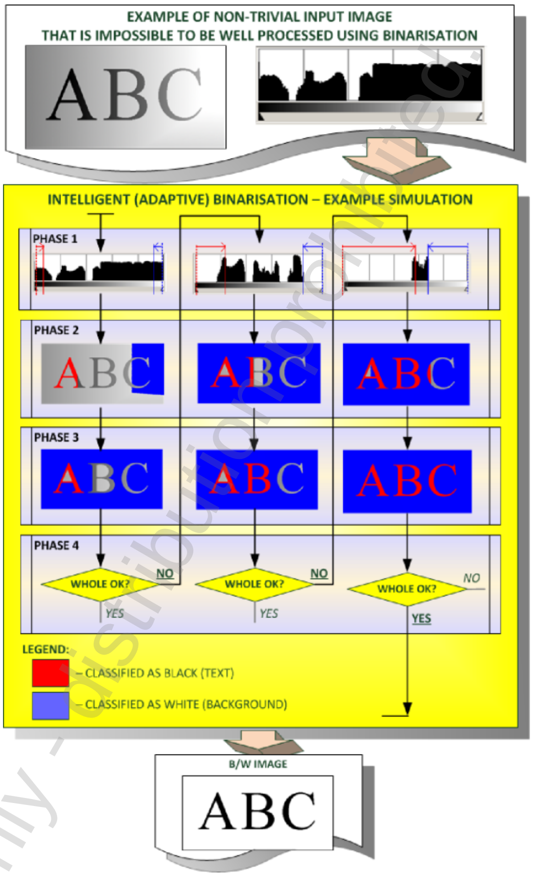
EXAMPLE OF NON-TRIVIAL INPUT IMAGE
THAT IS IMPOSSIBLE TO BE WELL PROCESSED USING BINARISATION
ABC
INTELLIGENT (ADAPTIVE) BINARISATION – EXAMPLE SIMULATION
PHASE 1
PHASE 2
ABC
ABC
ABC
PHASE 3
ABC
ABC
ABC
PHASE 4
WHOLE OK?
NO
YES
WHOLE OK?
NO
YES
WHOLE OK?
NO
YES
LEGEND:
🟥 – CLASSIFIED AS BLACK (TEXT)
🟦 – CLASSIFIED AS WHITE (BACKGROUND)
B/W IMAGE
ABC

Fig. 4. Results of trying to process non-trivial image using own method

4. Fourier descriptors and feature extraction

For the feature extraction purposes Fourier descriptors and some geometric algorithms were used.

After smooth normalization each shape is represented in the frequency domain by the Fourier descriptors. Usage of Fourier descriptors is helpful because they are invariant against translation, scale, rotation and their starting point.

Some example of such data is shown on Figure 5.

Each shape is also represented by some geometrical features that are computed and passed as the additional information together with the Fourier descriptors.

5. Classification modules

There is a set of two classifier modules, one for internal and one for external shapes. Every classifier module may be Feedforward Neural Network, SVM, k-NN, RBF or even another module that may be implemented.

Training set consists of letter images that are dynamically changed to add more abstract information to the classifier. That change includes rotation and scaling. Such changes may be helpful with the ability of differentiating between the shape itself and plain pixels raster.

6. Letter extraction module

After recognizing contours, another module was responsible for joining them into letters. The main problem was the ambiguity. Such ambiguity is defined by very similar or exact shapes that may be parts of different letters or even symbols. Plain example when using Polish national letters is the similarity of the “C” and the “Ć” letter which use common shape. Another example is the similarity of the period symbol and the “O” letter.

