20th International Conference
INFORMATION TECHNOLOGY INTERFACES

ITI '98

Pula, Croatia, June 16-19, 1998

WEIGHTED VOTING OF MULTIPLE ALN CLASSIFIERS FOR OCR

Dejan Đorđević, Dragan Mihajlov

Faculty of Electrical Engineering
Sts. Kiril & Metodij University

Correspondence to:

Dejan Đorđević, Faculty of Electrical Engineering, Sts. Kiril & Metodij University
Karpoš II b.b., PO Box 574, 91000 Skopje, Republic of Macedonia
phone: +389 91 36 35 66 fax: +389 91 36 42 62 E-mail:dejan@cerera.etf.ukim.edu.mk

WEIGHTED VOTING OF MULTIPLE ALN CLASSIFIERS FOR OCR

Dejan Đorđević, Dragan Mihajlov
Faculty of Electrical Engineering, Sts. Kiril & Metodij University
Karpoš II b.b., 91000 Skopje, Macedonia
dejan@cerera.etf.ukim.edu.mk, dragan@cerera.etf.ukim.edu.mk

Abstract: *This paper addresses the problem of combining the results of multiple adaptive logic networks that were used as classifiers in OCR system for Macedonian Cyrillic. Combining is performed by weighted voting of some or all the classifiers. The weights associated to every trained classifier are evaluated on a test sample. As a result of the weighted voting process, a list of character candidates along with their confidence levels is produced for every pattern. The confidence levels of the character candidates are used in the contextual postprocessing phase for word candidates generation and lexicon lookup.*

Keywords: OCR, ALN, weighted voting, classifier combining, character shape preclassification

1. Introduction

Despite the fact that many satisfactory results have been reported for the OCR problem, there is still a room for improvement especially in the cases of recognition of poorly printed and damaged documents. Recently, a promising direction was suggested that a number of classifiers can be used in parallel to tackle the difficulties. This paper addresses the problem of character recognition by combining the results of multiple adaptive logic networks (ALN) that were used as classifiers in OCR system for Macedonian Cyrillic. This system is part of a system for support of humans with damaged eyesight (Đorđević *et al.*, 1995), that includes Braille printing, archiving and speech synthesis.

The task of text recognition is to transform the image data containing textual information into its "usual" representation as strings over a given alphabet (e.g. ASCII). This is usually accomplished in three main operational steps: document layout analysis, character recognition, and contextual postprocessing. Document layout analysis consists of image filtering and thresholding, skew detection and correction, and segmentation. In most existing OCR systems, character recognition is performed on individual characters. Therefore, prior to character recognition it is necessary to isolate individual characters from the text image. The technique that partitions the image of lines and words in individual characters is called character segmentation.

The task of the character recognition is to assign the digitized image of the character into its symbolic class. A pattern recognition algorithm is used to extract the shape into the appropriate class. Number of different methods like: template matching, statistical, syntactical, and structural approaches exist for character recognition (Pavlidis, 1978). Artificial neural networks have emerged as the perfect choice for implementing classifiers for OCR (Bishop, 1995). Combining or integrating the outputs of multiple classifiers can also lead to substantial improvements in recognition.

It is evident that the performance of text recognition systems can be significantly improved by applying higher level knowledge in the form of contextual information. In the contextual postprocessing step a variety of different approaches exist, from simple valid bigram checkup to implementation of complex linguistic syntax knowledge.

2. Document segmentation

Approaches for segmenting document image components can be divided in top-down and bottom-up techniques. Top-down techniques divide the region into major regions, which are further divided into subregions, while bottom-up methods first extract small elements (typically connected components) and then combine them to form larger areas. There are also some hybrid methods that combine global and local strategies. Most of the segmentation and recognition algorithms are extremely sensitive on even a small skew, so the image is usually rotated to eliminate the skew prior to recognition. Number of techniques for text skew angle detection have been reported (Baird, 1987).

