Efficient Off-Line Cursive Handwriting Word Recognition

B. Gatos, I. Pratikakis, A.L. Kesidis, S.J. Perantonis

Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,
National Center for Scientific Research "Demokritos",
GR-153 10 Agia Paraskevi, Athens, Greece
http://www.iit.demokritos.gr/cil, {bgat,ipratika,akesidis,sper}@iit.demokritos.gr

Abstract

In this paper, we present an off-line cursive word handwriting recognition methodology. This is based on a novel combination of two different modes of word image normalization and robust hybrid feature extraction. Word image normalization is performed by using as a reference point the geometric center of the word image as well as by placing the baseline of the word in the center of a rectangular box. Additionally, image pre-processing is performed in order to correct word skew, word slant as well as to normalize the stroke thickness. At a next step, two types of features are combined in a hybrid fashion. The first one, divides the word image into a set of zones and calculates the density of the character pixels in each zone. In the second type of features, we calculate the area that is formed from the projections of the upper and lower profile of the word. The performance of the proposed methodology is demonstrated after testing with the reference IAM cursive handwriting database.

Keywords: Handwriting word recognition, Hybrid feature extraction

1. Introduction

Off-line cursive handwriting recognition has achieved a great attention for many years due to its important contribution in the digital libraries evolution.

In the literature, two general approaches can be identified: the segmentation approach and the global or segmentation-free approach. The segmentation approach requires that each word has to be segmented into characters while the global approach entails the recognition of the whole word. In the segmentation approach, the crucial step is to split a scanned bitmap image of a document into individual characters [6].

A segmentation-free approach is followed in [1][3][8][11][13][14][15][19] where line and word segmentation is used for creating an index based on word matching. In [15], a discussion on different approaches to word matching is given. In [1], Ulam's distance is used for image matching by identifying the smallest number of mutations between two strings. In [3], a two-dimensional image is converted into a one-dimensional string. The method describes how to extract information from the strings and compute the distance between them resulting in similar matches. In the segmentation-free approach of [19], word matching is based on the vertical bar patterns. Each word is represented as a series of vertical bars that is used for the matching process. Word image matching is also applied in [13] using the weighted Hausdorff distance. Before applying the matching process using the Hausdorff distance a normalization scheme is used for each word. Word matching is also performed in [11] where global and local features based on profile signatures and morphological cavities are used for each word characterization. In [18] a voting system is used for fusion of multiple handwritten word recognition techniques based on ranks and confidence values.

In this work, we present an off-line handwriting word recognition system that is based on a novel combination of two different modes of word image normalization and robust hybrid feature extraction. The remaining of the paper is organised as follows. In Section 2, the pre-processing step is detailed while in Section 3 a novel robust hybrid feature extraction is presented. Experimental results are discussed in Section 4 and, finally, conclusions are drawn in Section 5.

2. Pre-processing

2.1. Word Image Normalization

At the word image normalization step we first remove the skew and then resize the word in order to fit in a rectangular box while preserving its aspect ratio. The exact positioning of the word in the rectangular box can be achieved by (i) using as a reference point the geometric center of the word image or by (ii) placing the baseline of the word in the center of the rectangular box. Both word skew and baseline detection is accomplished using the following methodology based on horizontal projections:

Let *im(x,y)* be the word image array having 1s for foreground and 0s for background pixels, x_{max} and y_{max} be the width and the height of the word image, respectively. We first calculate the left and the right horizontal word projections *LP* and *RP* (see Figure 1) as follows:

$$LP(y) = \sum_{x=0}^{\frac{x_{\max}}{2}} im(x,y), \qquad RP(y) = \sum_{x=\frac{x_{\max}}{2}}^{x_{\max}} im(x,y) \qquad (1)$$