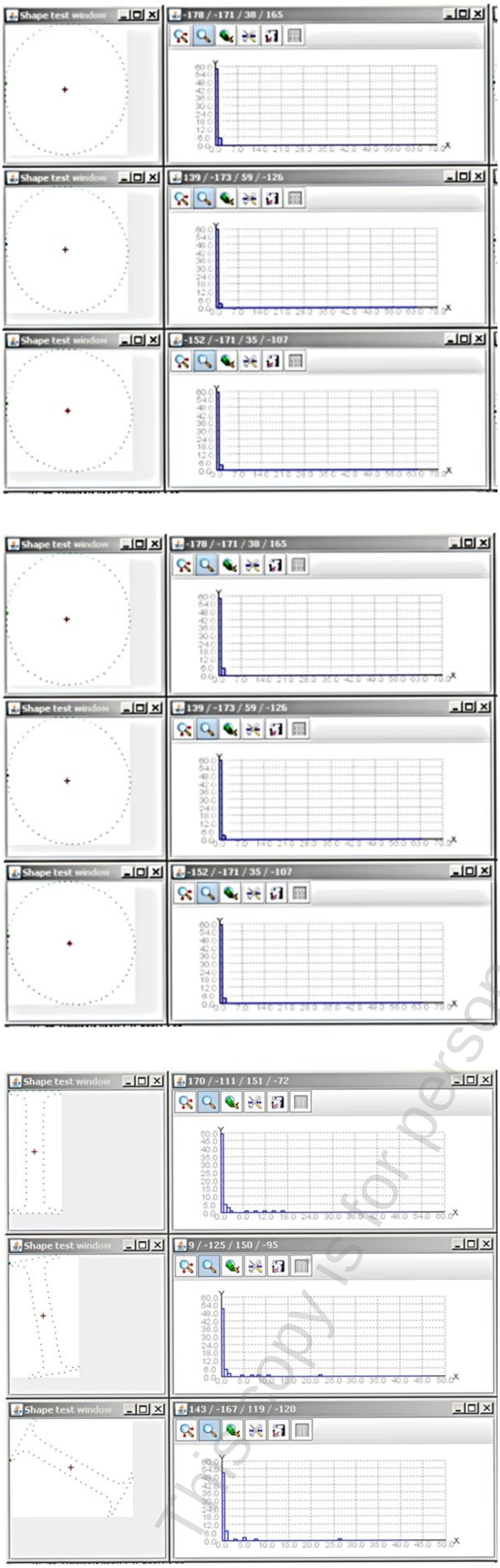

Fig. 5. Results of transforming shapes into Fourier Descriptors

Ambiguity problem is shown on an example Figure 6.

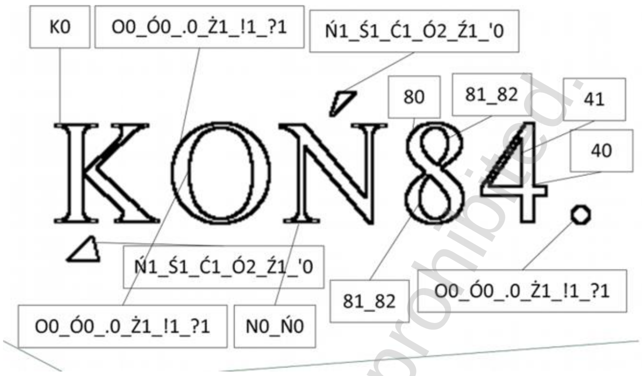

Fig. 6. Explanation of ambiguity of shapes

To overcome that problem, shapes are ordered into groups and processed using own expert system. The main cure is to extract more unambiguous shapes at the beginning of processing. Some geometric computations are helpful to determine which symbol may be treated as a part of a letter.

7. Concatenation modules

Recognizing letters does not finish the recognition procedure. Additional effort must be done to concatenate all letters into words and lines. That layer uses mainly some sorts of geometric equations to extract information about letter locations.

To make the algorithm more automatic, the k-NN method was used. That method clusters data into two or three sets: letters within one word, letters delimited with space, letters delimited with longer space or tabulator.

Similar algorithm was used to merge lines of text into whole page. Such algorithm also uses automatic functions to make the program more flexible and noise-proof.

8. Results

During the research a complete OCR system which uses Artificial Intelligence technologies was created. Another important achievements were for example own graphics algorithms for the image optimization purposes. A lot of time was spent on the test experiments. They were essential to retrieve the valid parameter values.

Interesting things are the example images with 100% recognition accuracy ratios shown on Figures 7, 8, 10, 11 and 12. Another nice feature is that some example images were recognized with 100% accuracy when using my system and in the same time they were recognized with the lower accuracy when using some of paid systems.

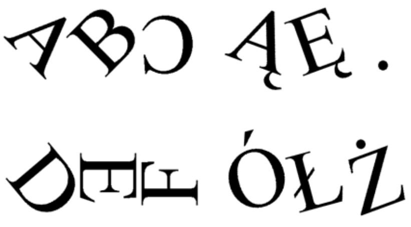

Fig. 7. Example image with 100% recognition accuracy ratio

Fig. 8. Example image with 100% recognition accuracy ratio

9. Future works

The cost of automation of the complex process is the additional time needed to find the optimal or suboptimal solution. On the other hand, the main advantage is that the presented system is able to recognize a lot of non-trivial images.

Future works may be done in terms of upgrading classifier modules with Genetic Algorithms to add the ability of self-optimizing. Another very promising approach is the fusion with other graphic methods used in [1] to improve the ability of recognizing full color texts. An example image that contains text and background with the same luminosity is presented on Figure 9. While classic approach will thread such image as simple solid fill, approach proposed in [1] would be able to perfectly segment that image into 2 classes.

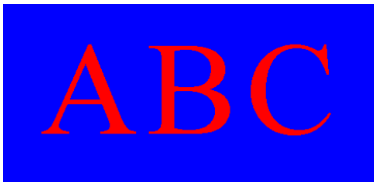

Fig. 9. Example of colorful image with the same luminosity of text and background

Of course the system could be able to recognize more characters including ligatures or even graphically concatenated ones. Unfortunately some of them may be time consuming.

