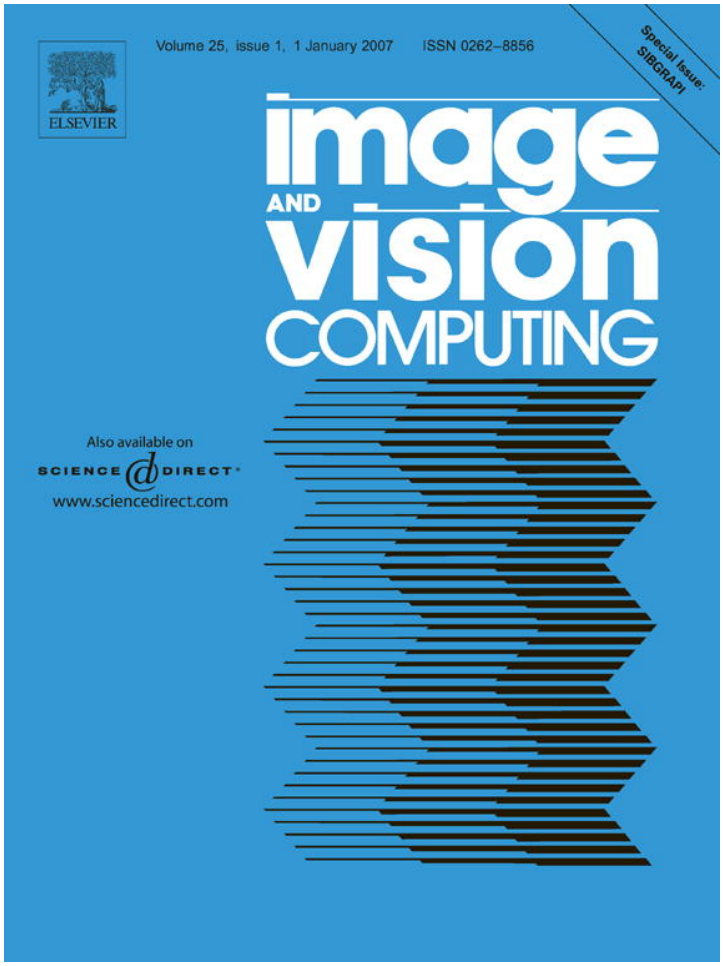

Available online at www.sciencedirect.com
SCIENCE DIRECT
Image and Vision Computing 25 (2007) 40–49
www.elsevier.com/locate/imavis

Methodology for the design of NN-based month-word recognizers written on Brazilian bank checks

Marcelo N. Kapp [a,*], Cinthia O. de A. Freitas [b], Robert Sabourin [a]

[a] École de Technologie Supérieure (ETS), 1100, Rue Notre Dame-Ouest, Montreal, Que., Canada H3C 1K3
[b] Pontifícia Universidade Católica do Paraná (PUCPR), Rua Imaculada Conceição, 1155, Prado Velho, 80215-901 Curitiba, PR, Brazil

Received 18 May 2004; received in revised form 2 September 2005; accepted 7 January 2006

Abstract

The study of handwritten words is tied to the development of recognition methods to be used in real-world applications involving handwritten words, such as bank checks, postal envelopes, and handwritten texts, among others. In this work, the focus is handwritten words in the context of Brazilian bank checks, specifically the months of the year, and no restrictions are placed on the types or styles of writing or the number of writers. A global feature set and two architectures of artificial neural networks (ANN) are evaluated for classification of the words. The objectives are to evaluate the performance of conventional and class-modular multiple-layer perceptron (MLP) architectures, to develop a rejection mechanism based on multiple thresholds, and to analyze the behavior of the feature set proposed in the two architectures. The experimental results demonstrate the superiority of the class-modular architecture over the conventional MLP architecture. A rejection mechanism with multiple thresholds demonstrates favorable performance in both architectures. The feature set analysis shows the importance of the structural primitives such as concavities and convexities, and perceptual primitives such as ascenders and descenders. The experimental results reveal a recognition rate of 81.75% without the rejection mechanism, and a reliability rate 91.52% with a rejection rate of 25.33%.

Keywords: Neural networks; Rejection; Feature selection; Handwritten word recognition

1. Introduction

The main objective of this work is the recognition of handwritten month words on Brazilian bank checks. To achieve this, we evaluate the performance of a conventional multiple-layer perceptron (MLP) feedforward architecture in relation to that of the class-modular MLP architecture for the recognition of the handwritten names of the months of the year in Brazilian Portuguese. This is an important task since it constitutes a sub-problem of bank check date recognition.

The development of an effective date processing system 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) (see Fig. 8), although this study deals with a limited lexicon of 12 classes: Jan**eiro**, Fever**eiro**, **Março**, Abril, **Maio**, **Junho**, **Julho**, Agosto, Set**embro**, Outu**bro**, Nov**embro**, and Dez**embro**. Some classes share a common sub-string (prefix and suffix), as shown in Fig. 1. We can also observe in Fig. 1 the similarity between the suffix and prefix of the words in the lexicon, a fact which increases the complexity of the recognition problem affecting the performance of the recognizer. This kind of problem has been raised in other works, as well as in [1,2], but no approach based on the intrinsic difficulties of the lexicon was proposed.

