Handwritten Recognition with Multiple Classifiers for Restricted Lexicon

J. J. de Oliveira Jr.[1], M. N. Kapp[2], C. O. de A. Freitas[2], J. M. de Carvalho[1], R. Sabourin[3]
[1] UFCG - Universidade Federal de Campina Grande,
Coordenação de Pós-Graduação em Engenharia Elétrica,
Postal Box 10105, 58109-970, Campina Grande, PB - Brazil
{josemar,carvalho}@dee.ufcg.edu.br
[2] PUCPR - Pontificia Universidade Catolica do Parana,
Rua Imaculada Conceição 1155, 80215-901, Curitiba, PR - Brazil
{mnk,cinthia}@ppgia.pucpr.br
[3] ÉTS - Ecole de Technologie Superieure,
1100 Rue Notre Dame Ouest, H3C 1K3, Montreal, QC - Canada
robert.sabourin@etsmtl.ca

Abstract

This paper presents a multiple classifier system applied to the handwritten word recognition (HWR) problem. The goal is to analyse the influence of different global classifiers taken isolatedly as well as combined in a particular HWR task. The application proposed is the recognition of the Portuguese handwritten names of the months. The strategy takes advantage of the complementary mechanisms of three different classifiers: Conventional Neural Network, Class-Modular Neural Network and Hidden Markov Models, yielding a multiple classifier that is more efficient than either individual technique. The recognition rates obtained vary from 75.9% using the stand alone HMM classifier to 96.0% considering the classifiers combination.

1. Introduction

The main objective of this work is to analyse the influence of different global classifiers taken isolatedly as well as combined in a particular HWR task. The application proposed is the recognition of the Portuguese handwritten names of the months. This is an important task, since it constitutes a sub-problem of bank check date recognition. To achieve this goal, we evaluate the performance of a multiple classifier system. The development of an effective handwritten date processing system for bank checks is very challenging. The system must consider different data types, such as digits and words written in different styles (pure cursive, uppercase, spaced discrete, and mixed).

This study deals only with recognition of the month names represented by a limited lexicon of 12 classes: Janeiro, Fevereiro, Março, Abril, Maio, Junho, Julho, Agosto, Setembro, Outubro, Novembro and Dezembro. However, the names for some of these classes share a common sub-string, which adds to the problem complexity. As can be observed in Figure 1, there is similarity between the suffix of some classes in the lexicon, which increases the confusion and affects the performance of the recognizer. Another source of confusion is that some names have the same first letter (e.g. junho and julho) and this letter is very important in the word recognition process, as observed by Schomaker [1]. Further difficulty is added by the fact that the vowels (a, e, i, o) exhibit low discriminatory power in the human reading process [1]. The same lexicon has been studied in other works [2, 3, 4, 5], with system performance always being limited by the confusions indicated in Figure 1.

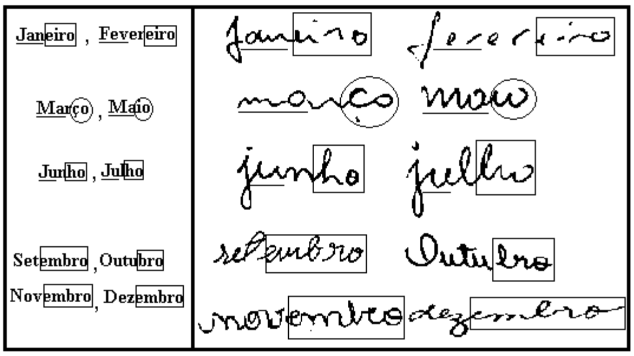

Figure 1. Complexity of the recognition problem: prefix and suffix.

In this work, we utilize an approach based on multiple classifiers, projected to avoid the intrinsic difficulties of the lexicon. This solution shows that by combining complementary information obtained with distinct classifiers, a better performance can be achieved than that of any individual classifier [6]. Therefore, this work presents a multiple classifier system based on three different classifiers: Conventional Neural Network - NN, Class-Modular Neural Network - CMNN and Hidden Markov Models - HMM. Figure 2 shows an overview of the system.

