Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp114-117)

A Robust Font Recognition Using Invariant Moments

AVILÉS-CRUZ CARLOS, VILLEGAS-CORTEZ JUAN AND OCAMPO-HIDALGO J.
Departamento de Electronica. Universidad Autónoma Metropolitana – Azcapotzalco.
Av. San Pablo No. 180 Col. Reynosa. C.P. 02200, México D.F.

Abstract: -A robust Font Recognition (OFR) is proposed in this work; this is based on the analysis of texture characteristics of document text using invariant moments (invariant to scale, rotation and translation). There is not need of explicit local analysis in our method since the central moments features are extracted as a global characteristic from each font. A printed text block with a unique font is suitable to provide the specific texture properties necessary for the process of recognition. The used fonts were: Courier, Arial, Bookman Old Style, Franklin Gothic Medium, Comic Sans, Impact, Modern and Times New Roman; and their respective styles: regular, italic, bold, italic with bold. The invariant moment technique is used in this study to extract the font characteristics by window size estimation; from an entry text set a data base was build for the learning stage, and then standard statistical classifiers were applied for the identification stage (combining Gaussian and KNN classifiers). We found that the invariant moments change significantly when the textures are rotated and scaled as digital images; good recognition rate was obtained in font recognition with noise.

Key-Words: Optical Font Recognition (OFR), Optical Character Recognition (OCR), Classifiers.

1 Introduction

Font recognition has a great technological importance as it identifies different types of letter or typography. In addition, it improves the results achieved with the Optical Character Recognition (OCR) and plays an important role in the difficult task of Automated Document Processing (ADP). Different approaches have been proposed in order to accomplish it. The first of them was based in the local analysis of texts [2], [5], [6], [7], [9]. Methods based on the recognition of global textures have been implemented very recently [1], [8]. It is under the framework of the latter that we have proposed an alternative font recognition method which implements the technique of second and third-order invariant moments over a text block formed of a unique-font. A text-block printed with only one font has specific texture properties. These are used in this work to identify the fonts through the calculus of the central moments.

The methodology consisted of the next stages.

1) A set of 512 texts with different textures was generated from the combination of eight of the most employed fonts (as lines above) with four different styles (as lines above) at four different scales (6, 8, 10, and 12 points) and four rotation levels (0°, 45°, 90°, 135°; the rotation was carried out from the preprocessed text-block using a bicubic interpolation method). Every generated text was preprocessed in two aspects, eliminating blank spaces between lines, characters and words; and applying text-padding to all the lines so that each line had the same length. The output of this stage was 512 uniformed text-blocks with a unique texture.

2) 100 test-windows were taken at random for every generated text-block; the size of the windows was fixed to four sizes (64×64, 128×128, 256×256, and 512×512 pixels). Then, a data base was created through the calculus of the invariant moments of every test window. The calculated invariant moments of each text-block represented the class of the studied textures.

3) The created data base served to test our classifier based on the Gaussian and KNN techniques. This classifier was also tested with other samples taken from electronic documents.

4) The classifier was validated performing three techniques on the data base: resubstitution, cross-validation and leave-one-out.

See figure 1 for an illustration of the above stages.

The rest of the paper is organized as follows. Section 2 details the preprocessing. Section 3 and 4 describe the processes of font feature extraction and classification, respectively. Experiments and results are discussed in Section 5. Finally conclusions and perspectives are presented in Section 6.

2 Preprocessing

The analyzed text was contained in a BMP format file (the BMP format is converted to gray level image) in order to be used for the learning and the identification stages. The text could include space characters or spaces between words, letters or lines. Characters may have had different sizes between words or lines. The font information was extracted going through the words in order to obtain a

Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp114-117)

uniform block of text. The same procedure was applied to the spaces between lines in order to eliminate them. The preprocessing stage was divided in the four stages described in the following subsections.

