Fonts Recognition by Using Typographic Features of Connected Components

Milen Dimitrov[1] and Antoaneta Popova[2]

Abstract: **Many different methods and systems of printed text recognition (OCR) are developed till now. These methods can be divided in two types. The mono/ single-font methods recognize characters from one, preliminary defined font only. Although first developed these methods give better results than the second type multi-fonts text recognition methods. The purpose of the multi-fonts text recognition is to recognize symbols with random/ unspecified shape, slope and size. Unfortunately the final results of these systems are not enough good. The presented paper describes a method for preliminary optical font recognition (a-priori OFR system) for giving this information on the next step to the recognition system. The purpose is to increase the multi-font text recognition accuracy. As recognition features are used text line typographic attributes extracted from the connected components and their bounding rectangles.**

A two layers neural network with back propagation is used as a classifier. The experiments with a font set of different typeface, weight, slope and size were carried out and described.

Keywords: **Optical font recognition, typographic features, text recognition**

I. INTRODUCTION

The hard-copy/paper based documents volume continues to increase with high speed although the soft copy/electronic documents usage, which is opposite to the expectation. The main reason is the human comfortable feeling when he reads and acquires of the paper documents.

From other side the soft copy/electronic documents have serious advantages in their storage, retrieval, restoration and updating. As results, the paper documents converting into their electronic versions, named print documents recognition (OCR) becomes very popular in the last 10-15 years. Analogical to the popularity of the newspapers, magazines, even after the radio discovery, the paper documents, which exist from hundred years, will continue to play an important role in our live.

The text recognition systems can be divided into three groups [2]:

- mono-font – Algorithms for a text processing only with a single font. Today the mono-font OCR systems reach a very high accuracy– more than 99%.
- multi-font – These systems work with a set of fonts, and often correspond to defined practical requirements.
- omni-font – These systems are independent on the font.

The entering of a great number new fonts in the computer industry makes very difficult the database supporting of base models for the multi-font OCR systems. This is the reason the omni-font OCR systems to be preferable. Unfortunately these systems give not so good results. The applying of a-priory algorithm for the optical font recognition (OFR) can increase significantly the accuracy. These methods are named a-priori.

A-posteriori methods for OFR are performed over a text with a known content (after OCR for instance). They are used for concrete font identification, as example for a completely electronic restoration of the original document.

The purpose of this paper is to research, develop and test an algorithm and a program model for the a-priory fonts' identification.

In a paragraph II is presented a brief analysis of some existing researches on the topic. A paragraph III describes the main definitions used in the developed algorithm. A paragraph IV presents the applied method and a paragraph V- the obtained results. At the end a paragraph VI contents the results analysis and describes the possible improvements.

II. EXISTING APPROACHES ANALYSIS

On the background of the huge number OCR publications the researches connected to the Optical Font Recognition (OFR) are discourage number. Ones of the most popular researches were published by Zramdini and Ingold [1], [3], [6]. They present a depth research of a completely font recognition system without the preliminary content knowledge (ApOFIS). The system extracts 8 careful chosen global typographic text features, which are passed to a Bayesian classifier. A created base fonts' model in advance is used with a priory known 280 fonts. In a literature [1] are presented and analyzed the efficiency of the features, as well as the text line length influence, and in [6] is done an evaluation of the recognition accuracy dependency on the input images quality decreasing.

A similar approach is used in [4]. The typographic features are extracted from a normalized image 9x9, and basically emphases on the edge symbols lines (serif). But this approach depends on the used languages, because of taking in account the specific symbols as g/g, a/a. As a classifier a back propagation neural network (BPNN) is applied.

Opposite to the methods using features, in [2] is shown an algorithm with a comparison to etalons by a nearest neighbor classifier, using single-side tangential distance.

[1] Milen Dimitrov is a Project Manager in Komero Technologies Int.- Sofia, Bulgaria
[2] Antoaneta Popova is an Associate Professor in the Technical University- Sofia, Bulgaria

In [7] an interesting algorithm is suggested for a dominated text determination with small key words (stop-words) vocabulary applying.

