Evaluating NN and HMM Classifiers for Handwritten Word Recognition

José J. de Oliveira Júnior[1], João M. de Carvalho[1], Cinthia O. de A. Freitas[2], Robert Sabourin[3]

[1]UFCG - Federal University of Campina Grande, Department of Electrical Engineering,
Postal Box 10105, 58109-970, Campina Grande, PB - Brazil
{josemar,carvalho}@dee.ufpb.br
[2]PUCPR - Pontíficia Universidade Católica do Paraná, Rua Imaculada Conceição 1155, Prado Velho
80215-901, Curitiba, PR - Brazil
cinthia@ppgia.pucpr.br
[3]ÉTS - École de Technologie Supérieure, 1100 Rue Notre Dame Ouest
H3C 1K3, Montreal, QC - Canada
sabourin@gpa.etsmtl.ca

Abstract. This paper evaluates NN and HMM classifiers applied to the handwritten word recognition problem. The goal is analyse the individual and combined performance of these classifiers. They are evaluated considering two different combination strategies and the experiments are performed with the same database and similar feature sets. The strategy proposed takes advantage of the different but complementary mechanisms of NN and HMM to obtain a more efficient hybrid classifier. The recognition rates obtained vary from 75.9% using the HMM classifier alone to 90.4% considering the NN and HMM combination.

1 Introduction

The main objective of this work is to evaluate the performance of Neural Networks (NN) and Hidden Markov Models (HMM) classifiers for a particular handwritten recognition problem. These are the main classifiers used in word recognition tasks by several authors [1, 2, 3]. Preliminary evaluation studies combining both classifiers in hybrid approaches have been reported [4, 5, 6]. However a more detailed analysis is necessary to determine their potential, either when considered individually or when combined.

NNs classifiers are well adequated to high dimensional space problems that present complex interactions between their variables, suggesting their aplication to handwritten word recognition. The main characteristics of these classifiers are a computation process that is inherently parallel, possibility of hardware implementation and emulation of the human learning process [7]. These classifiers exhibit powerful discriminative properties and have been used for handwritten recognition particularly with isolated characters, digits and words in small vocabularies [4].

HMMs have been applied intensively in the speech and handwritten word recognition areas because it is a method that uses probability measures to model sequential data represented by sequences of observation vectors [8]. Our interest in HMMs lies in their potential to efficiently model different knowledge sources. They correctly integrate different modeling levels (morphological, lexical, syntactical), and also provide efficient algorithms to determine an optimum value for the model parameters.

In this paper, we present and analyze the performance of NNs and HMMs applied to handwritten word recognition. The application involves 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. Although this study deals with a limited lexicon of 12 classes, there are classes that share a common sub-string, which can affect the overall system performance: Jan***eiro***, Fever***eiro***, ***Ma***rço, Abril, ***Ma***i***o***, ***Ju***n***ho***, ***Ju***l***ho***, Agosto, Set***embro***, Outu***bro***, Nov***embro*** and Dez***embro***.

Initially, the pre-processing operations applied to the image database is presented in Section 2. In Section 3 the NN scheme is presented, consisting of one implicit segmentation procedure, followed by extraction of two different kinds of features for the neural classifier. In Section 4 the HMM scheme is presented, including feature extraction, topology and applied algorithms. In Section 5 the experimental results are presented and analyzed, in order to determine the discriminating potential for each classifier. This is done based on the recognition rates obtained, considering each scheme individually as well as combinations of them. Finally, this paper is concluded analyzing the overall performance. Figure 1 shows a system overview.

2 Preprocessing

The preprocessing stage is used to reduce the effect of writing variability caused by different writing styles, particular writing characteristics and word slant. The operation of baseline skew and slant correction inspired by [3] and

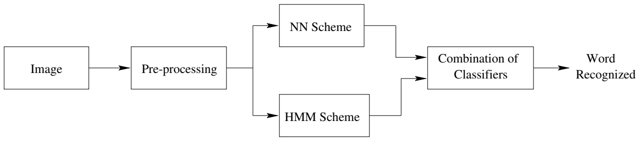

Figure 1: Overview of the NN/HMM system.

smoothing applied at this stage are described next:

- Slant Normalization - Corrects the characters average slant, calculated from an inclinated projection profile. This profile is obtained varying the inclination angle with respect to the Y-axis from -60 to 60 degrees by one degree steps, and calculating for each angle the image histogram. Next the entropy for each histogram is determined. The angle that produces the lowest entropy is the desired correction angle. Finally, a shear transformation is used to perform the correction.