As we use the top-down approach for layout analysis and segmentation, which is generally very sensitive to skew, automatic skew elimination is performed first. The estimation of the skew angle of the scanned document is performed using Hough transform on a reduced image of the document. The reduction factor depends on the original image size and can be different in vertical and horizontal direction. The skew is eliminated by rotating the image around its center by the estimated angle.

The block segmentation is performed using an adaptation of the recursive X-Y cuts technique (Wang *et al.*, 1989). Text line and character segmentation is performed using the projection profiles (Casey *et al.*, 1996). After the segmentation, a preclassification of the segmented characters is performed.

3. Character preclassification

The task of the preclassification is to prune the set of classes to a smaller number. This yields to: better performance, distinguishing capital letters for the characters that have similar shapes for lowercase and uppercase letters, and simplifying the recognition engine.

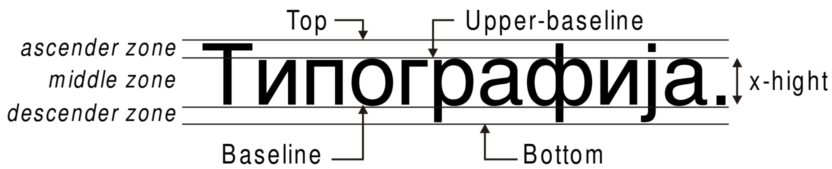

Fig. 1: Text line parameter positions: Top, Upper-baseline, Baseline and Bottom

Four horizontal lines define the boundaries of three significant zones on each text line (Fig. 1). Most of the characters in the Latin and Cyrillic alphabet can be readily mapped to a small number of classes according to their position relative to the baselines and their aspect ratio (Đorđević *et al.*, 1996). According to the vertical spread of the character boxes through the text line zones only, characters can be grouped in five basic classes. Taking in consideration the aspect ratio of the character boxes and their size relative to the x-height, these classes can be further divided.

In our approach, we use 12 character shape classes (Table 1). Baseline detection is performed on a text line level prior to character segmentation using the first difference of the horizontal projection. Baselines' positions are then corrected on isolated word level using the bounding boxes of the isolated characters. The character preclassification process is implemented as a decision tree (Đorđević, 1997). Since all character preclassification is based on character box vertical location, these techniques tend to be largely font independent (Spitz, 1995).

Table 1: Classifying the Macedonian Cyrillic characters in 12 shape classes

Class	Characters
𝔸	АБВГЃЕЖЗЅИЈКЛЉМНЊОПРСТ ЌУФХЧШ бќѓ éй 023456789 %?
𝕩	авгежзѕиклљмнњопстхчш +
𝕪	д р у ц џ
ℚ	Д Ц Џ Ф
𝕛	j ( )

Class	Characters
!	! 1 /
▫	.
,	,
`	' "
▭	– =
:	:
;	;

4. Character recognition

Character recognition includes classification of the digitized image of an isolated character into one of the symbolic classes that usually represent a symbol from the alphabet. In our approach adaptive logic networks (ALN) are used to classify the digitized image of an isolated character into one of the 62 symbolic classes.

Table 2: The dimensions of the destination matrix for scaling according to the shape class of the character

Shape class	Height	Width
𝔸 𝕩 ▭	16	16
ℚ 𝕪	24	16
𝕛 !	24	8
▫ , `	8	8
: ;	16	8

The binary matrix holding the image of the character is first scaled to fixed dimensions. These dimensions depend on the shape class of the isolated character and are chosen to roughly match the aspect ratio of the characters from the corresponding shape class (Table 2).

This scaled binary matrix is used as input to the ALNs. An adaptive logic network is a completely binary network organized as a binary tree (Armstrong *et al.*, 1975). It accepts binary input on the leave nodes of the tree and produces a single binary output at the root. According to this, an ALN can classify the pattern in only one of the two possible classes. It is obvious that the ALN is hard decision classifier, which can only judge whether the pattern is member of its symbolic class, or not. On the other hand, we would rather like to have some kind of measure that will represent fuzzy membership of the corresponding pattern to every one of the possible symbolic classes. These measures can then be used by the contextual postprocessor that will make the final decision about the pattern's membership exploiting the context information.