The image database and preprocessing operations are presented in Section 2. The NN strategy is described in Section 3, employing an implicit segmentation procedure followed by extraction of three different kinds of features for the neural classifier. Section 4 shows the CMNN scheme that consists of one feature extraction process based on prefix/suffix discrimination plus the class-modular neural network. The HMM classifier is described in Section 5, including feature extraction, topology and algorithms. In Section 6, the classifiers combination is described and the experimental results are presented and analyzed, in order to determine their discriminating potential, both individually as well as assembled. Finally, this paper concludes analyzing the overall system performance.

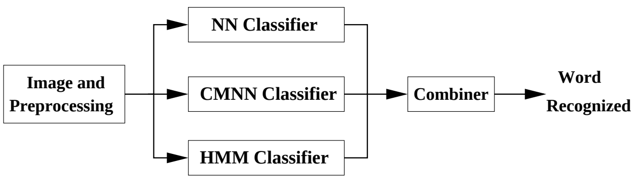

Figure 2. Overview of the multiple classifiers system.

2. Word Database

To develop the system it was initially necessary to construct a database that can represent the different handwriting styles present in the Brazilian Portuguese language. This was done by collecting samples of each month name, from 500 writers of different levels of education. Each writer was asked to fill an specific form where the word corresponding to each month name would be written once. No restrictions were imposed regarding writing style and no handwritten models were provided, which resulted in a very heterogeneous database. The words were digitized at 200 dpi. Figure 3 illustrates some samples from this database.

For the experiments, the database was randomly split into three data sets: Set 1 - Training Base with 3,600 words; Set 2 - Validation Base and Set 3 - Testing Base, both with 1,200 words. For each set, the words are evenly distributed among the classes.

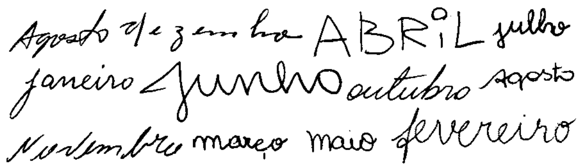

Figure 3. Sample images from the database.

2.1. Preprocessing

The form does not provide reference lines to the writer, resulting in words with different baseline skew and slant. To reduce this variability slant and baseline skew normalization algorithms were applied [2], using inclinated projection profiles and shear transformation.

3. Conventional Neural Network Classifier (NN)

3.1. Implicit Segmentation and Feature Extraction

A limitation with neural classifiers is the need for a fixed size input vector. To meet this requirement an implicit segmentation is performed spliting each sample image in to 8 sub-regions of equal size, as shown in Figure 4. This number corresponds to the average number of letters in the lexicon words. For each sub-region ten patterns are defined $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$, thus forming for each image a feature vector containing 80 patterns. Another requirement of the neural classifier are normalized input patterns, implying that all components of the feature vector need to be normalized based on the respective definition, as described in next section.

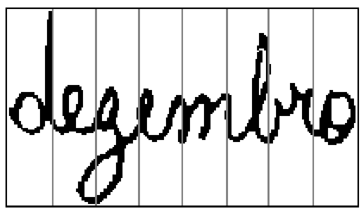

Figure 4. Example of implicit segmentation.

3.2. Perceptual Features

The perceptual features are considered high-level features due to the important role they play in human reading process, which uses features like ascenders, descenders and estimation of word lenght to read handwritten words [7].

To extract ascenders and descenders it is necessary to determine the image reference lines. To do this, the words horizontal projection histogram of black-white transitions is initially determined. The line with maximum histogram value is called Central Line (CL). Next, a smoothing procedure is applied to eliminate histogram discontinuities. The Upper (UL) and Lower (LL) Lines are the ones above and below CL, respectively, with 70% of the maximum histogram value [2]. The central region of the word, is defined as the area located between the UL and LL lines. An example of this procedure is presented in Figure 5.