The recognizer works at a high level of representation: the feature vector extracted from the word images based on the feature set, as presented in Fig. 1. Our contribution, based on feature extraction, is the relationship between shape and feature set representation, as can be observed in the prefix cases in Fig. 1 (**jan**, **fev**, **ma**, **ju**; marked by traces) and suffix cases (see Fig. 1: **eiro**, **io**, **ço**, **ho** and **embro**; marked by squares and circles). Another aspect of our case is that, given that the first letter of the word is very important in the recognition process [3], the similarities among the words in the lexicons are increased, since some words have the same first letter (e.g. **junho** and **julho**). Other difficulty is the fact that the vowels 'a, e, i, o' exhibit the same behavior for the human reader, which requires different abilities to distinct the vowel 'u' from

* Corresponding author. Tel.: +55 41 271 1353; fax: +55 41 271 1669.
E-mail addresses: kapp@livia.etsmtl.ca (M.N. Kapp), cinthia.freitas@pucpr.br (C.O.A. Freitas), robert.sabourin@etsmtl.ca (R. Sabourin).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.01.005

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

the consonant 'w', for example, as observed by Schomaker and Segers [3] and as expected for the recognition system.

In general, handwriting recognition generates high-dimensional problems [4,5]. This work suggests a simple feature set which is enable recognition in relatively small number of dimensions. The power of artificial neural networks (ANNs) is their capacity to generate an area of decision. ANNs have also been used in other works [1,6,7] for the recognition of words in small lexicons.

Performances can differ, depending on the architecture—conventional or modular. Regarding K classes are involved in the classification module, we can naturally think of the classes as a target of modularity, where each class can be individually characterized, it leads us directly to the *class-modularity* concept [8].

In the class-modular concept, each class should be managed independently of the other classes, at least conceptually [8]. A similar idea related to the committee-of-networks concept is described in [9]. In this work the conventional and class-modular feedforward neural network architectures are evaluated based on a feature set, and global techniques are applied for the extraction of patterns. Fig. 2 shows an overview of the methodology considered here. The conventional and class-modular architectures were implemented and evaluated based on a feature set extracted from the word images after a preprocessing stage.

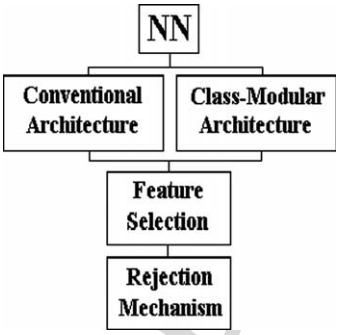

Fig. 2. Overview of the methodology.

Following the traditional pattern recognition approach, we divide the recognition task into two steps: first, a set of features is extracted from the images, and then a classifier computes the class-conditional probabilities based on these extracted features. So, the objective of feature extraction is to capture the most relevant and discriminatory characteristics of the object to be recognized. Accordingly, in our baseline system, the feature extraction algorithms use perceptual pattern recognition techniques, while the classification is based on a neural network approach.

In order to select a subset of original features by reducing irrelevant and redundant ones, feature selection algorithms have been applied [10]. In the absence of such algorithms, large training sets are mandatory. Feature selection methods attempt to find reduced feature sets, which minimizes the probability of error. Most of these methods use evaluation functions and search algorithms to achieve their objective. The evaluation functions measure how good a specific subset is in discriminating among classes, and can be divided into two categories: filters and wrapper methods [10]. Filters measure the relevance of feature subsets independently of the classifier, whereas wrappers use the classifier's performance as the evaluation function. Search algorithms, by contrast, are responsible for driving feature selection using a specific strategy, e.g. branch-and-bound, stepwise and genetic algorithm, among others [10]. For this purpose, our methodololy includes a feature selection study based on the wrapper/hill climbing approach.

Usually, recognition systems apply a global decision module, which decides either to accept the recognition result or to reject it. In classification, a pattern is considered ambiguous if it cannot be reliably assigned to a class, whereas a pattern assigned low confidence for all hypothesized classes can be treated as an outlier.

The purpose of a rejection mechanism is to minimize the number of recognition errors for a given number of rejects. A simple rejection scheme involves the rejection of an image with a global probability lower than a determined threshold, as denoted by Chow's rule $O_i < T$ [11].

Now, consider a simple one-dimensional classification task with two data classes ω_1 and ω_2 characterized by Gaussian distributions. Fumera et al. [12] hypothesized that significant errors affect the estimated probabilities in the range of feature values in which the two classes 'overlap'. It is easy to see, as

demonstrated in [12], that in this case the application of global threshold T to the estimated probabilities does not make it possible to obtain both the optimal decision regions and the reject region. In response, Fumera et al. suggest the use of N class-related reject thresholds (CRTs).

In this paper, we investigate the effects of estimate errors on Chow's rule and CRTs based on multiple reject thresholds related to the data classes, as shown in Tables 6–8. The reported experimental results show that such class-related reject thresholds provide a better error-reject trade-off than that in Chow's rule.