Then, we calculate the global maxima of *LP* and *RP* projections for $y=y_L$ and $y=y_R$. At a next step, we calculate values y_{L1} , y_{L2} and y_{R1} , y_{R2} which correspond to the nearest y values from both sides of y_L and y_R having $LP(y) < 0.2LP(y_L)$ and $RP(y) < 0.2RP(y_R)$:

$$y_{L1} = y : (LP(y) < 0.2LP(y_L) \,\&\, y = \max(y_i)), y_i \in [0, y_L]$$

$$y_{L2} = y : (LP(y) < 0.2LP(y_L) \,\&\, y = \min(y_i)), y_i \in [y_L, y_{\max}] \qquad (2)$$

$$y_{R1} = y : (RP(y) < 0.2RP(y_R) \,\&\, y = \max(y_i)), y_i \in [0, y_R]$$

$$y_{R2} = y : (RP(y) < 0.2RP(y_R) \,\&\, y = \min(y_i)), y_i \in [y_R, y_{\max}]$$

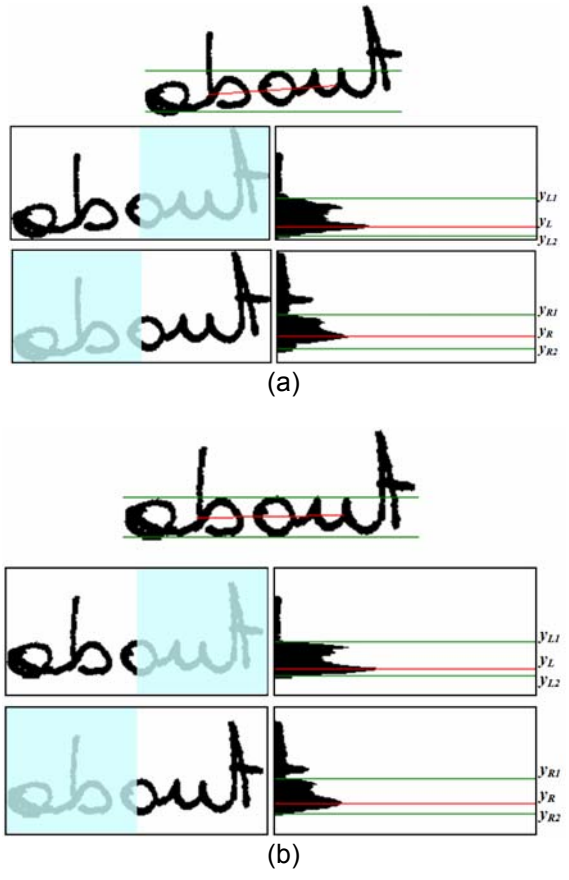

Figure 1. Skew correction of the word images. (a) The original word image and the left/right horizontal word projections; (b) The word image with corrected skew and the horizontal projections that help to accurately define the word baseline.

Due to the word skew, the distributions of the left and the right horizontal word projections *LP* and *RP* exhibit a vertical offset. The word skew is given by the following formula:

$$\theta = \tan^{-1}\left(\frac{y_{R1} + y_{R2} - y_{L1} - y_{L2}}{x_{\max}}\right) \qquad (3)$$

As shown in Figure 1b, after word skew correction, $y_{L1} \approx y_{R1}$ and $y_{L2} \approx y_{R2}$ and therefore the baseline is accurately detected in the y_{L1} - y_{L2} limits.

2.2. Slant Correction

The word slant is chosen as the slant which gives the minimum entropy of a vertical projection histogram [3]. The vertical projection histogram is calculated by counting the number of foreground pixels in each column of the binary image. The distribution is then normalised to have an area equals to 1.

The basic idea can be demonstrated using a vertical line as an example. When the line is slanted at an angle, it will have a low flat distribution. When the line is upright, the distribution will be tall and narrow, which will result in a lower entropy measure than for the low flat distribution of the slanted line.

The vertical projection histogram is calculated for a range of slant correction angles a_i, where the angle ranges in $\pm 45°$. The correction angle a_i is measured relative to the normal. The word slant, a_m, is given as :

$$a_m = \min_{a \in \pm R} H \qquad (4)$$

$$H = -\sum_{i=1}^{N} p_i \log p_i \qquad (5)$$

where N is the number of bins in the vertical projection histogram that equals to x_{max} and p_i is the probability of the foreground pixel appearing in bin i. The word is then corrected by a_m using :

$$x' = x - y\tan(a_m) \qquad (6)$$

$$y' = y \qquad (7)$$

An example of slant corrected word image is shown in Figure 2b.