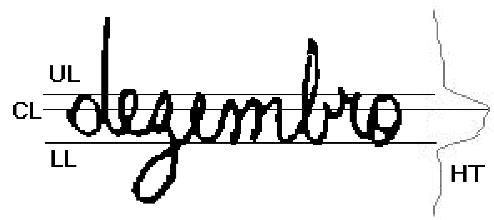

Figure 5. Example of reference lines detection.

The 10 patterns used in the perceptual feature set are:

- x_1 **- Ascender position:** Position of the ascender central pixel, normalized by sub-region width;
- x_2 **- Ascender size:** Height of ascender normalized by the height of central region;
- x_3, x_4 **- Descender position and size:** Same as defined for ascenders, considering the descender sub-region;
- x_5 **- Closed loop size:** Number of pixels inside a closed loop normalized by the respective sub-region area. A closed loop is defined as the region where from an internal (background) pixel a black (or contour) pixel is always reached for any moving direction;
- x_6, x_7 **- Closed loop location:** Coordinates of the closed loop center of mass. The x and y coordinates are normalized by the sub-region width and height, respectively;
- x_8, x_9 **- Concavity angles:** Initially the convex hull is constructed starting at the bottom-most point of the boundary as shown by Parker [8]. The leftmost and rightmost points in the hull are detected and the angles (relative to the horizontal) defined by the line segments joining them to the starting point are measured. The angles are normalized by 90°;
- x_{10} **- Estimated segment lenght:** Number of transitions (black-white) in the central line of the sub-region, normalized by the total number of transitions in the central line of the word. One transition is defined as the *background-foreground* or *foreground-background* transition outside of the closed loops.

When a pattern does not occurs in a sub-region it is necessary to assign a value to represent this absence. The zero value is not a good choice, because the occurrence of many null patterns would degrade the NN performance. Therefore, it was heuristicaly decided to assign 0.001 to indicate absence of a pattern.

3.3. Directional Features

The directional features can be considered intermediate-level features, conveying relevant information about the image background. In this paper, the directional features defined are based on concavity testing [8], where for each white image pixel (or background pixel) it is tested which of the four main directions (NSEW) leads to a black (contour) pixel, as shown in Figure 6. Labels are assigned to the background pixels according with Table 1 using the result of these directional tests. Label 10 is assigned to isolated letters, without ligature strokes.

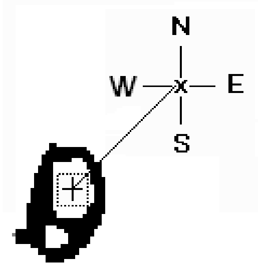

Figure 6. Example of directional features extraction.

For each of the image sub-regions, the components of the feature vector $(x_1, x_2, x_3,\ x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$ are obtained by counting the number of pixels assigned to the corresponding label, normalized by the sub-region area. When there are no pixels of a given label, the value 0.001 is assigned to the vector.

Label	Type
1	Closed in all directions
2	Open down
3	Open up
4	Open right
5	Open left
6	Open right and up
7	Open left and up
8	Open left and down
9	Open right and down
10	Open down and up

Table 1. Convention used for the directional feature set.

3.4. Topological Features

Topological features reflect pixel density over the image regions, being classified as low-level features. To determine these features a zoning was performed spliting each subregion in two parts, above and below the word central line. Furthermore, the upper and lower parts were each divided in 4 zones, as shown in Figure 7.

The feature vector components $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ represent the count of black pixels in each of the eight zones, normalized by the respective zone area. Components (x_9, x_{10}) correspond to the sub-region center of mass coordinates normalized by the sub-region width and height, respectively. When the number of black pixels is zero, the value mapped to the vector is 0.001.

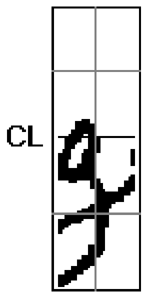

Figure 7. Example of zoning used.