This paper is organized as follows. Section 2 describes the feature set extracted from the word images. Sections 3 and 4 introduce the conventional and class-modular architectures, respectively. Section 5 presents the month–words database constructed from Brazilian bank checks, and in Section 6 the experimental results are provided, with some analysis and discussion. Section 7 presents the feature selection based on the wrapper-modified/hill climbing approach. In Section 8, the reject option with multiple thresholds is summarized. In Section 9, the experimental results are presented, with some analysis and discussion. In Section 10, concluding remarks and plans for future work are provided.

2. Feature extraction

Feature extraction plays an important role in handwriting recognition systems, as described in [5]. Thus, all the studies in pattern recognition, and more specifically in handwritten word recognition have, as one of their relevant points, feature set selection through which the most relevant and discriminatory characteristics of the object to be recognized are selected.

In this work, perceptual features [5] and characteristics based on concavities/convexities (and others) are explored for the recognition of the handwritten names of the months of the year in Brazilian Portuguese. Basically, there are a number of occurrences of such features. Since, these discrete primitives alone do not make the recognition system more robust [4], therefore, a zoning mechanism was added to the feature set during extraction of the primitives. This zoning mechanism provides a location to the prefix and suffix in the word image.

The zoning is used in two areas separated by the center of gravity of the word: *left-area* = *prefix* and *right-area* = *suffix*, as shown in Fig. 3. This is done because, in each midfield, the occurrence of some features gives more useful information to the pattern classifier and provides a high-level representation in terms of prefix and suffix. The zoning is based on two areas and represents an initial attempt at emphasizing the isolated feature extraction of these sub-strings that result in conflicts in classifications like 'eiro', from Jan**eiro** and Fever**eiro**, or 'embro', from Set**embro**, Nov**embro** and Dez**embro**.

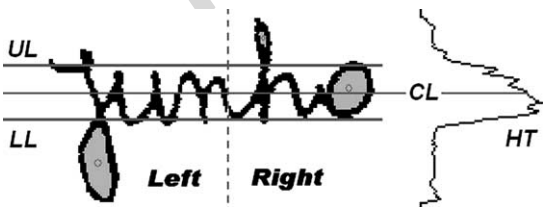

Fig. 3. Example of zoning mechanism and area detection.

The ascending and descending zones are computed taking into account the upper line (UL) and lower line (LL) lines. UL and LL are based on a maximum horizontal projection histogram of black–white transitions, establishing the central line (CL) [13], as presented in Fig. 3.

The feature set can be described as follows:

- Number of loops on the *left/right-areas* (NLL = 1 and NLR = 2), Fig. 3;
- Number of concave semicircles on the *left/right-areas* (NSCVL = 2 and NSCVR = 3), Fig. 4a;
- Number of convex semicircles on *left/right-areas* (NSCXL = 1 and NSCXR = 3), Fig. 4b. The concavities and convexities are only extracted in the body of the word based on the skeleton by mathematical morphology;
- Number of crossing-points on the *left/right-areas* (NCPL = 1 and NCPR = 0), Fig. 4c;
- Number of branch-points on the *left/right-areas* (NBPL = 5 and NBPR = 6), Fig. 4d;
- Number of end-points on the *left/right-areas* (NEPL = 5 and NEPR = 2), Fig. 4e;
- Number of crossings between the stroke and the horizontal axis (NCH = 13), Fig. 4f;
- Number of ascenders on the *left/right-areas* (NAL = 0 and NAR = 1);

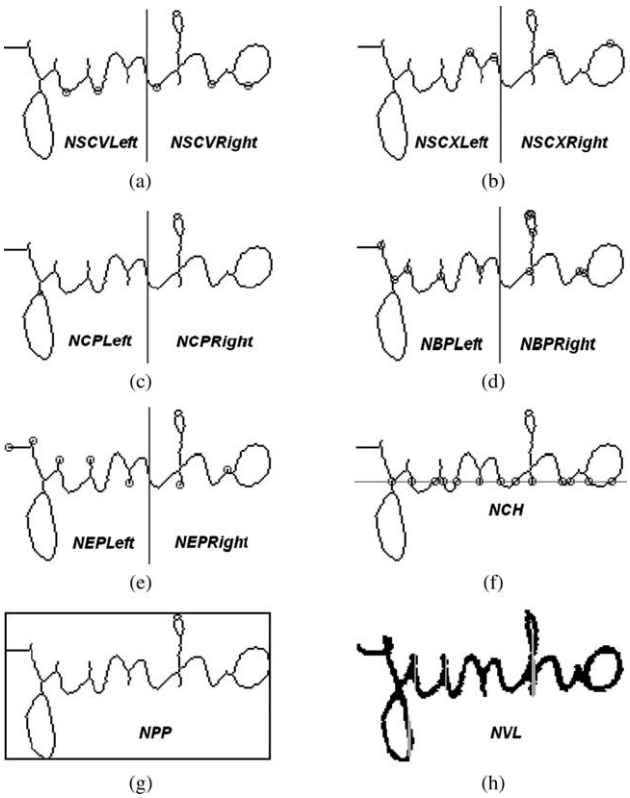

Fig. 4. Feature extraction: (a) concave semicircles; (b) convex semicircles; (c) crossing-points; (d) branch-points; (e) end-points; (f) crossings between the stroke and the horizontal axis; (g) proportion of black pixels; (h) vertical lines.