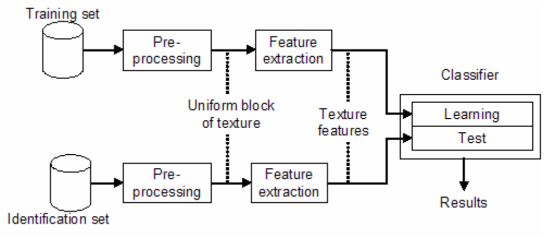

Fig. 1. Scheme of font identification system.

2.1. Locating Text Lines

The location of a text line was determined calculating the horizontal projection profile (HPP) of the whole text, which was determined by adding over each line the intensity of the pixels that belonged to it (gray levels). The values were then normalized with respect to the maximum value. The valleys between peaks corresponded to blank spaces between text lines. The distance between two valleys gave the high of each text line. As a result, it was possible to locate and determine the height of each text line.

2.2. Text Line Normalization

Since a text can contain several types of fonts and different sizes of them it is necessary to normalize letters and words of different sizes to a standard one. Once a text line was located, fonts were normalized to have all of the same size. However, it is worth mentioning that small font deformations could lead to small mistakes when the normalization stage was applied, for example an original letter “i” could appear as “l” or “1”.

2.3. Text Padding

Since the text may not be justified, refilling of blank spaces was performed when the text did not end with the rest of the lines. The chosen option consisted on copying parts of the text of the preceding (or another) line, so there is a bigger probability that words be of the same type when they are close to each other.

3 Features Extraction

The uniform text obtained in the previous section could be analyzed with any texture technique. As pointed out in the introduction section, there are only two previous papers using “global texture analysis” [1], [8]. We propose the use of the second and third order moments as they have a very good performance with non-structured random textures [1]. We wanted to report a methodology based on moment invariants technique for the optical font recognition problem, and also to compare results previously obtained [1], [8]. The next subsections describe the second and third order invariant moments techniques (details may be found in [4]).

3.1. Invariant Moments

Statistical moments represent average values of processes (powered to order *n*) when a random variable is involved. Here, the original and preprocessed images were considered as two dimensional arrays of a random variable of dimension $N \times N$. The random variables took values from level 0 to 255, as the images were considered in gray levels quantized in 8 bits (gray levels were obtained from BMP format). The images were obtained from word processors or from electronic documents.

Moments were calculated for the random variable *X*, which was identified with the image block. In addition, *X* is a matrix of two coordinates *(x,y)*, obtained from the image matrix *f(x,y)*. The definition of *(p+q) order invariant moment* around the origin is given by:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dxdy \qquad (1)$$

For an image, equation (1) could be expressed as (2), as *(p+q)* is the order of the central invariants moments:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\,dxdy \qquad (2)$$

where *(x, y)* are the pixel coordinates, and $\bar{x}$, $\bar{y}$ are the average values. The third order *central moments* are eight estimated and no redundant *(p+q)* combinations $(\mu_{00}, \mu_{20}, \mu_{02}, \mu_{11}, \mu_{30}, \mu_{12}, \mu_{21}, \mu_{03})$.

The normalized central moments, η_{pq} are given by:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \text{where} \quad \gamma = \frac{p+q}{2}+1 \qquad (3)$$

for $p+q = 2,3,\ldots$

From the second and third moments we get a set of seven invariant moments [4]:

$$
\begin{aligned}
&\phi_1 = \eta_{20} + \eta_{02} \\
&\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
&\phi_3 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
&\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
&\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\
&\quad -3\left(\eta_{21} + \eta_{03}\right)^2] + \left(3\eta_{21} - \eta_{03}\right)\left(\eta_{21} + \eta_{03}\right) \\
&\quad \left[3\left(\eta_{30} + \eta_{12}\right)^2 - \left(\eta_{21} + \eta_{03}\right)^2\right] \\
&\phi_6 = \left(\eta_{20} - \eta_{02}\right)\left[\left(\eta_{30} + 3\eta_{12}\right)^2 - \left(\eta_{21} + \eta_{03}\right)^2\right] \\
&\quad + 4\eta_{11}\left(\eta_{30} + \eta_{12}\right)\left(\eta_{21} + \eta_{03}\right) \\
&\phi_7 = \left(\eta_{21} - \eta_{30}\right)\left(\eta_{30} + \eta_{12}\right)\left[\left(\eta_{30} + \eta_{12}\right)^2 - 3\left(\eta_{21} + \eta_{03}\right)^2\right] \\
&\quad + \left(3\eta_{12} - \eta_{30}\right)\left(\eta_{21} + \eta_{03}\right)\left[3\left(\eta_{30} + \eta_{12}\right)^2 - \left(\eta_{21} + \eta_{03}\right)^2\right] \quad (4)
\end{aligned}
$$

According to [4] this set of moments are invariant for translation, rotation or scaling, thus the computing of (4) for an image under transformations ensures that its moments will no change significantly. We will see later that these changes result from both the nature of the digital images and their images resulting after the transformation process.

3.2. Noise Sensitivity

We also evaluated the impact of noise on the images. Random noise could arise from (a) the scanning process, (b) the scanning of low contrasted photocopies or from low degradation. Pepper noise was investigated in reference [8], but the performance of the employed method was not very good. Noise power is given by:

$$P_N = \frac{\text{Image Power}}{10^{(\text{SNR}/10)}} \quad (5)$$

where P_N is the power of noise and SNR is the signal-noise-rate. Once the signal to noise relationship was defined (in order to test over different SNR on image simulating level noise coming from the scanning process or photocopy process [1]), the image power was calculated with:

$$\text{Image Power} = \frac{1}{N \times M} \sum_{x=1}^{N} \sum_{y=1}^{M} f\left(x, y\right)^2 \quad (6)$$

where N and M are the width and height dimensions of the image respectively.

4 Font Classification

The algorithm applied for the processing of images consisted in the estimation of features not over complete images but on regions of them called “sub-images” or “test-windows”. The estimated attribute arrays of each sub-image consisted of the reference database used to recognize the type of each font (100 test-windows were taken randomly from each full text-block). A Bayes classifier was chosen to fulfill the latter task. As it belongs to the supervised type, two stages were required: (a) a learning stage and (b) a classification stage. The parameters required for it where the mean value and the variance-covariance matrix of features. A set of random windows was estimated for each text-block, this leaded to a matrix composed of 800 vectors of seven components for the set of fonts, as our learning database.

5 Results

Several tests were performed to validate our method. As described above, a set of 8 fonts, with 4 styles (as lines above) was used. A total of 32 combinations of styles and fonts were then considered. These fonts are from the Microsoft Word Processor. The images considered in this work were digitally converted by the word processor and then converted to images at 300 DPI. This way, is possible to control all the fonts and their sizes accurately. The texts were generated at four scales. The final text block of every text was sized in a variable manner according to the number of lines and scale. Finding the right size of each test-window was essential in order to make a good estimation, thus we try with five window sizes. Figure 4 shows the estimated errors. It can be seen from it that the bigger the window the lower the error of the validation methods. The validation methods employed were confusion matrix based (Resubstitution, Cross-Validation [3], Leave-One-Out). Thus considering an error level under 5%, the window of size 512×512 was then chosen.

As Fig. 1 shows, through the Resubstitution method we obtained the classification idealized results, due the same database is used for learning and classification stage, while the other methods take unknown values for the learning stage. Table 1 shows one confusion matrix for regular font. It can be observed that most fonts are of 100% identified at 512×512 size window.

5.1. Noise Sensitivity Results

The previous results were obtained without random noise. Nevertheless, it is important to consider noise because images are usually contaminated. The noise tests were made taking noise levels of 5, 11, and 17%. Fig. 1 shows results for three window sizes. As previous experiments, the bigger the window size, the better the recognition.