III. Definitions

The font is described with the following properties [5], [3]:

- family (typeface) - a descriptive name
- size - describing in typographic points (pt)
- weight - normal, light, bold
- slope - roman, italic

The typeface fonts are divided into 2 groups – serif (with small lines on the symbols' edges) and sanserif (without such lines).

A big variety of typeface fonts exist. While some of them are easy distinguished even by a beginner, others are very similar and difficult for identification even by an expert. But for the text recognition (OCR) improvement is more important to determine a correct size, weight and slope in the stage of OFR and it is not so important to find the exact typeface.

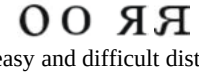

Figure 1. Example for easy and difficult distinguished symbols from two font typefaces (Times New Roman и Journal)

Although a single document contents a dominated font, the font is not a property on the document level. Theoretically the font is a property on a symbol level, but in practice it is a feature of the word as a part of the text line. Furthermore could be considered, that in one text line (sentence) exists only one typeface and font size, but the fonts are different mainly in the slope and/or weight.

The typographic structure of a single text line is defined as:

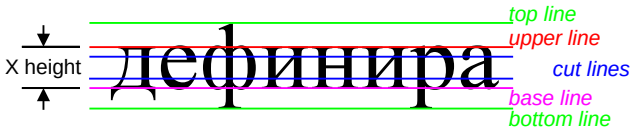

Figure 2. Typographic structure of a text line

The dominated symbols in the line are the small letters (lower case). In consequence of the fact that they can have or not ascenders or descenders the text line consists by three vertical zones (Figure 2): upper, central and bottom. Four dividing lines form these zones: top, upper, base and bottom lines. The height of the central zone is named X-height.

The experiments in the present paper are done on the text line/row, which can contain one or more words.

IV. Used Method Description

The main stages in the suggested method of the fonts' recognition are:

- Typography determination – detection of typographic zones and dividing lines
- Features extraction
- Learning
- Classification

A. Typographic line analysis

Most often the typographic zones founding is performed on the base of the vertical image profile [1], [3], [8]. However many times, the correct local maximum detection is impossible because of a lightly or noisy profile distribution.

Figure 3. Profile with an impossible correct detection of lines

In this paper are suggested the applying of the connected components approach. Because of this first the connected components (CC) in the binary images are detected: using 4 or 8 connected neighborhood. The coordinates of the bounding rectangles are defined too.

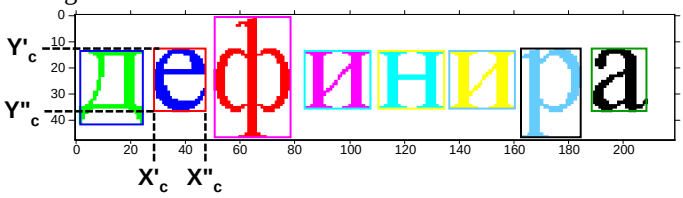

Figure 4. Connected components and the bounding rectangles

The distribution (histogram) determination of the Y coordinates for the bounding rectangles is the next step. This distribution appears as a profile P_y^V similar to the image profile. Two profiles are calculated - for the upper and the bottom zones. The upper and bottom edges $Y_c^{'}/Y_c^{"}$ of the bounding rectangles are used:

$$P_y^{VU} = \sum_{x=1}^{W} Y_c^{'},\ P_y^{VD} = \sum_{x=1}^{W} Y_c^{"},\ \ P_y^V = P_y^{VU} + P_y^{VD} \qquad (1)$$

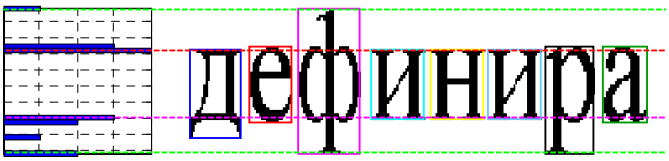

Figure 5. Profile of the symbol bounding rectangles

The text lines are analyzed on base of the found profile. All local maximums are detected in the profile - y is a local maximum if:

$$P_{y-1}^V < P_y^V \ and \ P_{y+1}^V < P_y^V \qquad (2)$$

During the local maximums detection the profile values less than a defined minimum value are not used.