2.3. Stroke Thickness Normalization

For the stroke thickness normalization process, we use an iterative skeletonization method presented in [12]. This method is simply an extension of Zhang and Suen's method [20]. The skeleton obtained is not truly 8-connected, since some non-junction pixels have more than two neighbors, making the skeleton useless for algorithms that require this constraint. Therefore, some pixels have to be removed. The skeleton is inspected, and each pixel is tested using a lookup table. The result is a true 8-connected skeleton where only junction pixels have more than two 8-neighbors (see Figure 2c). Finally, we normalize the stroke thickness by applying a dilation operator (see Figure 2d).

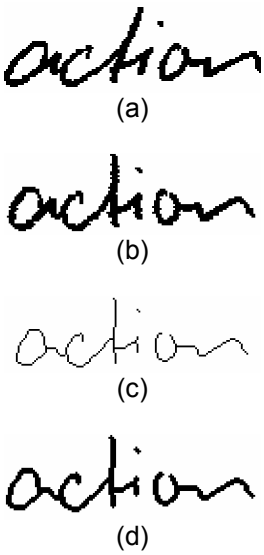

Figure 2. (a) Binarized image; (b) Slant correction; (c) Skeletonization; (d) skeleton dilation.

3. Hybrid feature scheme

For the word matching, feature extraction from the word images is required. Several features and methods have been proposed based on strokes, contour analysis, zones, projections etc. [1][2][4][17]. In our approach, we employ two types of features in a hybrid fashion. The first one, which is based on [2], divides the word image into a set of zones and calculates the density of the character pixels in each zone. The second type of features is based on the work in [17], where we calculate the area that is formed from the projections of the upper and lower profile of the word.

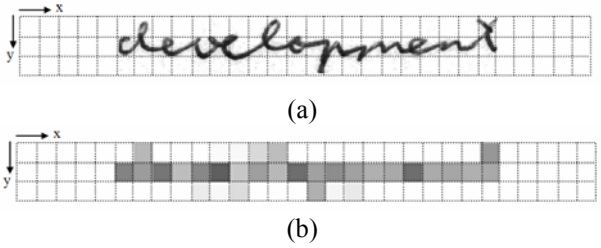

Figure 3. Feature extraction of a word image based on zones. (a) The normalized word image; (b) Features based on zones. Darker squares indicate higher density of character pixels.

In the case of features based on zones, the image is divided into horizontal and vertical zones. In each zone, we calculate the density of the character pixels (see Figure 3). Let Z_H and Z_V be the total number of zones formed in both horizontal and vertical direction. Then, features based on zones $f^z(i)$, i=0… Z_H Z_V -1 are calculated as follows:

$$f^z(i) = \sum_{x=x_s(i)}^{x_e(i)} \sum_{y=y_s(i)}^{y_e(i)} im(x,y) \qquad (8)$$

where,

$$x_s(i) = (i - \left\lfloor \frac{i}{Z_H} \right\rfloor Z_H)\frac{x_{\max}}{Z_H} \quad , \quad x_e(i) = (i - \left\lfloor \frac{i}{Z_H} \right\rfloor Z_H + 1)\frac{x_{\max}}{Z_H}$$

$$y_s(i) = \left\lfloor \frac{i}{Z_H} \right\rfloor \frac{y_{\max}}{Z_V} \quad , \quad y_e(i) = (\left\lfloor \frac{i}{Z_H} \right\rfloor + 1)\frac{y_{\max}}{Z_V}$$

In the case of features based on word (upper/lower) profile projections, the word image is divided into two sections separated by the horizontal line $y = y_t$ which passes through the center of mass of the word image (x_t, y_t) (see Eq. 9).