In our approach, we use multiple ALNs, each of which is trained to recognize single symbolic class, i.e. to accept the pattern of its symbolic class and to reject the patterns of all other symbolic classes. Every character pattern is forwarded to all ALNs that correspond to the pattern's shape class. Each ALN judges whether the pattern is member of its symbolic class or not, and outputs a single binary value. In our approach, several ALN trees (called voters) are provided for every class, and the number of trees per class is not necessarily the same (Đorđević, 1997). Every ALN is associated a weight vector that represents the errors each ALN is making while classifying patterns of different symbolic classes. The individual decisions of all polled trees are then combined using weighted voting which yields to a list of character candidates along with their levels of confidence (Fig. 2).

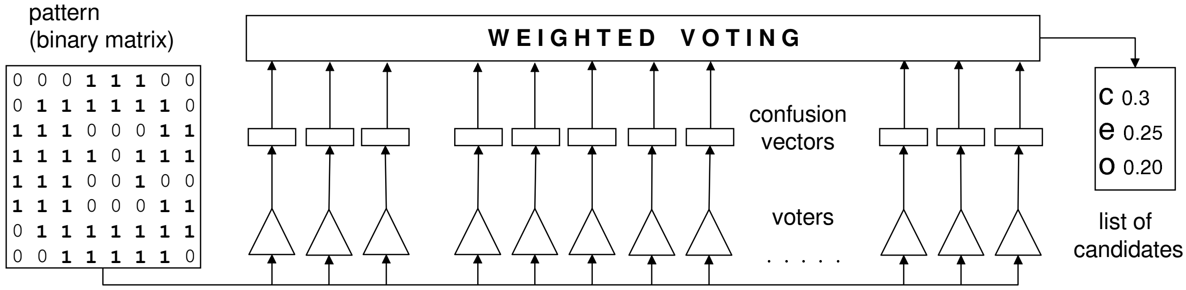

Fig. 2: Character recognition using weighted voting

Let denote with $e_i^c(x)$ the output of the i-th voter responsible for recognition of the symbolic class K_c when the pattern x is feed to its input. If the number of symbolic classes that can be recognized is N_c, and the number of voters responsible for recognition of the symbolic class K_c is N_v^c, polling them all will produce a set of events $e_i^c(x) = \{0,1\}$, which can be represented as binary vector. However, we would like to produce a real vector $\mathbf{T} = [t_1 \quad t_2 \quad \dots \quad t_{N_c}]$ which will represent the measures $t_i = M(x \in K_i)$ that the corresponding pattern is member of the symbolic class K_i. We define these measures as:

$$M(x \in K_c) = \begin{cases} \dfrac{\sum_{i=1}^{N_v^c} e_i^c(x)}{\sum_{s=1}^{N_c}\sum_{i=1}^{N_v^s} e_i^s(x)} \cdot \dfrac{\sum_{i=1}^{N_v^c} e_i^c(x)}{N_v^c}, & \text{if } \sum_{s=1}^{N_c}\sum_{i=1}^{N_v^s} e_i^s(x) \neq 0 \\[2ex] 0, & \text{if } \sum_{s=1}^{N_c}\sum_{i=1}^{N_v^s} e_i^s(x) = 0 \end{cases}, \quad c = 1, \dots, N_c \qquad (1)$$