3.5. Neural Classifier

The neural network used was of the MLP-type implemented via the SNNS simulator program [9]. Each NN is composed by 80 neurons in the input layer, one hidden layer and 12 neurons in the output layer. Input data is shuffled before presentation and the back-propagation with momentum algorithm plus one update function for optimizing the adaptation weights were used for training. Validation was employed in order to avoid over-learning. The error obtained in the validation set for each training epoch was used as stop criterion.

4. Class-Modular Neural Network Classifier (CMNN)

4.1. Feature Extraction

In this classifier, perceptual features [7] and characteristics based on concavities / convexities are represented by the number of their occurrences. However, only these discrete primitives are not enough to produce a robust recognition system [10]. Therefore, a zoning mechanism was added to the features set during primitives extraction.

Zoning splits the image in two areas, defined at the right and at the left of the word center of gravity, as shown in Figure 8-a. This mechanism explores the information provided by the occurrence of features in each specific zone.

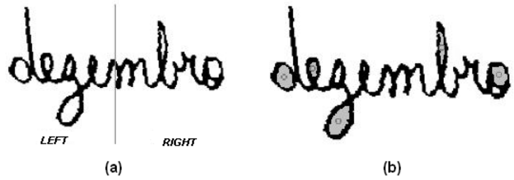

Figure 8. Example of zoning used: (a) areas detection and (b) loops detection.

The feature set can be described as following:

- Number of loops on the left/right-areas, Figure 8-b;
- Number of concave and convex semicircles on the left/right-areas, Figure 9-a and Figure 9-b, respectively. The concave and convex points are obtained by mathematical morphology;
- Number of crossing-points, branch-points and end-points on the left/right-areas, Figures 9-c, 9-d and 9-e, respectively;
- Number of horizontal axis crossings by stroke (NHAC), Figure 9-f;
- Number of ascenders and descenders on the left/right-areas;

- Proportion of white/black pixels inside the word bounding box (NPP), Figure 9-g, given by

$$\text{prop} = \frac{tp - tpp}{tp}$$

where tp and tpp are, respectively, the total number of pixels and the amount of black pixels inside the bounding box;

- Number of vertical lines, Figure 9-h, number of horizontal lines, number of ascenders with loop on the left/right-areas and number of descenders with loop on the left/right-areas.

These 14 features are extracted from each word in order to generate a feature vector of dimension 24. When a feature is not found in the word, a small value is assumed, for our case, 0.001.

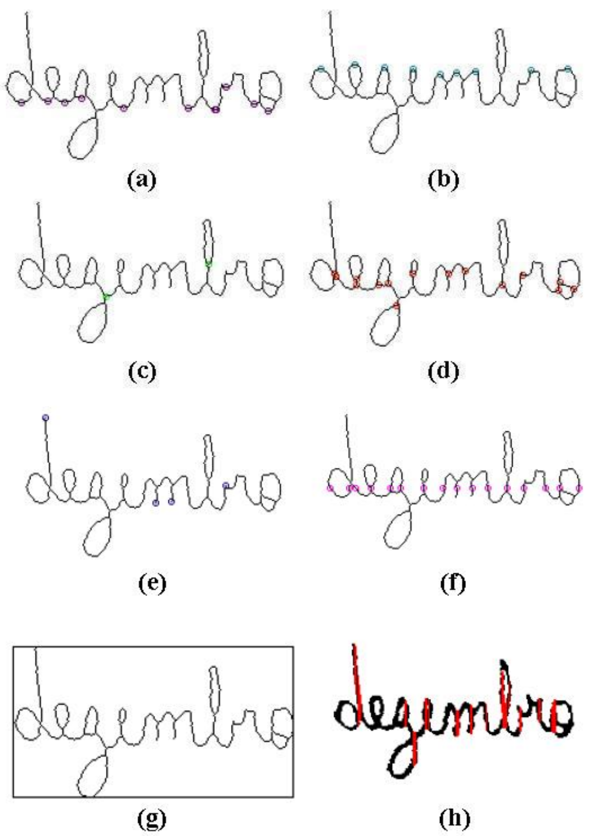

Figure 9. Feature extraction: a) concave semicircles, b) convex semicircles, c) crossing-points, d) branch-points, e)end-points, f) NCH, g) NPP and h)vertical lines.