$$y_t = \frac{\sum_x \sum_y im(x,y) \cdot y}{\sum_x \sum_y im(x,y)} \qquad (9)$$

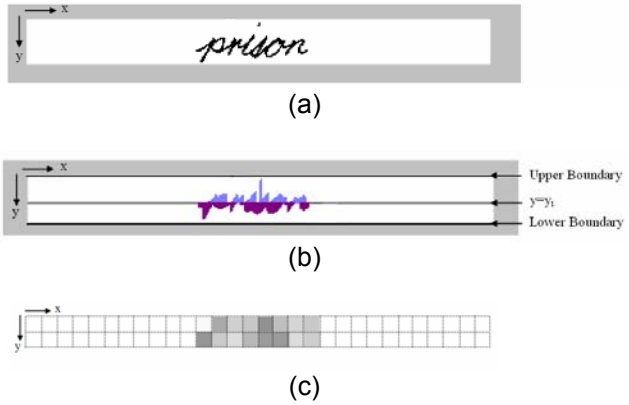

Figure 4. Feature extraction of a word image based on word profile projections. (a) The normalized word image; (b) Upper and lower word profiles; (c) The extracted features. Darker squares indicate higher density of zone pixels.

Upper/lower word profiles (Eq. (10),(11)) are computed by considering, for each image column, the distance between the horizontal line $y=y_t$ and the closest character pixel to the upper/lower boundary of the word image (see Figure 4):

$$y_{up}(x) = y_t - y_0,$$

$$\text{where } y_0 = \begin{cases} y_t, \text{ if } \sum_{y=0}^{y_t} im(x,y) = 0 \\ y : (im(x,y) = 1 \;\&\; y = \min(y_i)),\ y_i \in [0, y_t], \text{else} \end{cases} \qquad (10)$$

$$y_{lo}(x) = y_0 - y_t,$$

$$\text{where } y_0 = \begin{cases} y_t, \text{ if } \sum_{y=y_t}^{y_{\max}} im(x,y) = 0 \\ y : (im(x,y) = 1 \;\&\; y = \max(y_i)),\ y_i \in [y_t, y_{\max}], \text{else} \end{cases} \qquad (11)$$

Let P_V be the total number of blocks formed in each produced zone (upper, lower). For each block, we

calculate the area of the upper/lower word profiles denoted as in the following:

$$f^{p}{}_{up\_ar}(i) = \sum_{x=x_s(i)}^{x_e(i)} y_{up}(x) \quad (12)$$

$$f^{p}{}_{lo\_ar}(i) = \sum_{y=x_s(i)}^{x_e(i)} y_{lo}(x) \quad (13)$$

where,

$$x_s(i) = (i - \left\lfloor \frac{i}{P_V} \right\rfloor P_V) \frac{x_{\max}}{P_V} \quad , \quad x_e(i) = (i - \left\lfloor \frac{i}{P_V} \right\rfloor P_V + 1) \frac{x_{\max}}{P_V}$$

and $i=0$ .. P_V-1. Figure 4 illustrates the features extracted from a word image using projections of word profiles.

The overall calculation of the proposed hybrid feature vector is given in Eq. 14. The corresponding feature vector length equals to $Z_H Z_V + 2P_V$.

$$f(i) = \begin{cases} f^{z}(i) = \sum_{x=x_s(i)}^{x_e(i)} \sum_{y=y_s(i)}^{y_e(i)} im(x,y), & i = 0 \ldots Z_H Z_V - 1 \\ f^{p}{}_{up\_ar}(i) = \sum_{x=x_s(i-Z_H Z_V)}^{x_e(i-Z_H Z_V)} y_{up}(x), & i = Z_H Z_V \ldots Z_H Z_V + P_V - 1 \\ f^{p}{}_{lo\_ar}(i) = \sum_{x=x_s(i-Z_H Z_V + P_V)}^{x_e(i-Z_H Z_V + P_V)} y_{lo}(x), & i = Z_H Z_V + P_V \ldots Z_H Z_V + 2P_V - 1 \end{cases} \quad (14)$$

4. Experimental Results

For our experiments, we have used the IAM handwriting database v3.0 [10] that is publicly available and has been used by several research groups meanwhile [16]. The original database consists of 115320 isolated and labeled words. For a meaningful experimentation we have used 26970 words which have been correctly segmented as well as each of them having many instances. We have split the used dataset into a training set of 23171 words and a testing set of 3799 words.