- Baseline Skew Normalization - Detects and corrects the baseline skew of the word, calculated from horizontal inclinated projection profiles of the images lower contour. The profiles are obtained calculating the projected histogram for angles varying between -60 and 60 degrees with the Y-axis by one degree steps. As in slant normalization, the angle that presents lowest entropy is the skew angle. A shear transformation is again applied to correct the images.

- Smoothing - Eliminates imperfections in the word contour, like speckles and cavities, producing a smooth and continuous contour line using masks.

Judging by subjective visual inspection, the preprocessing applied produces good results in 99% of the images forming the database. Figure 2 examplifies the results obtained at this stage.

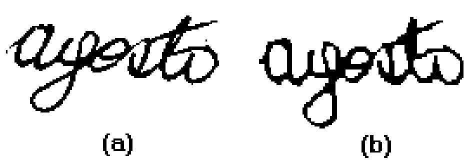

Figure 2: Example of pre-processing: (a) original image and (b) preprocessed image.

3 NN Scheme

3.1 Implicit Segmentation and Feature Extraction

A limitation with neural classifiers is the need for a fixed size input vector. To solve this problem an implicit segmentation is performed by which each sample image is divided in 8 sub-regions, as shown in Figure 3. 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 a feature vector containing 80 patterns for each image. Another requirement of the neural classifier are normalized input patterns, implying that all components of the feature vector need to be normalized accordingly with the features definition, as described in the next section.

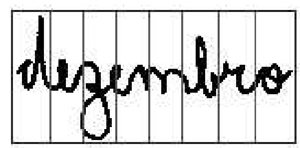

Figure 3: Example of implicit segmentation.

Two different feature sets were determined based in the classification system proposed by Madhvanath [9], a perceptual and a directional feature set. They are described next.

3.2 Perceptual Features

The perceptual features are considered high-level features and their utilization is justified by the human reading process, which uses features like ascenders, descenders and estimation of word lenght to read handwritten words [9].

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 (HT) is initially determined. The line with maximum histogram value is called Central Line (CL). After this, 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. This percentage was obtained heuristically by Freitas [1]. An example of this procedure is presented in Figure 4.

The 10 patterns used in the perceptual feature set are:

- x_1 **- Ascender position:** Position of the ascender central pixel, normalized by the sub-region width;

- x_2 **- Ascender size:** Height of ascender normalized by the height of central sub-region;

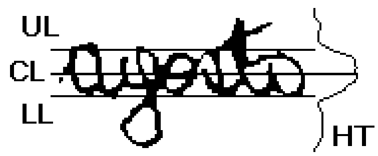

Figure 4: Example of reference lines detection.

- 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 the closed loop normalized by the sub-region area. A closed loop is defined as the region where from an internal pixel a black 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 in [10]. 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 word 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 decided to assign unity value to indicate absence of a pattern, the same value used when a given pattern assumes value greater than 1.0.

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 inspired by the idea of concavity measurements [10], where for each white image pixel it is verified in each of the main four directions (NSEW) if a black pixel can be reached, as shown in Figure 5.

Depending on the number and combination of the open directions the background pixels are labeled by the convention depicted in Table 1.

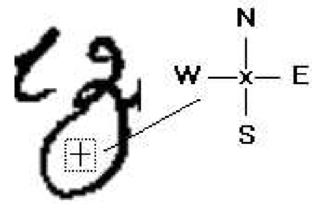

Figure 5: Example of directional features extraction.

Table 1: Convention used for the directional feature set.

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

Label 9 is used to represent letters without ligature strokes. The components of the feature vector $(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$ for each sub-region are obtained by counting the number of pixels attributed to each label, normalized by the sub-region area. When there are no pixels of some label, the corresponding value mapped to the vector is 1.0.

3.4 Neural Classifier

The neural network (NN) used was of the MLP-type implemented via the SNNS simulator program [13]. Each NN is composed by 80 neurons in the input layer, one hidden layer of variable size 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. Training with validation was employed in order to avoid overlearning. The error obtained in the validation set for each training epoch was used as stop criterion.