4.2. Class-Modular Classifier

A single task is decomposed into multiple subtasks and each subtask is allocated to an expert network. In this paper, as well as in Oh et al. [11], the K-classification problem is decomposed into K 2-classication subproblems. For each one of the K classes a 2-classification subproblem is solved by the 2-classifier specifically designed for that class.

Therefore, the 2-classifier is only responsible for one specific class and discriminates that class from the other $K - 1$ classes. In the class-modular framework, K 2-classifiers solve the original K-classification problem cooperatively and the class decision module integrates the outputs from the K 2-classifiers.

In Figure 10-a, we can see the MLP architecture for a 2-classifier. The modular MLP classifier consists of K sub-networks, M_i for $0 \leq i \leq K - 1$, each responsible for one of the K classes. The architecture for the entire network is shown in Figure 10-b.

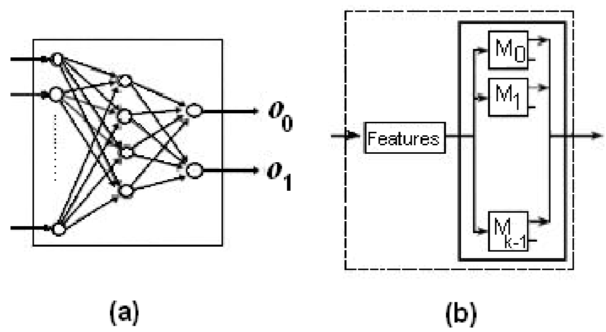

Figure 10. Class-modular architecture [11]: (a) sub-network and (b) whole network with K modules.

5. HMM Classifier

The features utilized by the HMM classifier are the same presented in Section 3.2 and 3.3, although using different extraction and representation, both adapted to this approach. The features are extracted from the word images and a pseudo-segmentation process is applied to obtain a sequence of corresponding observations. A segment is delimited by two consecutive black-white transitions over the Central Line. A symbol is designated to represent the extracted set of features for each segment, making up a grapheme. Transitions that are found inside the loops of the word body are not considered in this analysis. In case of no feature being extracted from the analyzed segment, an empty symbol denoted by X is emitted. This feature set is capable of representing the ligature between letters and separating graphemes. The character # denotes a separator between two graphemes. Figure 11 illustrates the feature extraction and segmentation processes.

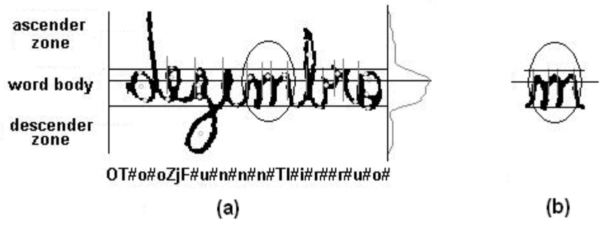

Figure 11. Example of feature extraction: a) feature set and b) segmentation.

In order to select an informative subset of graphemes to be used as input data to the HMM a mutual information criterion was used to define the symbol alphabet [5]. This criterion is based on the information content of each extracted feature and on the occurrence of combinations of these features in the same pseudo-segment. The entire and definitive alphabet is composed of 29 different symbols selected from all possible symbol combinations, using the mutual information criterion.

5.1. Word Recognition Method

Our word HMM models are based on a left-right discrete topology where each transition can skip at most two states. The lexicon size allows one model for each class. Model training is based on the Baum-Welch Algorithm and the Cross-Validation process [12, 13]. The objective of the Cross-Validation process is to monitor the general outcome during the training process. It is done on two data sets: training and validation. After the Baum-Welch Algorithm iteration on the training data, the likelihood of the validation data is computed using the Forward Algorithm [12]. During the experiments, the matching scores between each model λ_i and an unknown observation sequence O are carried out using the Forward algorithm.