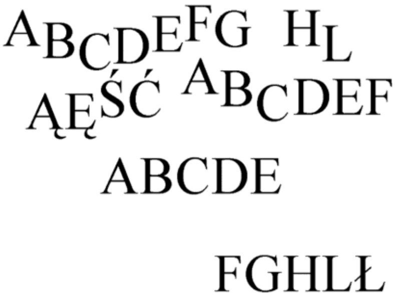

Fig. 10. Example image with 100% recognition accuracy ratio

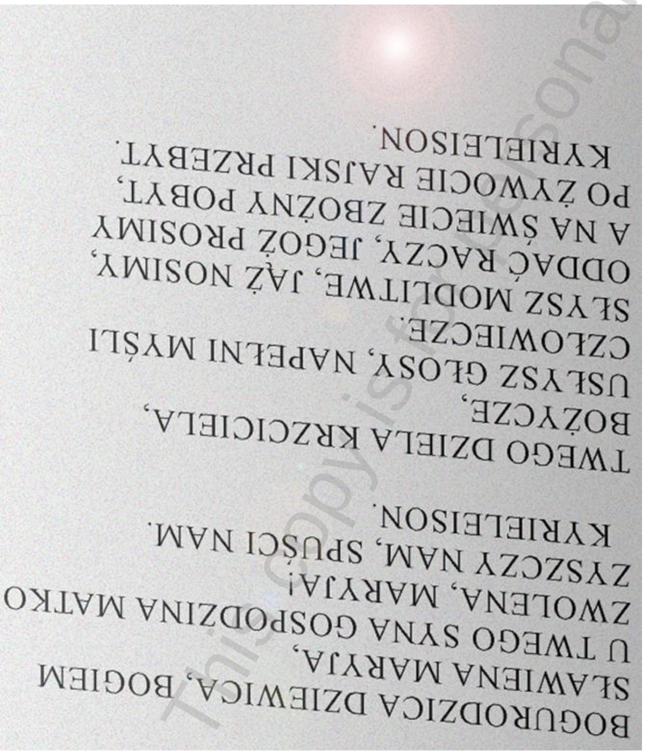
BOGURODZICA DZIEWICA, BOGIEM
SŁAWIENA MARYJA,
U TWEGO SYNA GOSPODZINA MATKO
ZWOLENA, MARYJA!
ZYSZCZY NAM, SPUŚCI NAM.
KYRIELEISON.
TWEGO DZIELA KRZCICIELA,
BOŻYCZE,
USŁYSZ GŁOSY, NAPEŁNI MYŚLI
CZŁOWIECZE.
SŁYSZ MODLITWĘ, JĄŻ NOSIMY,
ODDAĆ RACZY, JEGOŻ PROSIMY,
A NA ŚWIECIE ZBOŻNY POBYT,
PO ŻYWOCIE RAJSKI PRZEBYT.
KYRIELEISON.

Fig. 11. Example image with 100% recognition accuracy ratio

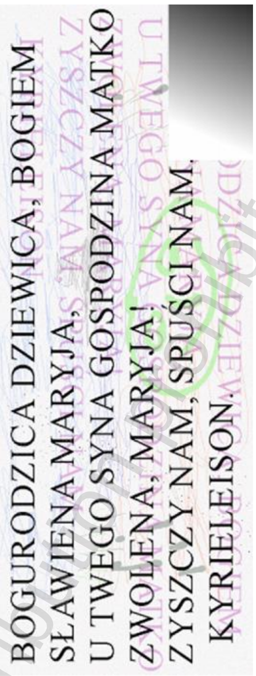

Fig. 12. Example image with 100% recognition accuracy ratio

10.Acknowledgements

Author would like to express his gratitude to Prof. Peter Szczepaniak, Ph.D. Eng. – H.M. Vice-Rector for University Development for proposing the very interesting field of expertise for the Master's Thesis creation purposes and for his very significant additional, invaluable help during researches. He would also thank to Kamil Stokfiszewski, Ph.D. Eng. and Darius Puchała, Ph.D. Eng. for his additional help with that project. He would also express his thanks to Jagoda Lazarek, M.Sc. Eng. for her help with proposing very promising improvement approaches.

References

[1] Lazarek J., Szczepaniak P.: Detection of Semantically Significant Image Elements Using Neural Networks. Computer Recognition Systems 4, Tom 4.

[2] Musiał A., Szczepaniak P.: Optical Character Recognition using Artificial Intelligence Technologies. Master's Thesis at the Institute of Information Technologies. Lodz University of Technology.

[3] Puchała D., Yatsymirskyy M.: Neural Network in Fast Adaptive Fourier Descriptor Based Leaves Classification. Artificial Intelligence and Soft Computing – ICAISC 2008.

[4] Szczepaniak P.: Obliczenia inteligentne, szybkie przekształcenia i klasyfikatory. Akademicka Oficyna Wydawnicza Exit, 2004.

M.Sc. Eng. Adam Musiał
e-mail: adam.musial.ftims@gmail.com

Ph.D. student at The Faculty of Technical Physics, Information Technology and Applied Mathematics at Lodz University of Technology. Presented information are mainly the results of his research for Master's Thesis purposes. There are some new theoretical improvements achieved during the PhD studies.

Prof. Piotr Szczepaniak

H.M. Vice-Rector for University Development at Lodz University of Technology. General manager of the The Faculty of Technical Physics, Information Technology and Applied Mathematics. Major advisor of the Master's Thesis which is the base source of this paper.

otrzymano/received: 01.10.2013 *przyjęto do druku/accepted: 30.03.2014*