4 HMM Scheme

4.1 Feature Extraction

Three zones are established based on the horizontal transition histogram: ascender, body and descender. The body of the word is the area located between the UL and LL lines, defined in Section 3.2. The features are extracted from the word images and a pseudo-segmentation process is applied to obtain a sequence of corresponding observations, as shown in Figure 6. A segment is delimited by two consecutive black-white transitions over the Central Line. A corresponding 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.

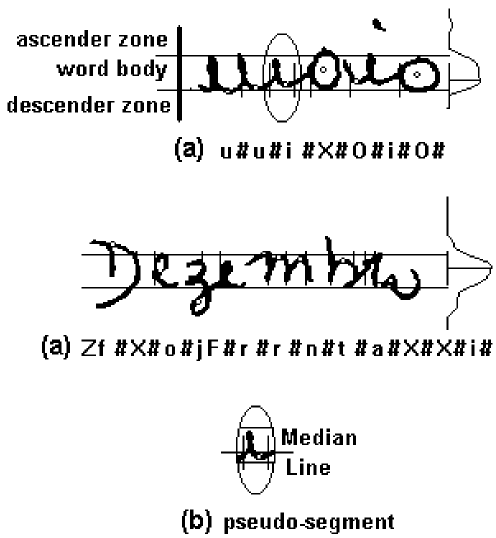

Figure 6: Feature extraction: a) PFCCD and b) segmentation.

Figure 6-a ilustrates the PFCCD feature set based on perceptual features, concavities and convexities deficiencies. It takes into account the concave and convex deficiencies in addition to the perceptual features (ascenders, descenders and loops). Concavities and convexities deficiencies in the word body are extracted and labeled as presented in Parker [10] and extracted as shown in Figure 6-b. These deficiencies are obtained by labeling the background pixels of the input images [11]. The character # denotes a separator between two graphemes. This feature set is a classification capable of representing the ligature between letters and separating graphemes made up of C, S, E and Z or, u, n, r and i.

4.2 Word Recognition Method

The Hidden Markov Model (HMM) theory successfully has been used to model handwritten variability. The theory of HMM is beyond the scope of this paper. An excellent introduction to this subject can be found in Rabiner & Juang [8]. Our interest in the HMM is due to its ability to efficiently model different knowledge sources. It correctly integrates different modeling levels (morphological, lexical, syntactical), and also provides efficient algorithms to determine an optimum value for the model parameters.

Our word HMM models are based on a left-right discrete topology (Bakis Topology), where each transition can skip at most two states. The lexicon size allows one model for each class (12 classes).

Model training is based on the Baum-Welch Algorithm and the Cross-Validation process, as done by Rabiner & Juang and El Yacoubi [8, 12]. The objective of the Cross-Validation process is to monitor the general outcome during the training process. It is done over two sets of data: 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 [8]. The training process computes

$$\bar{\lambda} = argmax(P(\lambda|O_i, i = 1, ..., t)) \quad (1)$$

where $\bar{\lambda}$ is the reestimated model, O_i are observation sequences in the training database and $P(\lambda|O_t)$ is the probability of observation sequence O given model λ.

During the experiments, the matching scores between each model λ_i and an unknown observation sequence O are carried out using the Forward Algorithm, as following:

$$P(O|\lambda_k) = argmax(P(O|\lambda_i)) \quad (2)$$

where λ_k is the model for the recognized class.

5 Experimental Results

This section describes the characteristics of the used database and presents the results obtained with the schemes described, individually and combined.

5.1 Database

To develop the system it was necessary to construct a database that can represent as well as possible the different styles of handwriting present in the Brazilian Portuguese language. This was done by collecting 500 samples of each month name, from writers of different levels of education. Each writer was asked to fill a 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 using a scanner set to 200 DPI. Figure 7 illustrates some samples from this database.