6. Experimental Results

For each feature set presented in Section 3, one NN is trained and tested. The class that presents the maximum output value is the class recognized. The amount of neurons in the hidden layer was empirically determined, different configurations being tested. The best results were obtained using 75, 80 and 85 neurons for perceptual features (NN-P), directional features (NN-D) and topological features (NN-T), respectively.

For the class-modular MLP, each of the K 2-classifiers is independently trained using the training and validation sets. The backpropagation algorithm was used in each case in the same way as in the conventional MLP. To train a 2-classifier for each word class, we reorganize the original training and validation sets into two sub-sets, Z_0 and Z_1. Z_0 has the samples from current class and Z_1 contains the samples from all other classes, taking into account the a priori probability for each class. To recognize the input patterns, the class decision module considers only the O_0 outputs from each subnetwork (Figure 10-a) and uses a simple winner-takes-all scheme to determine the final class.

The HMM scheme was evaluated with the same sets used for the other classifiers and for each class one model was trained and validated. The model that assigns maximum probability to one test image represents the class recognized.

Table 2 shows the results obtained for each scheme individually. It can be seen that the best results were obtained using the Conventional NN with the perceptual feature set. Table 3 presents the confusion matrix for the NN-P classifier. The rows in the matrix represent the classification result.

Classifier	NN-P	NN-D	NN-T	CMNN	HMM
RR	**86.8 %**	86.1 %	84.9 %	83,0%	75.9 %

Table 2. Recognition rate obtained for each classifier individually.

	J	F	M	A	M	J	J	A	S	O	N	D
J	**77**	9	1	1	2	5	2			1	1	1
F	10	**80**			2	2	4				1	1
M		3	**90**	1	2	1	1		1		1	
A			4	**90**	2				1	2		1
M	1	1	2	1	**88**		3		2		1	1
J	1		1			**88**	2		3	2	1	2
J		1		3		3	**92**		1			
A	1	1		6				**89**				3
S	2	1		1					**84**	8	2	2
O	1	1		1		1			1	**94**		1
N	1	2	1	1			1		7	1	**86**	
D	2	2			1	1	1	3	3		4	**83**

Table 3. Confusion matrix for NN and perceptual feature set (NN-P).

6.1. Classifiers Fusion

To obtain the hybrid classifier it is necessary to define a combination rule for the classifiers output. In this work, three combining strategies have been considered based on the definitions presented by Kittler et al. [6]. Initially, we make the assumption that an object Z must be assigned to one of the K possible classes $(w_1, \cdots, w_K)$ and assume that L classifiers are available each representing the given pattern by a distinct measurement vector. Denote the measurement vector used by the ith classifier as x_i and the a posteriori probability $P(w_j|x_1, \cdots, x_L)$. Therefore the combining rules are:

- Sum (S): Assigns Z to class w_j if

$$\sum_{i=1}^{L} p(w_j|x_i) = \max_{k=1}^{K} \sum_{i=1}^{L} p(w_k|x_i); \qquad (1)$$

- Product (P): Assigns Z to class w_j if

$$\prod_{i=1}^{L} p(w_j|x_i) = \max_{k=1}^{K} \prod_{i=1}^{L} p(w_k|x_i); \qquad (2)$$

- Weighted sum (WS): Assigns Z to class w_j if

$$\sum_{i=1}^{L} \alpha_i \cdot p(w_j|x_i) = \max_{k=1}^{K} \sum_{i=1}^{L} \alpha_i \cdot p(w_k|x_i); \qquad (3)$$

  where α_i, $i = 1, \cdots, L$ are weights for the classifiers.