Each measure combines the classification results considering the number of votes given by the voters for the appropriate class, the total number of votes the pattern received from all the voters, and the number of voters responsible for recognizing the appropriate class. This measure represents the belief that the pattern is a member of a given symbolic class. However, (1) does not take in consideration the reliability of the classifiers, i.e. the mistakes every classifier is making while classifying patterns from different classes. The reliability of one classifier's decisions can be expressed by its confusion vector. The confusion vector carries information about the substitution errors every classifier is making, i.e. patterns of which symbolic classes it recognizes as its owns and how often. The confusion vector for every classifier can be obtained by polling it with a test pattern set. Every element ${}^j v_i^c = n_j$ of the confusion vector $\boldsymbol{v}_i^c = [n_1 \;\; n_2 \;\; \cdots \;\; n_j \;\; \cdots \;\; n_{N_c} \;\; n_\phi]$, represents the number of patterns from the symbolic class K_j that were recognized as members of the class K_c by the classifier e_i^c. ϕ represents the patterns that are supposed to be rejected by all the classifiers (for example missegmented characters).

Using the confusion vector a weight vector $\boldsymbol{w}_i^c$ can be produced (Đorđević, 1997) which represents the reliability of the corresponding ALN. Every single classifier e_i^c is associated a weight vector $\boldsymbol{w}_i^c$ that contains confidence levels representing the likeliness of the decisions the classifier makes. Every element ${}^j w_i^c = \boldsymbol{w}_i^c[j]$ of the weight vector represents the likeliness that the pattern belongs to class K_j, if it has been recognized by the classifier e_i^c. The values in this vector can also be regarded as prior knowledge of an expert who upon receipt of the evidence – the occurrence of the event $e_i^c(x)$, expresses his beliefs with uncertainty on each of N_c mutually exclusive propositions $x \in K_i$, $i=1,..,N_c$, by a real number called confidence level.

Considering the weights, the measure $M(x \in K_c)$ that the corresponding pattern is member of the symbolic class K_c can be expressed as:

$$M(x \in K_c) = \begin{cases} \dfrac{\sum_{s=1}^{N_c} \sum_{i=1}^{N_v^c} e_i^c(x) \cdot {}^c w_i^s}{\sum_{k=1}^{N_c} \sum_{s=1}^{N_c} \sum_{i=1}^{N_v^s} e_i^s(x) \cdot {}^k w_i^s} \cdot \dfrac{\sum_{s=1}^{N_c} \sum_{i=1}^{N_v^c} e_i^c(x) \cdot {}^c w_i^s}{\sum_{s=1}^{N_c} \sum_{i=1}^{N_v^s} {}^c w_i^s}, & \text{if } \sum_{s=1}^{N_c} \sum_{i=1}^{N_v^s} e_i^s(x) \neq 0 \\ 0, & \text{if } \sum_{s=1}^{N_c} \sum_{i=1}^{N_v^s} e_i^s(x) = 0 \end{cases} \tag{2}$$

where:

- N_c is the total number of symbolic classes;
- N_v^c is the number of voters for the symbolic class K_c;
- $e_i^c(x)$ the output of the i-th voter responsible for recognition of the symbolic class K_c when the pattern x is feed to its input;
- ${}^c w_i^s$ is the c-th element of the weight vector associated to the i-th voter responsible for recognition of the symbolic class K_s, representing the belief that the pattern is a member of the symbolic class K_c if recognized by this voter.

During the recognition process the pattern is forwarded to all the ALNs responsible for recognition of characters form the symbolic classes of the pattern's shape class. The outputs of all polled ALNs are then combined using (2), and the measures $M(x \in K_i)$ representing the belief that the corresponding pattern is member of the shape class K_i are normalized and sorted in descending order producing a list of character candidates for the pattern. The first few candidates from this list along with their levels of confidence are then forwarded to the contextual postprocessor.

5. Contextual postprocessing

The object of the contextual postprocessing is to detect and possibly correct part of the recognition mistakes, as well as to try to "fill up the blanks" i.e. to guess the unrecognized characters according to their context. In our approach the contextual postprocessing is incorporated on a word level using a lexicon organized as a reTRIEval tree.