The lines defining X-height appear global maximums corresponding in P_{VU} and P_{VB}:

Upper line - $L_u = \arg\max_y P_y^{VU}$

Base line - $L_b = \arg\max_y P_y^{VD}$

X-height is determined by:

$$H_X = L_b - L_u \quad (3)$$

If this line distance H_X is more than the threshold $H_{X\min}$ the given text line is accepted for an analyzing and opposite if is less than the threshold the text line is rejected:

$$H_X > H_{X\min} \quad (4)$$

The threshold is determined on the base of the supposed minimal font's size S_{min}:

$$H_{X\min} = 0.5\frac{S_{\min}}{72.27}R \quad (5)$$

Typically S_{min} = 8 pt. R is e the scanning resolution. In the below researches are applied *R=300 dpi*.

The rest local maximums are used for a founding the others dividing lines:

Top line - L_t - within the borders of $y = 0 \div L_u - 1$

Bottom line - L_d - within the borders of $y = L_b + 1 \div H$

B. Features extraction

One of the main text line features is the X-height determined by (3).

Typically the upper and bottom zones' heights are approximately 40-60% of the X-height. Because of this if the corresponding local maximum is outside of this zone are assumed that this dividing line and its corresponded zone do not exist.

The goal of the next step is to decrease the influence of the non-character components (".", "," and etc.) and the small objects (noise appears during the scanning). The components are filtered referring to the criteria:

a) The area A_c of the connected components to be bigger than a given threshold T_p:

$$A_c = \sum_x^{W_c}\sum_y^{H_c} B_{xy}^c > T_P \quad T_P = 0.2H_X^2, \quad (6)$$

where B_{xy}^c are pixels of the connected component/object *c*.

b) The components' sizes W_c and H_c to be in the defined boundaries:

$$\begin{aligned} T_{W\max} \ge W_c \ge T_{W\min} \quad W_c = X_c^{''} - X_c^{'} + 1 \\ T_{H\max} \ge H_c \ge T_{H\min} \quad H_c = Y_c^{''} - Y_c^{'} + 1 \end{aligned} \quad (7)$$

The concrete thresholds depend on the possible symbols in the script, and for the Latin and Cyrillic they are in the ranges:

$$\begin{aligned} T_{W\max} = 1.8 \div 2.0H_X \quad T_{W\min} = 0.05 \div 0.15H_X \\ T_{H\max} = 1.8 \div 2.2H_X \quad T_{H\min} = 0.7 \div 0.8H_X \end{aligned} \quad (8)$$

If the rejected connected components exceed 40% the text line is eliminated from the analyzing. The normalized connected components area is the percentage content of black/object elements in one connected component:

$$A_c^{'} = \frac{A_c}{W_c H_c} \quad (9)$$

The averaged value of this area A_v is used as a next feature, playing an important role in the font weight determination:

$$A_v = \sum_{c=1}^{C} A_c^{'}, \quad (10)$$

where C is the connected components' number after their filtration.

The next feature is the averaged value of the components bounding rectangles widths:

$$W_v = \sum_{c=1}^{C} W_c \,. \quad (11)$$

As a next feature is used the transition width between the background-object-background R_k on the cutting lines (black runs). Two cutting lines instead one central are used with a purpose to avoid the strokes in the symbol's center (e, g, к, ж and etc.):

$$L_T^{'} = L_u + 0.25H_X \quad L_T^{''} = L_b - 0.25H_X \quad (12)$$

The average value of the transitions' widths R_v (not bigger than given threshold) for the both lines is the next selected feature:

$$R_v = \frac{1}{K_{L_T}}\sum_k^{K_{L_T}} R_k \;\; for \;\; R_k < 0.4W_v \quad (13)$$

where K_{L_T} is the number of all transitions for both cut lines.