To guarantee that the classifier outputs represent probabilities, an output normalization is performed:

$$P^*(w_j|x_i) = \frac{P(w_j|x_i)}{\sum_K P(w_j|x_i)}. \qquad (4)$$

For the weighted sum rule, the optimum weights are obtained by an exhaustive search procedure where for each classifiers combination 2,000 different weight vectors with random adaptation are tested.

The average recognition rates obtained considering different classifiers combination are presented in Table 4. It can be seen that the best result was obtained using combination by weighted sum of NN-P, NN-D, NN-T, CMNN and HMM classifiers. This result means that for this specific problem and considering these classifiers the weighted sum rule represents the best solution. However, Table 4 shows that other combining strategies produce recognition rates very close to 96.0%. This indicates that a different best result can be obtained for a distinct problem. The confusion matrix for the present best result is shown in Table 5.

Classifiers	Fusion rules		
	S (%)	P (%)	WS (%)
NN-P and NN-D	90.0	90.9	91.1
NN-P and NN-T	89.8	90.8	90.8
NN-D and NN-T	88.4	88.7	89.1
NN-P and CMNN	91.7	90.8	92.5
NN-D and CMNN	91.5	92.0	92.2
NN-T and CMNN	91.2	90.6	91.2
NN-P and HMM	90.2	90.2	90.7
NN-D and HMM	90.6	90.9	90.6
NN-T and HMM	89.1	89.3	89.2
NN-P, NN-D and NN-T	92.0	91.3	92.2
NN-P, NN-D and CMNN	95.0	94.3	95.2
NN-P, NN-D and HMM	93.1	93.1	93.2
NN-P, NN-T and CMNN	94.2	93.7	94.6
NN-P, NN-T and HMM	92.6	92.4	93.0
NN-D, NN-T and CMNN	93.8	92.8	94.1
NN-D, NN-T and HMM	91.8	92.6	92.3
NN-P, CMNN and HMM	94.0	93.8	94.3
NN-D, CMNN and HMM	93.2	94.0	93.7
NN-T, CMNN and HMM	93.3	93.5	93.7
NN-P, NN-D, NN-T and CMNN	94.9	94.3	95.6
NN-P, NN-D, NN-T and HMM	93.5	93.0	93.7
NN-P, NN-D, CMNN and HMM	95.5	95.4	95.7
NN-P, NN-T, CMNN and HMM	95.5	94.7	95.6
NN-D, NN-T, CMNN and HMM	94.8	94.3	95.2
NN-P, NN-D, NN-T, CMNN and HMM	95.3	95.2	**96.0**

Table 4. Recognition rate obtained using different classifiers combination.

	J	F	M	A	M	J	J	A	S	O	N	D
J	**93**	4			2							
F	4	**96**										
M	1		**99**									
A				**100**								
M	1		3		**93**		2				1	
J	1	1		1		**95**			1	1		
J				1			**98**			1		
A			1	3				**93**				3
S									**98**		1	1
O									1	**99**		
N	1	1							4		**94**	
D		2				1			1		2	**94**

Table 5. Confusion matrix for the best result.

7. Discussion and Conclusions

Comparing the confusion matrix for the best individual classifier (shown in Table 3) with that of the best combination (presented in Table 5) an improvement of 10.6% in the average recognition rate can be observed. An overall improvement can be observed for all classes, mainly for classes $Janeiro$ and $Fevereiro$ that present a rate increase of approximately 20%. This result shows that combining different classifiers by a hybrid approach can yield a significant performance improvement.

To summarize, this paper presents a multiple classifier hybrid system applied to the recognition of the Portuguese handwritten names of the months. This system is based on a Global Approach, which extracts global features from the word image, hence avoiding the need for explicit segmentation. This approach explores word context information, while allowing for aspects based on psychological models. Therefore, unlike other proposed systems, it is not dependent on the sucess of a segmentation process.

We have evaluated the efficiency of combining NN, CMNN and HMM classifiers for a problem of handwritten word recognition. The main conclusion obtained is that the analyzed classifiers are complementary and the combining strategy proposed enhances their complementarity. Therefore, the multiple classifier system is a better solution to the analyzed problem than either of the classifiers taken individually. This result indicates that a similar strategy can be applied to other restricted lexicons.