The extended information supplied by the classifiers is of crucial importance in the contextual postprocessing step. The contextual postprocessor is being forwarded the first few most confident candidates from the isolated character classifier along with their levels of confidence. For every character position in the word several character candidates may be supplied (Đorđević *et al.*, 1998). The validity of the word candidate is examined by checking the lexicon for its existence. The word candidates are generated and checked one at a time, in sequence of decreasing word confidence, and the first candidate found in the lexicon is accepted.

If after examining all possible word candidates obtained by combining the character candidates on all character positions, a valid word was still not found, it means that for some character position in the word the right character is not among the character candidates, or the incomplete lexicon is missing this word. In such cases we can proceed in several ways: supplying more candidates for every character position to the contextual postprocessor, rejecting some of the unique candidates with low confidence levels and treat them as unrecognized or ignoring the lexicon and adopting the character classifier's best choices as valid recognition.

6. Conclusion

In this paper an optical character recognition system for Macedonian Cyrillic is presented. A special font independent character shape preclassification was described. Multiple adaptive logic networks were used as character classifiers. Each ALN is associated a weight vector representing the likeliness of the decisions the classifier makes. The pattern is forwarded to all the ALNs responsible for recognition of characters from the pattern's shape class. The outputs of all polled ALNs are combined by weighted voting producing list of character candidates along with their levels of confidence. This list is forwarded to the contextual postprocessor that incorporates context knowledge on an isolated word level using a lexicon for the Macedonian language to make the final recognition decision.

This weighted voting schema allows transparent addition of new voters without affecting the present ones. Adding new trained ALNs for the classes with lower performance can improve the recognition of the characters from the corresponding classes resulting in total performance increase. Incorporating some kind of syntactic knowledge above word level (word collocations for e.g.) can further increase the performance of the contextual postprocessing.

7. References

1. Armstrong, W. and Godbout G. (1975), "Properties of Binary Trees of Flexible Elements Useful in Pattern Recognition", *Proc. IEEE Int. Conf. Cybernetics & Society*, San Francisco, pp. 447-449.
2. Baird, H. S., 1987, "The skew angle of printed documents", *Proc. SPSE 40th Conf. Symp. Hybrid Imaging Systems*, Rochester, NY, pp. 21-24.
3. Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Clarendon Press, London.
4. Casey, R. G. and Lecolinet, E., (1996), "A Survey of Methods and Strategies in Character Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 7, pp. 690-706.
5. Đorđević D., Mihajlov, D., and Josifovski, Lj. (1995), "Computer System for Support of Humans With Damaged Sight: Subsystem for Optical Character Recognition of Printed Cyrillic Text", YU Info'95, Brezovica, *Zbornik radova YU INFO '95*, Knjiga 3, pp. 240-243.
6. Đorđević D., Josifovski, Lj., and Mihajlov, D. (1996), "Character Shape Preclassification in Mixed Script OCR for Macedonian Language", *Proceedings of the 18th International Conference on Information Technology Interfaces*, ITI'96, Pula, pp. 55-60.
7. Đorđević D. (1997), "System for Optical Character Recognition of Macedonian Cyrillic Texts Aimed for Helping Visually Impaired Persons", master thesis, Skopje, October, 1997.
8. Đorđević D., Mihajlov D., and Josifovski Lj. (1998), "Word Candidate Generation in Cyrillic OCR Based on ALN Classifiers", accepted for presentation on "9th Mediterranean Electrotehnical Conference - melecon '98", Tel-Aviv, Israel, 18-20 May, 1998.
9. Pavlidis, T. (1978), "A Review of Algorithms for Shape Analysis", *Computer Graphics and Image Processing*, Vol. 7, pp. 243-258.
10. Spitz, A. L. (1995), "An OCR Based on Character Shape Codes and Lexical Information", *Proc. 3rd Int. Conf. Document Analysis and Recognition*, Montreal, pp. 723-728.
11. Wang, D. and Srihari, S. (1989), "Classification of Newspaper Image Blocks Using Texture Analysis", *Computer Vision, Graphics and Image Processing*, Vol. 47, pp. 327-352.