- Number of descenders on the *left/right-areas* (NDL = 1 and NDR = 0);
- Proportion of black pixels in relation to white ones (NPP = 0.968610), Fig. 4g. The pixel proportion is part of the surface in relation to the context of the word (NPP). A bounded box is used, and the proportion can be obtained as follows: $npp = (tp - tpp/tp)$, computed inside the bounded box, where tp is the total number of pixels inside the bounded box and tpp is the total number of black pixels in the word stroke;
- Number of vertical lines (NVL = 4), Fig. 4h;
- Number of horizontal lines (NHL = 0);
- Number of ascenders with loop on the *left/right-areas* (NALL = 0 and NALR = 1);
- Number of descenders with loop on the *left/right-areas* (NDLL = 1 and NDLR = 0).

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

3. Conventional neural network architecture

The MLP has been used extensively in implementing the K-classification module for word recognition. One of the distinct properties of the conventional MLP architecture is that all the K classes share one large network [8], as shown in Fig. 5, where each O_i represents a certain class.

The essential task in designing a character recognition system is to choose a feature type with a good discriminative power. As well, the network should divide the K class regions well in the chosen feature space.

However, determining the optimal decision boundaries for the K-classification module for word recognition in a feature space with a very large number of dimensions is very complex, and can seriously limit the recognition performance of the character recognition system using MLP [8,14].

In particular, when the training set is not large enough relative to the classifier size (i.e. the number of free parameters in the classifier), a convergence problem can occur [14]. The size of the training set directly influences the performance of any classifier trained nonparametrically (e.g. as a neural network). This class of learning machines requires a great deal of data for appropriate training, because there are no prior assumptions about the data. It is important to know how the requirement on the size of the training set is scaled as a function of the size of the network for a given mapping precision [9].

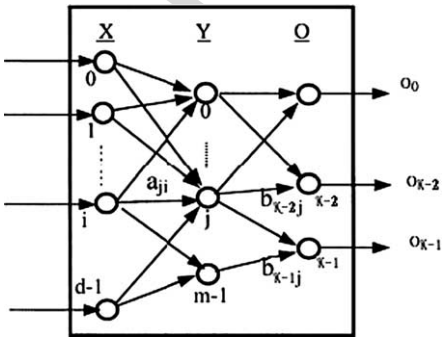

Fig. 5. Conventional architecture where K classes are intermingled [8].

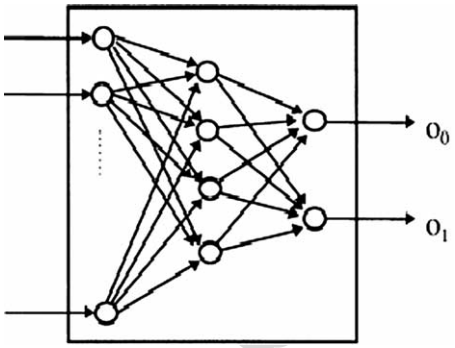

Fig. 6. Class-modular architecture: sub-network [8].

4. Class-modular neural network architecture

A single task is decomposed into multiple subtasks and each subtask is allocated to an expert network. In this paper, as well as in [8], in the class-modular classification, the K-classification problem is decomposed into K 2-classication subproblems, one for each of the K classes. A 2-classification subproblem is solved by the 2-classifier specifically designed for the corresponding class.

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 Fig. 6, 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 constructed by K sub-networks is shown in Fig. 7.

5. Database

The database used is composed of the names of the months of the year, and the data were collected by the UFPB (Federal University of Campina Grande-Paraíba-Brazil) (for more details, see [1]). In total, there are 6000 word images in the database, with 500 of each class.

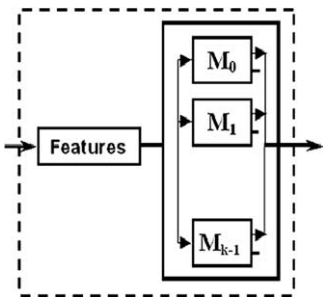

Fig. 7. Class-modular architecture: whole network with M modules [8].

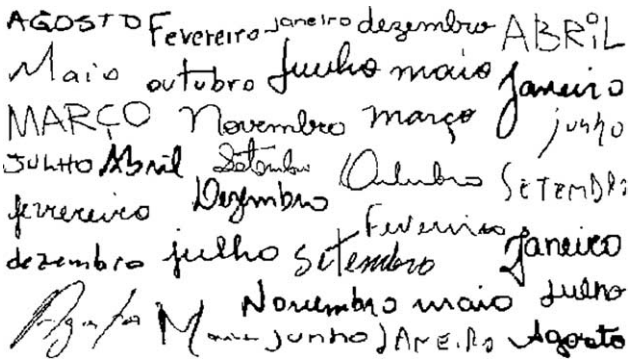

Fig. 8. Sample images from the database.

Table 1
Information on the distribution of the writing styles in the sets

Writing styles	Training (%)	Validation (%)	Test (%)
Pure cursive	7097	7125	7292
Uppercase	583	575	483
Spaced discrete	1206	11	908
Mixed	1114	1200	1317

All the images have been preprocessed, i.e. the baseline skew and slant were corrected, reducing writing variability (different writing styles and particular writing characteristics). For the experiments, the database was randomly divided into three data sets: training set (60%), validation set (20%), and test set (20%). Fig. 8 shows sample images from the database and the Table 1 shows the distribution of the writing styles found in the database.