6 Conclusions and Future Work

Three main results were obtained with the methodology employed. First, the number of performed operations was lower with respect to other studies [8], [1]. Second, as opposed to what the theory predicts [4], we found that the invariant

Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp114-117)

moments change significantly when the textures are rotated and scaled as digital images. And third, the introduction of random noise over the samples yielded good levels of classification. As an average, we reached levels of classification as good as 95% with test-windows of 512×512 DPI, as it can be appreciated in Figure 4.

The findings of this work stimulate us to make test over transformed texts (scaled and rotated) such as photographs, which have the advantage of avoiding the noise emanating from the calculus of the transformations. Also, the power of the classifiers could be tested evaluating documents with mechanical and pepper noise such as those resulting from mechanical scanning. Our classifier could also be tested with Fonts associated to other languages such as Chinese, Japanese, Arabic, etc.). Finally, the invariability of the calculus of the moments can be studied with lower rotations (5° or 10°) in order to find a better identification of the font descriptors for every language.

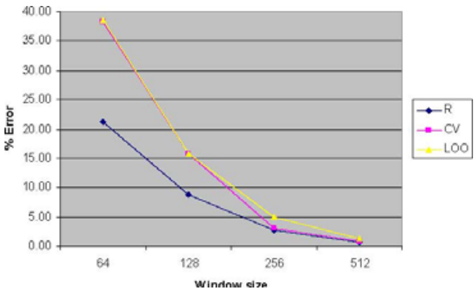

Fig. 4. Error convergence as function of test-window size. R=Resubstitution, CV=Cross-Validation, LOO=Leave One Out.

	Courier	Arial	Bookman Old Style	Franklin Gothic Medium	Comic Sans	Impact	Modern	Times New Roman
Courier	100	0	0	0	0	0	0	0
Arial	0	100	0	0	0	0	0	0
Bookman Old Style	0	0	100	0	0	0	0	0
Franklin Gothic Medium	0	0	0	100	0	0	0	0
Comic Sans	0	0	0	0	100	0	0	0
Impact	0	0	0	0	0	100	0	0
Modern	0	0	0	0	0	0	98	4
Times New Roman	0	0	0	0	0	0	2	96

Table 1. Confusion matrix for eight fonts in regular style, using cross-validation method (50%) and a Bayes classifier.

References

1. Aviles-Cruz Carlos, Rangel-Kuoppa Risto et al, “High-order statistical texture analysis – Font recognition applied”, Pattern Recognition Letters, v.26, pp. 135-145. 2005.
2. Cooperman R. “Producing Good Font Attribute Determination Using Error-Prone Information”. In Proceedings of the SPIE – Document Recognition IV, pages 50-57, 1997.
3. Fakunaga Keinosuke, “Introduction to Statistical Pattern Recognition”, Edit. Academic Press Inc. 1990.
4. González Rafael, Woods Richard, “Tratamiento Digital de Imágenes”, Edit. Addison-Wesley / Diaz de Santos. 1996.
5. Hongwei Shi, Theo Pavlidis. “Font Recognition and Contextual Processing for More Accurate Text Recognition”, *Proceedings of the 4th International Conference on Document Analysis and Recognition*, p.39-44, August 18-20, 1997.
6. Malik, J. Belongie, S., Shi, J., Leung, T. “Textons, Contours and Regions: Cue Integration in Image Segmentation”. IEEE Int. Conference on Computer Vision, CORFO, Greece, September 1999.
7. Schreyer, P. Suds, and C. Maderlechner, "Font Style Detection in Documents Using Textons", Proc. Third Document Analysis Systems Workshop. Assoc. for Pattern Recognition Int'l, 1998.
8. Yong Zhu, “Font Recognition Based on Global Texture Analysis”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 23, no. 10, pp. 1192-1200, Oct. 2001.
9. Zramdini A., Ingold R., “Optical Font Recognition Using Typographical Features”, IEEE Transactions on Pattern Analysis and Machine Intelligence, v.20 n.8, p.877-882, August 1998.