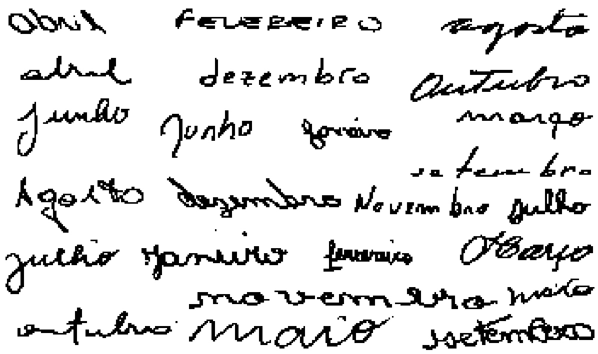

Figure 7: Sample images from the database.

The complete database has a total of 6000 words, 500 for each class. For the experiments it was randomly split into three data sets: Set 1 - Training Base with 3600 words (60% of the total); Set 2 - Validation Base and Set 3 - Testing Base, both with 1200 words (20% of the total, each). For each set, the words are evenly distributed among the classes.

5.2 Results with Individual Schemes

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 quantity of neurons in the hidden layer is empirical, different configurations being tested. The best results were obtained using 75 and 80 neurons for perceptual features (NN-P) and directional features (NN-D), respectively. The HMM Scheme was evaluated with the same samples used in NN Scheme 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 NN Scheme with the perceptual feature set. Tables 3, 4 and 5 show the confusion matrix obtained for each feature set.

Table 2: Average recognition rate obtained for each scheme individually.

Set	Recognition rate
Perceptual (NN-P)	81.8 %
Directional (NN-D)	76.6 %
HMM	75.9 %

5.3 Results with Combined Schemes

To evaluate the combined potential of the NN and HMM the individual networks outputs (considering both feature sets) and the estimated probability for each model were combined.

The probability $P(O|\lambda_i)$ has small values ($\approxeq 10^{-8}$) which can affect the combination outcome since the order of the networks output values is $\approxeq 10^{-2}$. To deal with this problem a normalized probability was calculated for model λ_i:

$$P^*(O|\lambda_i) = \frac{P(O|\lambda_j)}{\sum_j P(O|\lambda_j)} \tag{3}$$

Let $f(n_v, F_i)$ denote the output of neuron n_v associated with class v given feature vector F of classifier i and $\Psi = 1, 2, 3 \ldots, v$ denote a set of word classes.

Using this definitions, two combination strategies have been considered:

- Combination by average: The value assigned to each class is the average of the neural networks outputs and the normalized probability of that class.

$$v^* = argmax \left(\frac{f(n_v, F_i) + f(n_v, F_j) + P^*(O|\lambda_v)}{3} \right) \tag{4}$$

  where v^* is the recognized class and F_i, Fj are differente feature sets.

- Combination by multiplication: The value assigned to each class is the product of the neural networks outputs and the normalized probability for that class.

$$v^* = argmax \left(f(n_v, F_i) \times f(n_v, F_j) \times P^*(O|\lambda_v) \right) \tag{5}$$

The average recognition rates obtained considering different scheme combinations are presented in Table 6. The best result was obtained using combination by multiplication of HMM, NN-P and NN-D. The confusion matrix to this result is presented in Table 7.

Table 6: Recognition rate obtained using hybrid approaches.

	By averaging	By multiplication
Sets	Rec. Rate(%)	Rec. Rate (%)
NN-P and NN-D	87.2	87.0
HMM and NN-P	88.1	88.7
HMM and NN-D	87.0	87.6
HMM, NN-P and NN-D	90.2	**90.4**

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

Month	J	F	M	A	M	J	J	A	S	O	N	D
Janeiro	**75**	10	2	1	2	4	4				2	
Fevereiro	7	**77**	1	1	5	1	2			1	4	1
Março		1	**84**	5	4	1	1	2			1	1
Abril		1	3	**88**	2	1		2	1	1	1	
Maio	2	1		2	**82**	6	1	2	2	2		
Junho	3		1	1	1	**82**	4	2	2	2	1	1
Julho				2	3	6	**87**	1	1			
Agosto	1		2	2	1		2	**88**			1	3
Setembro	1	3	1	1	1	4			**76**	6	4	3
Outubro		3		3		3	1		5	**85**		
Novembro	1	1	2	3		2			6	1	**82**	2
Dezembro	3	2		1	1	3	1	3	1	5	4	**76**

Table 4: Confusion matrix for NN and directional feature set (NN-D).