As it has already been described in Sections 2 and 3 we have used a normalization step followed by a feature extraction step. During this, the size of the normalized word images used is x_{max}=300 and y_{max}=30. In the case of features based on zones, the word image is divided into three (Z_H =3) horizontal and thirty (Z_V =30) vertical zones forming a total of ninety (90) blocks with size *10x10* (see Figure 3). Therefore, the total number of features is ninety (90). In the case of features based on word (upper/lower) profile projections we keep the same size of the normalized image, while the image is divided into thirty (30) vertical zones (P_V =30) (see Figure 4). Consequently, the total number of features equals to sixty (60). Combination of features based on zones and features based on word profile projections led to the hybrid feature extraction model (Eq. 14) that uses a total of one hundred and fifty (150) features. Moreover, we have tested a combination of two different modes of normalization (baseline and geometric center adjustment) preceding the hybrid feature extraction scheme. In this case the extracted features are doubled (150 + 150).

For the particular classification problem, the classification step was performed using two well-known classification algorithms, namely the Minimum Distance Classifier (MDC) [7] and the Support Vector Machines (SVM) [5].

Formally, the support vector machines (SVM) require the solution of an optimisation problem, given a training set of instance-label pairs (x_i, y_i), i=1,…,m, where $x_i \in R^n$ and $y_i \in \{1,-1\}^m$. The optimisation problem is defined as follows :

$$\begin{aligned} \min_{\omega,b,\xi} \quad & \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{m}\xi_i \\ subject \ \ to \quad & y_i(\omega^T\phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \quad (15)$$

According to this, training vectors x_i are mapped into a higher dimensional space by the function ϕ. Then, SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. For this search, there are a few parameters that play a critical role at the classification performance. Firstly, the parameter C in Eq. 15, that applies a penalty at the error term. Secondly, the so-called kernel function denoted as: $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$.

In our case, SVM was used in conjunction with the Radial Basis Function (RBF) kernel, a popular, general-purpose yet powerful kernel, denoted as:

$$K(x_i, x_j) \equiv \exp(-\gamma \left\| x_i - x_j \right\|^2) \quad (16)$$

Furthermore, a grid search was performed in order to find the optimal values for both the variance parameter (γ) of the RBF kernel and the cost parameter (C) of SVM (see Eq. 15).

Table 1 depicts the (%) recognition rate achieved after combining different normalization and pre-processing modes as well as using either single features or the hybrid feature extraction scheme. We can draw several conclusions. First, in all cases the use of the hybrid model outperforms the use of a single feature either based on zones or based on projections. Second, the skew and slant correction, as well as the stroke thickness normalization pre-processing stages improve the performance of the classification system. Finally, the best performance is achieved by using the SVM classifier in the case of the combination of two different modes of normalization preceding the hybrid feature extraction scheme. The corresponding recognition rate equals to 87,68% and can be considered one of the highest performances among the state-of-the-art approaches for offline cursive handwriting word recognition. Similar efforts that have been tested against the IAM database have achieved a classification accuracy up to 80.76% [9].

Table 1. Experimental results

PRE-PROCESSING				NORMALIZATION		FEATURE EXTRACTION		CLASSIFIER	
Experiment	Skew correction	Slant correction	Stroke Thinkness normalisation	Baseline	Geom. Center	Zones	Projections	MDC	SVM
1	-	-	-	-	√	√	-	70,54%	76,18%
2	-	-	-	-	√	-	√	63,36%	69,49%
3	-	-	-	-	√	√	√	75,60%	80,71%
4	-	-	-	√	-	√	√	78,02%	82,97%
5	-	-	-	√	√	√	√	80,10%	84,23%
6	√	-	-	√	√	√	√	80,49%	84,23%
7	√	√	-	√	√	√	√	82,00%	85,39%
8	√	√	√	√	√	√	√	82,34%	**87,68%**

5. Conclusions

This paper proposes an off-line cursive word handwriting recognition methodology that is based on a novel combination of two different modes of word image normalization and robust hybrid feature extraction. After a validation of the proposed approach with the reference IAM database we have achieved a performance which one of the highest among the state-of-the-art.