6. First experimental results

This section presents the results obtained with the conventional and class-modular MLP architectures, without the use of rejection schemes.

6.1. Conventional architecture results

Conventional MLP is composed of 24 nodes in the input layer, 45 nodes in the hidden layer, and an output layer with 12 nodes. Validation sets were employed in order to avoid over-training. The stop criterion is the increase in the error evaluated in the validation set. All the classes were trained together. The class that presents the maximum output value is the class considered to be recognized. The recognition rate obtained for the conventional architecture is 77.08%. The confusion matrix for the test set is shown in Table 2. We observed in this matrix that there was confusion among the words sharing the same prefix or suffix. The **Junho–Julho** case is a good example in which a prefix and suffix are shared between these words, resulting in the symmetrical confusion shown in Table 2.

6.2. Class-modular MLP architecture results

In class-modular MLP, each of K 2-classifiers is trained independently of the other classes using the training and validation set. The backpropagation algorithm was used in each of the 2-classifiers in the same way as in conventional MLP. To train 2-classifiers for each word class ($K=12$), we re-organize the samples in the original training and validation set into K-two groups, $Z0$ and $Z1$, such that $Z0$ contains the samples from the current class and $Z1$ all the others, taking into account

Table 2
Confusion matrix for the conventional architecture

Des/Obt	J	F	M	A	M	J	J	A	S	O	N	D	Rec.(%)
Janeiro	**77**	5	4	0	1	1	1	2	1	2	5	1	77
Fevereiro	6	**87**	0	0	0	0	0	0	2	4	0	1	87
Março	5	1	**74**	2	12	2	2	1	0	1	0	0	74
Abril	0	1	5	**84**	4	3	2	1	0	0	0	0	84
Maio	1	2	9	5	**77**	0	2	1	0	2	1	0	77
Junho	2	2	1	0	4	**77**	10	0	1	3	0	0	77
Julho	1	1	2	3	4	10	**73**	3	0	3	0	0	73
Agosto	6	2	5	4	0	1	2	**73**	0	0	2	5	73
Setembro	2	8	0	1	0	1	0	0	**71**	7	6	4	71
Outubro	3	3	3	0	0	2	1	0	9	**76**	3	0	76
Novembro	0	2	6	0	0	1	0	0	3	1	**82**	5	82
Dezembro	4	6	0	0	1	1	1	2	4	1	6	**74**	74
											Recognition		**77,08%**

Table 3
Confusion matrix for the class-modular architecture

Des/Obt	J	F	M	A	M	J	J	A	S	O	N	D	Rec. (%)
*Jan**eiro***	**83**	8	2	0	0	2	0	0	0	2	1	2	83
*Fever**eiro***	5	**83**	1	1	1	0	0	1	3	1	2	2	83
*Mar**ço***	3	3	**75**	5	10	0	0	0	0	2	2	0	75
Abril	1	1	1	**93**	2	0	2	0	0	0	0	0	93
*Ma**io***	1	0	10	5	**80**	2	1	0	0	0	1	0	80
*Jun**ho***	1	3	0	0	5	**84**	4	0	1	2	0	0	84
*Jul**ho***	1	0	0	6	4	9	**76**	0	0	3	1	0	76
Agosto	2	4	2	3	0	3	0	**78**	0	3	3	2	78
*Set**embro***	1	10	0	0	0	0	0	0	**73**	6	8	2	73
*Outu**bro***	4	4	2	0	0	0	0	0	4	**85**	1	0	85
*Nov**embro***	3	2	0	0	0	0	0	0	4	1	**89**	1	89
*Dez**embro***	3	5	0	0	0	2	1	1	0	0	6	**82**	82
											Recognition		**81,75%**

the a priori probability for each class. To recognize the input word patterns, the class decision module takes only the values of O_0 and uses the simple winner-takes-all scheme to determine the final class (see Fig. 7).

A conventional network sees each of the training instances once per epoch. However, in the case of modular network, each subnetwork sees each training instance once per epoch, so the whole network sees each sample K times per epoch [8]. The recognition rate obtained for the class-modular architecture was 81.75%, demonstrating a significant improvement. The confusion matrix for this experiment is presented in Table 3. We observed in this matrix that there was confusion among the words sharing the same suffix, such as Jan**eiro**–Fever**eiro**. Note that the class-modular approach resolves some of the confusions previously presented in Table 2 (**Jun**ho and **Jul**ho).

6.3. Discussions

Table 4 summarizes the experimental results obtained in this work and in some other studies with the same database [1]. Observe that ANNs in general obtained better results than hidden Markov models (HMM), when a similar feature set based on perceptual features is applied. The Conventional network obtained almost the same recognition rate as HMM, but it was improved by the class-modular network. The HMM and class-modular approaches are based on the same principle: obtaining an individual model or network for each word class. The difference between these approaches is in the training stage: the HMM models for each word class contain no information about the other classes. This does not occur during the class-modular network training stage.

Two concluding remarks can be made based on the experimental results, as follows:

- The class-modular network was superior in terms of convergence over the conventional network (according to the monitoring of the MSE—mean-square error); and
- The class-modular network was superior in terms of recognition capability than the conventional network, as it was in [8].