Month	J	F	M	A	M	J	J	A	S	O	N	D
Janeiro	**74**	10		1	2	5	2		2	3		1
Fevereiro	8	**77**	4	1	3	2	1	2			2	
Março		2	**86**	1	3	2		3		1		2
Abril		1	1	**89**	5			1	1	1	1	
Maio	1		2	2	**78**	3	1	2	2	5	2	2
Junho	5		1	1		**69**	14	1	2	2	4	1
Julho	1	1	1		3	17	**65**	3	1	5	2	1
Agosto	2	3		3	1		2	**79**			1	9
Setembro	1	3		1	1	3		1	**64**	19	6	1
Outubro		2	1		2		1		13	**79**	2	
Novembro	2	1		2		3		1	4		**86**	1
Dezembro	4	3			1			8	3	5	2	**74**

Table 5: Confusion matrix for HMM scheme (HMM).

Month	J	F	M	A	M	J	J	A	S	O	N	D
Janeiro	**72**	13	1	1	5	2	2		2		1	1
Fevereiro	9	**75**		3	1	5	3				2	2
Março	3	3	**80**	3	6		1		3		1	
Abril	14			**82**	1		3					
Maio	10	1	6	2	**67**		10	1	1	1	1	
Junho	3	2		4	3	**75**	3	2	3	1		4
Julho	4	2		7	1	4	**80**	1		1		
Agosto	4			5			1	**80**		2		8
Setembro	1	1	1	5	2	5	4		**61**	8	4	8
Outubro	1	1		2			2	1	5	**87**	1	
Novembro	1	1		5	1				17	1	**70**	4
Dezembro	1	1		1	1	3	2	4	3	1	1	**82**

Table 7: Confusion matrix for HMM, NN-P and NN-D combined by multiplication.

Month	J	F	M	A	M	J	J	A	S	O	N	D
Janeiro	**86**	7			1	3	2		1			
Fevereiro	7	**87**	1	1	1	1	1			1		
Março		1	**92**	1	3		1	1			1	
Abril		1		**96**	2					1		
Maio				2	**91**	3	2	2				
Junho	1				2	**94**	1	1		1		
Julho	1			1	3	7	**88**					
Agosto	2		1	3	1	1		**91**				1
Setembro	1	2				1			**86**	9	1	
Outubro							1		3	**96**		
Novembro	1	1	1	1		1			5		**89**	1
Dezembro		1		1	2	1		2	2	2		**89**

5.4 Error Analysis and Discussions

To investigate the discriminative potential of classifiers, the tables 3, 4 and 5 are analyzed considering the classes grouped in clusters, defined by the presence of a common sub-string:

- Jan**eiro**-Fever**eiro**: The results show that most of the confusion for these classes happened between themselves for both classifiers.

- **Ma**rço-**Ma**io: Despite the similarity of the words, confusion is relatively low for this pair, likely due to the descender in *Março*. However, when the NN-D set are considered the word $Maio$ produces considerable confusion with several other classes. The same happens to the HMM classifier which yields high confusion occurs between $Maio$ and both $Janeiro$ and $Julho$.

- **Ju**nho-**Ju**l**ho**: The main confusions for these classes occur between themselves for the NNs classifiers, mainly for set NN-D. Considering the HMM scheme, the same is not true since this confusion occurs for several classes, specially between $Julho$ and $Abril$.

- **A**bril-**A**gosto: These words have no similarities, thus the occurrence of confusions between them is low. For NNs classifiers the word $Abril$, shows no confusion with any specific class. However, for the HMM there is an high level of confusion between $Abril$ and $Janeiro$. For $Agosto$ a high level of confusion with $Dezembro$ can be observed for both NN-D and HMM classifiers.

- Setem**bro**-Outu**bro**-Novem**bro**-Dezem**bro**: The main confusions for those words are among themselves, mainly between $Setembro$ and $Outubro$ for the NN-D classifier and a high confusion between $Novembro$ and $Setembro$ for the HMM.

This analysis shows that the main problems considering both classifiers occurs generally between classes in the same cluster, i.e., classes that present a common sub-string. The individual results shows that to this application the neural classifier presents a better input-output mapping than the HMM.