Performance comparison with other reported cursive handwritten recognition systems is not easy, due to the use of distinct databases and/or lexicons. The only known studies for the Portuguese language were published by Morita et al. [4], which utilized the same lexicon as ours from a distinct database, having achieved a recognition rate of 91.5% using an analytical approach with verification. Another similar study by Kim et al. [14] has combined HMM and MLP classifiers to a lexicon with 12 classes, corresponding to the english month names, extracted from the CENPARMI database. For the standalone MLP classifier with 12 classes, corresponding to the full english month names, a recognition rate of 79.4% was achieved. For the combined HMM-MLP classifier this rate goes up to 88.8%. This comparison shows that our system obtains better rates than other similar systems recently reported. Future work will focus on the analysis of rejection mechanisms.

Acknowledgements

The authors would like to thank CAPES-PROCAD for the financial support of this work.

References

[1] L. Schomaker and E. Segers. "A Method for the Determination of Features used in Human Reading and Cursive Handwriting". *In proceedings of IWFHR'1998*, pp.157-168, Bangalore, India, 1998.

[2] J. J. de Oliveira Jr., J. M. de Carvalho, C. O. de A. Freitas and R. Sabourin. "Evaluating NN and HMM Classifiers for Handwritten Word Recognition". *In proceedings of SIBGRAPI'2002*, pp. 210-217, Brazil, 2002.

[3] M. N. Kapp, C. O. de A. Freitas, J. Nievola and R. Sabourin. "Evaluating the Conventional and Class-Modular Architectures FeedForward Neural Networks for Handwritten Word Recognition". *In proceedings of SIBGRAPI'2003*, pp. 315-319, Brazil, 2003.

[4] M. Morita, R. Sabourin, F. Bortolozzi and C. Y. Suen. "Segmentation and Recognition of Handwritten Dates: An HMM-MLP Hybrid Approach". *IJDAR*, 6:248-262, 2004.

[5] C. O. de A. Freitas, F. Bortolozzi and R. Sabourin. "Handwritten Isolated Word Recognition: An Approach Based on Mutual Information for Feature Set Validation". *In proceedings of ICDAR'2001*, pp.665-669, Seattle, USA, 2001.

[6] J. Kittler, M. Hatef, R. P. W. Duin and J. Matas. "On Combining Classifiers". *IEEE Trans. on PAMI*, 20(3):226-239, 1998.

[7] S. Madhvanath and V. Govindaraju. "The Role of Holistic Paradigms in Handwritten Word Recognition". *IEEE Trans. on PAMI*, 23(2):149-164, 2001.

[8] J. R. Parker. *Algorithms for Image Processing and Computer Vision*. Jonh Wiley & Sons, 1997.

[9] A. Zell et al. *SNNS - Stuttgart Neural Network Simulator, User Manual*. University of Stuttgart, 1994.

[10] O. D. Trier, A. K. Jain and T. Taxt. "Feature Extraction Methods for Character Recognition - A Survey". *Pattern Recognition*, 29:641-662, 1996.

[11] I-S. Oh and C. Y. Suen. "A Class-Modular FeedForward Neural Network for Handwriting Recognition". *Pattern Recognition*, 35:229-244, 2002.

[12] L. Rabiner and B. H. Juang. *Fundamental of Speech Recognition*. Prentice Hall Inc., 1993.

[13] A. El Yacoubi, M. Gilloux, R. Sabourin and C. Y. Suen. "Unconstrained Handwritten Word Recognition Using Hidden Markov Models". *IEEE Trans. on PAMI*, 21(8):752-760, 1999.

[14] J. H. KIm, K. K. Kim, C. P. Nadal and C. Y. Suen. "A Methodology of Combining HMM and MLP Classifiers for Cursive Word Recognition". *In proceedings of ICPR'2000*, Barcelona - Spain, 2000.