7. Feature selection

In supervised machine learning, a learning algorithm is applied to a selection problem of some feature subsets on which to focus, while ignoring the rest [15]. There are two approaches that can be followed at this point: filters or wrapper. The filter approach selects features using a preprocessing step, ignoring the induction algorithm. The main disadvantage of this is that it totally ignores the effects of the selected feature set in the performance of the induction algorithm. In [15], the authors present some algorithms based on the filter approach, which are the algorithms Focus and Relief, and a filter based on Decision Trees.

In the wrapper approach [16], the feature selection algorithm works like a 'packer' of the induction algorithm. The feature subset drives a search for a good subset using its own induction algorithm as part of the feature subset evaluation function. The idea behind this approach is the following: the

Table 4
Comparison of word recognition results

Set	Recognition (%)
HMM [1]	75.90
Conventional architecture	77.08
Class-modular MLP	81.75

induction algorithm is considered as a black box, that is, knowledge of this algorithm is not necessary, just of its interface. The induction algorithm is run on a database, usually a validation set, with different data feature sets removed.

In the proposed methodology the approach adopted for feature selection is the wrapper approach. This choice was based mainly on the fact that, in this phase, the architectures of the trained ANNs were already available, that is, our induction algorithms were already prepared, and there was still an interest in investigating the effects of the proposed feature set on the classifiers. No preprocessing of the data was necessary with the retreat of some features and retraining of the classifiers, as is the case with the filter approach. Outlines of the wrapper approach can be found in [16].

Since, the size of the search space for n features is $O(2)^n$, it is impractical to look exhaustively for the space whole, unless n is small. The objective of the search is rather to find the state set with the highest evaluation, using a heuristic function to guide it.

In the present work, the search process adopted for the approach wrapper is hill climbing. As Skalak did in [17], hill climbing is used in a random way which continues for a specific number of cycles. It is denominated, due to random mutation, to be executed in aleatory fashion. The basic idea of this algorithm is described in [17] as follows:

(1) Choose a binary string. Call this string *best-evaluated*;
(2) Mutate a bit chosen at random in the *best-evaluated* string;
(3) Compute the fitness of the mutated string. If the fitness is strictly greater than the fitness of the *best-evaluated* string, then set the mutated string as *best-evaluated*.
(4) If the maximum number of iterations has been performed, return *best-evaluated*; otherwise, go to step 2.

The wrapper approach drives a search in the space of possible parameters. A search requests a state space, as an initial, an end conduction, and a search process. The organization of the search space used in this methodology is represented in the following way, for each features set there is a bit which indicates the presence (1) or the removal (0) of a referred feature. Operators determine the conectivity among the states, as removal operators and addition of features of a state (0 or 1).

When a feature was absent, we substituted its average in the classifier training set in the ANN inputs, i.e. creating a wrapper-modified [18,19]. It is important to remember, that the idea at the moment is to apply the ANNs already trained, and observe the feature set. Feature selection using the validation set is shown in Fig. 9 for both the conventional and class-modular architectures.

The wrapper/hill climbing approach is also applied on each one of the modules of the class-modular architecture, thereby achieving class-dependent feature selection. The results obtained for each isolated module of the class-modular architecture on the validation set are presented in Fig. 10.

Table 5 present experiments carried out with the initial feature set and the new subsets obtained through the use of the validation set and applied on the test set.

7.1. Discussions

From the completed experiments, it was noted that, in both architecture using the validation and test set, there is not a significant improvement in recognition rates with fewer features (see Table 5).

Already, the investigation of features for each module, shown in Fig. 10 as the feature subsets is important for the