Comparing the confusion matrices for individual classifiers with that of the best combination result presented in Table 7, an improvement in recognition is noticeable for almost all classes. This is mainly true for classes $Fevereiro$, $Junho$, $Setembro$, $Outubro$ and $Dezembro$. This result shows that different classifiers when combined in a hybrid approach can yield an improved recognition rate. This can be observed considering combination of NNs alone and also occurs when combining NNs with HMMs. Examples of misclassification produced by the best combination of classifiers are shown in Figure 8.

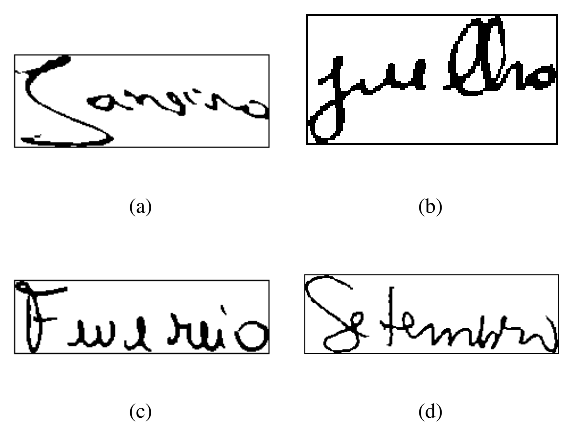

Figure 8: Examples of misclassification:(a) **Janeiro** classified as *Fevereiro*, (b) **Julho** classified as *Junho*, (c) **Fevereiro** classified as *Julho* and (d) **Setembro** classified as *Fevereiro*.

6 Conclusions

We have verified the efficiency of combining NN and HMM classifiers for handwritten word recognition. This evaluation was made using NN and HMM schemes with similar feature sets, therefore the results are influenced mainly by the classifiers performance.

The results show that the classifier with the best individual performance is the NN. The difference in performance is probably due to the distinct model generation technique used by the classifiers. While NNs train all classes and search an optimum solution for separating them, the HMM trains individual models for each class, without sharing information between them. However, the supremacy of NN only occurs in problems that have a limited lexicon, because a high number of classes is prejudicial to the training process, degrading the overall performance.

The main conclusion obtained is that the analyzed classifiers are complementary and the combination procedure proposed enhances their complementary. Our future work will focus on the analysis of rejection mechanisms for both classifiers.

Acknowledgements

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

References

[1] C. O. de A. Freitas. *Uso de Modelos Escondidos de Markov para Reconhecimento de Palavras Manuscritas*. Tese de Doutorado, Pontíficia Universidade Católica do Paraná, Brazil, 2001.

[2] D. Guillevic. *Unconstrained Handwriting Recognition Applied to the Processing of Bank Cheques*. PhD Thesis. Department of Computer Science at Concordia University, Canada, 1995.

[3] M. Côté. *Utilisation d'un Modèle d'Accès Lexical et de Concepts Perceptifs pour la Reconnaissance d'Images de Mots Cursifs*. Thèse de doctorat. École Nationale Supérieure des Télécomunications, France, 1997.

[4] A. L. Koerich, R. Sabourin, Y. Leydier and C. Y. Suen. "A Hybrid Large Vocabulary Handwritten Word Recognition System Using Neural Networks with Hidden Markov Models". *In proceedings of IWFHR'2002*, accepted for publication.

[5] J. H. Kim, K. K. Kim, C. P. Nadal and C. Suen. "A Methodology of Combining HMM and MLP Classifiers for Cursive Word Recognition". *In proceedings of ICPR'2000*, vol. 2, pp. 319-322, Barcelona-Spain, 2000.

[6] G. Zavaliagkos, Y. Zhao, R. Schwartz and J. Makhoul. "A Hybrid Segmental Neural Net/Hidden Markov Model System for Continuous Speech Recognition". *IEEE Trans. on Speech and Audio Processing*, 2(1):151-160, 1994.

[7] R. Schalkoff. *Pattern Recognition - Statistical, Structural and Neural Approaches*. Jonh Wiley & Sons, 1992.

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

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

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

[11] 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.

[12] 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.

[13] A. Zell et al. *SNNS - Stuttgart Neural Network Simulator, User Manual, Version 4.2*. University of Stuttgart, 1994.