On the base of the horizontal profile P_x^H and its first derivative $P_x^{H'}$:

$$P_x^H = \sum_y^H B_{xy} \quad P_x^{H'} = P_{x+1}^H - P_x^H \quad x = 1 \div W - 1 \quad (14)$$

The profile density P_v is calculated:

$$P_v = \frac{1}{W_S}\sum_x^{W-1} P_x^{H'} \; if \; P_x^H > 0, \quad (15)$$

where W_S is the total width of CC, excluding overlaying.

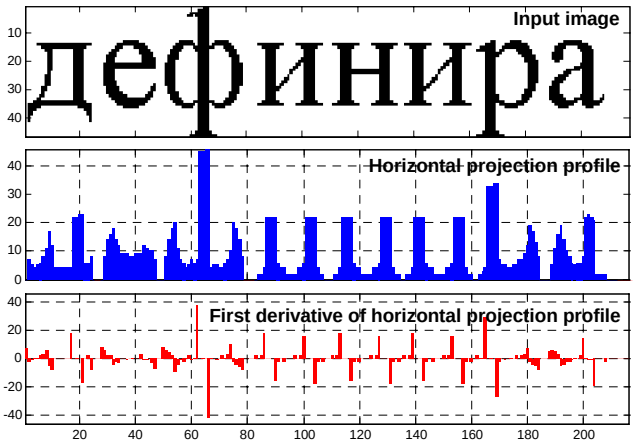

Figure 6. Horizontal profile and its first derivative

Only the parts of the profile, where the connected components (CC) are projected, are taken into account. The

purpose is to avoid the empty spaces between letters and words. For a better distinguish between serif and sanserif typeface fonts the contour elements (CE) of the connected components are used. CE is determined by applying of 8 masks in the 3x3 pixels region (white – background, colored – object, gray – value does not matter):

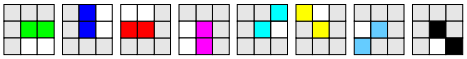

Figure 7. Contour elements masks

For a influence decreasing of the text content only CE are used, which are positioned in the upper and bottom edge of CC:

$$B_{xy} \ for\ y \in \left[Y_c^{'}, Y_c^{'} + 0.2H_c\right] \cup \left[Y_c^{''}, Y_c^{''} - 0.2H_c\right] \quad (16)$$

For each mask m is obtained a CC number e_c^m, which is normalized with the CC area:

$$E_c^m = e_c^m / A_c .$$

An average value is used as a final feature:

$$E_v^m = \frac{1}{C}\sum_{c=1}^{C} E_c^m,\ m = 1 \div M = 8 \quad (17)$$

C. Neural Network classification

A completely connected back propagation neural network is applied with 14 inputs, 700 neurons in the hidden layer and 144 in the output layer. Each output neuron corresponds to a given font and its target value is binary (0/1).

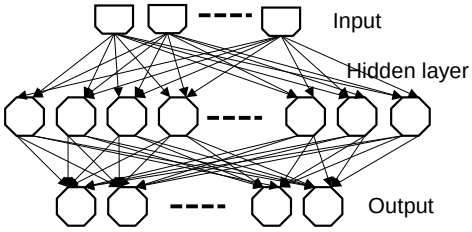

Figure 8. A back propagation neural network classifier structure

V. Experimental Results

A set of 12 typeface fonts is used, 3 sizes (10, 12, 14 pt) and 4 combinations of weight/slope (Normal, Italic, Bold, BoldItalic) - total 138 fonts (AGRevueCyr doesn't have Bold and BoldItalic). Automatically were generated 50 text lines for each font (total 7200 images of text lines). The width of text lines is arbitrary (from 2 to 9 words/15 to 60 characters).

The neural network was learned with all 7200 input images entered on the occasionally order with 1000 epochs/iterations.