Our future research will focus on exploiting new features as well as fusion methods to further improve the current performance.

References

[1] D. Bhat, "An evolutionary measure for image matching", *Proceedings of the 14th International Conference on Pattern Recognition*, ICPR'98, volume I, 1998, pp. 850-852.

[2] M. Bokser, "Omnidocument technologies", *Proceedings of the IEEE*, 80(7), 1992, pp. 1066-1078.

[3] R. Buse, ZQ Liu, "A structural and relational approach to handwritten word recognition", *IEEE Transactions on Systems, Man, and Cybernetics*. 27, 1997, pp. 847-861.

[4] S-H Cha, Y-C Shin, S. N. Srihari, "Approximate stroke sequence string matching algorithm for character recognition and analysis", *Proceedings of the 5th International Conference on Document Analysis and Recognition* (ICDAR'99), 1999, pp 53-56.

[5] C. Cortes, V. Vapnik, "Support-vector network", *Machine Learning* 20, 1995, pp. 273-297.

[6] B. Gatos, N. Papamarkos, C. Chamzas, "A binary tree based OCR technique for machine printed characters", *Engineering Applications of Artificial Intelligence*, 10(4), 1997, pp 403-412.

[7] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Addison Wesley, 1997.

[8] D. Guillevic, C. Y. Suen, "HMM word recognition engine", *Fourth International Conference on Document Analysis and Recognition* (ICDAR'97), 1997, pp 544-547.

[9] S. Günter, H. Bunke, "An Evaluation of Ensemble Methods in Handwritten Word Recognition Based on Feature Selection", *International Conference on Pattern recognition* (ICPR'04), 2004, 388-392.

[10] IAM handwriting database v3.0, http://www.iam.unibe.ch/~fki/iamDB/

[11] P. Keaton, H. Greenspan, R. Goodman, "Keyword spotting for cursive document retrieval", *Workshop on Document Image Analysis* (DIA'97), 1997, pp. 74-82.

[12] H. J. Lee, B. Chen, "Recognition of Handwritten Chinese Characters via Short Line Segments", *Pattern Recognition* 25 (5), 1992, pp. 543-552.

[13] Y. Lu, C. Tan, H. Weihua, L. Fan, "An approach to word image matching based on weighted Hausdorff distance", *Sixth International Conference on Document Analysis and Recognition* (ICDAR'01), 2001, pp 10-13.

[14] S. Madhvanath, V. Govindaraju, "Local reference lines for handwritten word recognition", *Pattern Recognition*, 32, 1999, pp 2021-2028.

[15] A. Marcolino, V. Ramos, M. Ármalo, J. C. Pinto, "Line and Word matching in old documents", *Proceedings of the 5th IberoAmerican Sympsium on Pattern Recognition* (SIARP'00), 2000, pp 123-125.

[16] U. Marti, H. Bunke, "The IAM-database: an English sentence database for off-line handwriting recognition", *International Journal of Document Analysis and Recognition*, 5,2002, pp. 39-46.

[17] T. M. Rath, R. Manmatha, "Features for word spotting in historical documents", *Proceedings of the Seventh International Conference on Document Analysis and Recognition* (ICDAR'03), 2003, pp 218-222.

[18] B. Verma, P. Gader, W. Chen, "Fusion of multiple handwritten word recognition techniques", *Pattern Recognition Letters* (22) 9, 2001, pp. 991-998.

[19] H. Weihua, C. L. Tan, S. Y. Sung, Y. Xu, "Word shape recognition for image-based document retrieval", *IEEE International Conference on Image Processing* (ICIP'01), 2001, pp 8-11.

[20] M. Zhang, C. Suen, *Digital Image Processing*, 2nd edition, 1987, pp. 398-402.