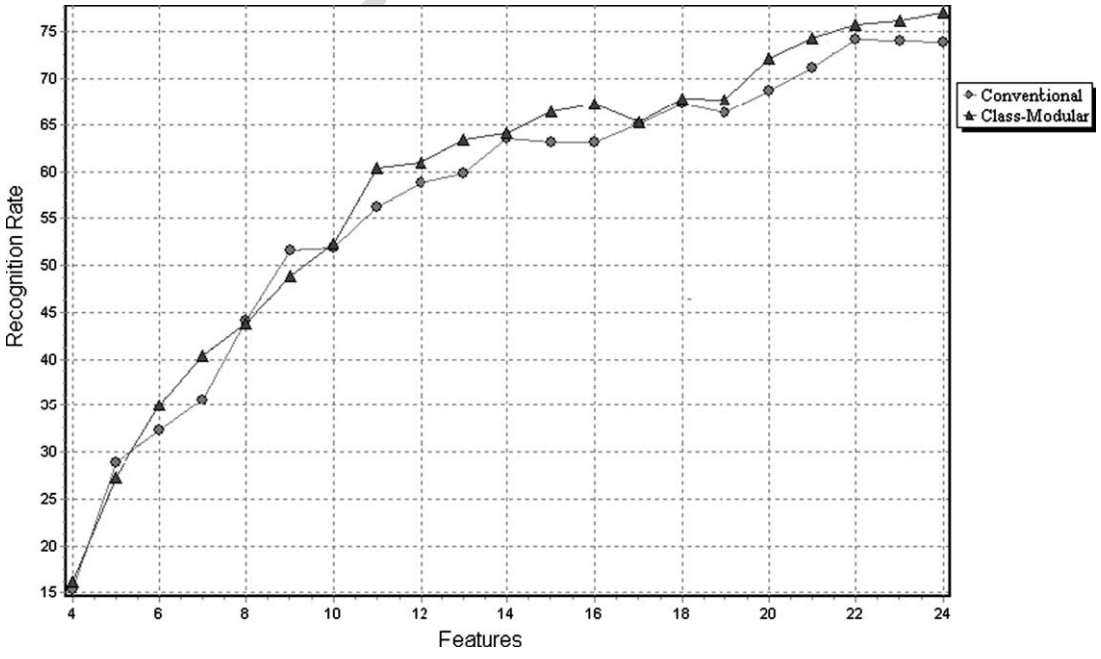

Fig. 9. Results of the wrapper/hill climbing approach for the conventional and class-modular architectures on the validation set.

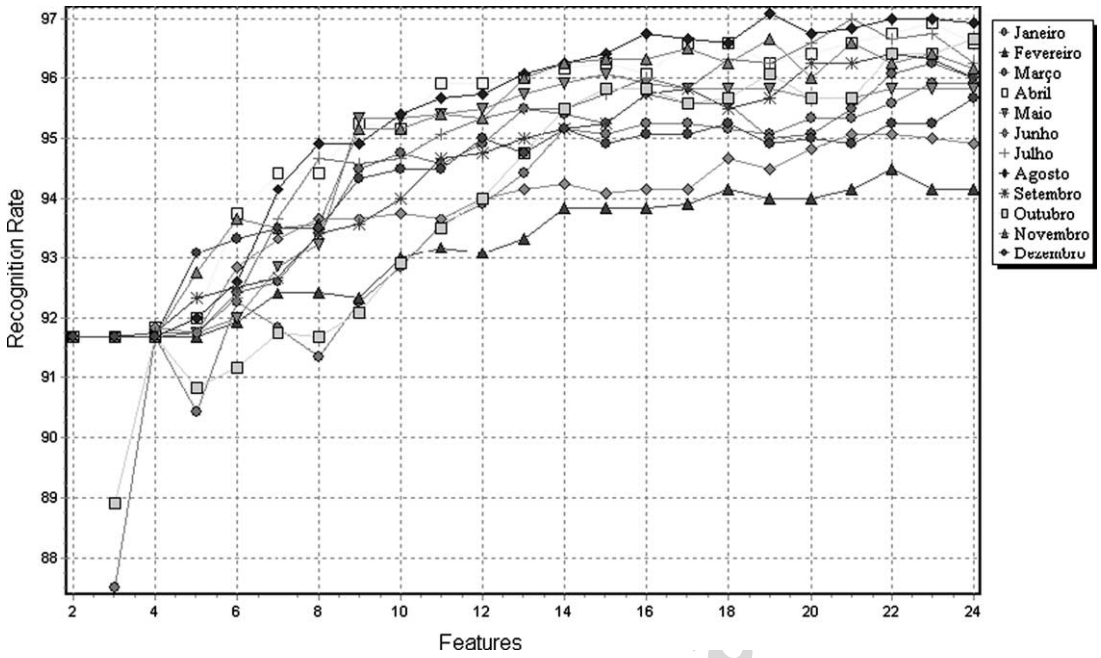

Fig. 10. Results of the wrapper/hill climbing approach for each isolated module of the class-modular architecture on the validation set.

separation of one particular class from all the remaining ones. When the subsets of each module are organized in such a way that they work together, there is difficulty in separating them, and the performance of the system is committed again.

However, in agreement with Raman and Ioerger in [20], there are two main problems in the development of feature selection architectures. The first, is that finding a minimum feature set is a NP-complete problem, and the second, is that the feature set that provides the maximum increase in precision in the classification does not necessarily define a consistent hypothesis, as problem of relevance and irrelevance of examples described in [15,20]. The main conclusion, therefore, is that more features (such as, topological and geometric features) are needed for this word recognition system.

8. Reject option with multiple thresholds

An K-class classifier is designed to subdivide the feature space into K decision regions D_i, $i=0,\ldots,K-1$, such that the patterns of class w_i belong to region D_i. According to statistical pattern recognition theory, such decision regions are defined so as to maximize the probability of correct recognition, commonly referred to as classifier accuracy (Eq. (1))

$$\text{Accuracy} = P(\text{correct}) = \sum_{i=0}^{K-1} \int_{D_i} p(x|w_i)P(w_i)\mathrm{d}x \tag{1}$$

and, consequently, to minimize classifier error probability (Eq. (2)):

$$P(\text{error}) = \sum_{i=0}^{K-1} \int_{D_i} \sum_{\substack{j \neq i, \\ j=0}}^{K-1} p(x|w_j)P(w_i)\mathrm{d}x \tag{2}$$

To this end, the so-called Bayes decision rule assigns each pattern x to the class for which the a posteriori probability $P(w_i|x)$ is at its maximum. An error probability lower than that provided by the above Bayes rule can be obtained using the so-called 'reject' option [12], by which the patterns that are the most likely to be misclassified are rejected (i.e. they are not classified). Therefore, a trade-off between error and reject is mandatory. The formulation of the best error-reject trade-off and the related optimal reject rule was given by Chow [11].