The classification is tested with 10 text lines for each font (1380). The test images are obtained with scanning of real documents (hard copies).

The achieved classification rates is given in the below table.

TABLE1.
THE FONTS CLASSIFICATION RATE EXPERIMENTAL RESULTS

Size	10 pt				12 pt				14 pt			
Typeface	N	B	I	BI	N	B	I	BI	N	B	I	BI
AGRevueCyr	98.63	-	97.77	-	99.80	-	99.54	-	99.94	-	99.84	-
Arial	96.24	96.99	95.98	96.07	97.62	97.91	96.89	97.33	98.57	98.92	96.84	97.27
Balkan	97.89	98.01	95.68	96.93	98.92	98.73	97.60	97.89	98.57	98.92	97.74	98.37
CourierNew	97.70	97.96	97.08	97.45	98.77	98.92	97.58	97.76	98.99	98.87	97.71	97.66
Hebar	97.30	97.69	97.46	97.80	98.65	99.09	97.86	97.59	99.00	99.24	97.99	97.90
Kudriashov	97.50	98.45	97.23	97.99	98.49	98.68	96.14	97.33	98.14	98.52	96.90	96.85
Lazurski	97.81	97.90	98.02	97.97	97.68	97.98	96.30	97.62	97.88	98.04	97.55	97.86
LozenCondensed	96.48	96.69	95.00	95.99	96.23	96.64	95.07	97.41	98.28	98.67	98.06	98.22
Maritsa	97.13	97.52	97.09	97.63	98.48	98.92	97.69	97.68	98.83	98.48	95.78	97.73
Peterburg	96.43	97.26	96.31	96.08	97.77	97.93	96.53	97.12	98.27	98.74	97.00	97.28
Times New Roman	96.24	96.99	95.98	96.07	97.62	97.91	96.89	97.33	98.57	98.92	96.84	97.27

VI. Conclusion

In this paper we have presented a font identification model, which uses the Neural Network classifier, capable to perform a-priori OFR. The training of the system was carried out using dominated lowercase letters, and including also uppercase letters and digits of the considered fonts. The recognition rate for characters reaches 95.75- 100%. The work has proved that the suggested 14 topographic features are suitable for the font identification. The parameters like size, slope and weight are recognized with very high accuracy. The font's family parameter remains with no so high classification rate. The font families which are differ from others (like AGRevueCyr) are easy for recognition.

Some improvements can be done in future: determination of the discrimination possibilities of the features; evaluation of the text line length influence (number of symbols); taking in account the text lines with dominated capital letters. The next step will be to implement "OFR+OCR" module aimed at document characterization or classification.

References

[1] A. Zramdini, R. Ingold, "Optical Font Recognition Using Typographical Features", *IEEE Trans. On PAMI*, Vol. 20, N:8, 1998.

[2] S. La Manna, A. M. Colla, A. Sperduti, "Optical Font Recognition for Multi-Font OCR and Document Processing".

[3] A. Zramdini, R. Ingold, "Optical Font Recognition from Projection Profiles", *Electronic Publishing*, Vol. 6, 1993.

[4] M. C. Jung. Y. C.Shin, S. N. Srihari, "Multi-font Classification using Typographical Attributes". http://www.math.unipd.it

[5] S. Ozturk, B. Sankur, "Font Clustering and Cluster Identification in Document Images".

[6] A. Zramdini, R. Ingold. "A Study of Document Image Degradation Effects on Font Recognition", 3-th Int. Conf. of Document Analysis and Recognition, Montreal, Canada, ICDAR 1995.

[7] T. K. Ho. "Fast Identification of Stop Words for Font Learning and Keyword Spotting", Bell Laboratories.

[8] D. Dordevic, L. Josifovski, D. Mihajlov. "Character Shape Preclassification in Mixed Script OCR for Macedonian Language", 18-th Int. Conf. Information Technology Interfaces ITI, Pula, Croatia, 1996.