A careful analysis of Chow's work reveals that his reject rule provides the optimal error-reject trade-off only if the *a posteriori* probabilities are known exactly. Therefore, in Fumera et al. [12], the authors suggest the use of multiple reject thresholds to obtain the optimal decision and reject regions, even if the a posteriori probabilities are affected by errors.

It is easy to see that if such thresholds are applied to the estimated probabilities it becomes possible to obtain both the optimal decision regions and the rejection region. The use of K class-related reject thresholds (CRTs) can provide a better error-reject trade-off than Chow's rule [12]. In particular, under the assumption that the a posteriori probabilities are affected by significant errors, the authors have proved in [12] that, for any reject rate R, such values of the CRTs $T_0,\ldots T_{K-1}$ exist such that the accuracy of the corresponding classifier $A(T_0,\ldots T_{K-1})$

Table 5
Recognition rates and features set applied on the test set

Architecture	Number of features	Validation rec (%)	Test rec (%)
Conventional	24	7392	7708
Conventional	22	7417	7458
Class-modular	24	7708	8175
Class-modular	Class-dependent	Class-dependent	8016

is equal to or higher than the accuracy $A(T)$ provided by Chow's rule, see Eq. (3):

$$\forall R\ \exists\ T_0,T_1,\ldots,T_{K-1}\colon\ A(T_0,T_1,\ldots,T_{K-1}) \geq A(T) \tag{3}$$

The authors therefore proposed in [12] the following reject rule, named the CRT rule, for a classification task with K data classes, which are characterized by estimated a posteriori probabilities $\hat{P}(w_i|x), i = 0,\ldots,K-1$. A pattern x is rejected if

$$\max_{k=0,\ldots,K-1} \hat{P}(w_k|x) = \hat{P}(w_i|x) < T_i \tag{4}$$

while it is accepted and assigned to class w_i if:

$$\max_{k=0,\ldots,K-1} \hat{P}(w_k|x) = \hat{P}(w_i|x) \geq T_i \tag{5}$$

The CRTs take on values in the range [0,1]. It is worth noting that, by analogy with Chow's rule, the values of the CRTs must be estimated according to the classification task at hand in real applications. In our experiments, such as in [12], we considered the usual error-reject requirement of real pattern recognition applications, i.e. obtaining the highest accuracy and an error rate below a given value $E_{\min}$. Accordingly, the CRT values were estimated by solving the following constrained minimization problem (Eq. (6)):

$$\begin{cases} \max_{T_0,\ldots,T_{K-1}} A(T_0,\ldots,T_{K-1}) \\ E(T_0,\ldots,T_{K-1}) \leq E_{\min} \end{cases} \tag{6}$$

It is worth noting that, according to (Eq. (3)), for any given $E_{\min}$, the CRT values obtained as solutions of the above maximization problem provide an accuracy equal to or higher than that in Chow's rule. Therefore, [12] takes on a finite number of values in the range [0,1] and (Eq. (5)) represents a constrained maximization problem the 'target' and 'constraint' functions of which are discrete valued functions of continuous variables. Our algorithm takes into account that $E(T_0,\ldots,T_{K-1})$ is an increasing function of the variables $T_0,\ldots,T_{K-1}$ (i.e. the number of rejected patterns cannot decrease for increasing CRT values) and also assumes that $A(T_0,\ldots,T_{K-1})$ is an increasing function $T_0,\ldots,T_{K-1}$.

8.1. Obtaining and testing multiple thresholds using conventional and class-modular architectures

The basic idea is to solve (Eq. (6)) iteratively. We start with CRT values, which provide a reject rate equal to zero, and at each step increase the value of one of the CRTs in order to increase accuracy until the reject rate exceeds the value $E_{\min}$. It is worth noting that our algorithm does not guarantee an optimal solution to (Eq. (6)). Nevertheless, the experimental results reported below show that it affords CRT values which provide a better error-reject trade-off than that in Chow's rule for handwritten word recognition.

To obtain the thresholds for the conventional architecture, several validation subsets must be used for each class of the problem, and each subset is submitted in a conventional architecture, as shown in Fig. 11, which illustrates the way in which thresholds for a class w_0 are obtained, where the validation set possesses only samples of class w_0 and where the corresponding NN output is y_0. Note that the estimation of all thresholds ends when the rejection rate reached is greater than 20%. The application of the thresholds with a test set is based on (Eq. (4)) where a pattern x is accepted or rejected.

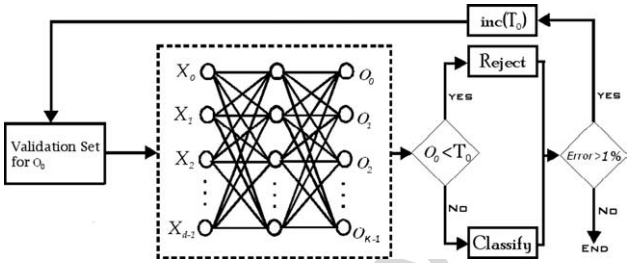

Fig. 11. Obtaining threshold T_0 for class w_0 using the conventional architecture.

The way the thresholds were computed for the class-modular architecture is based on the previous procedure used for a conventional architecture. The idea is to obtain the threshold T_0 for a class w_0 using the class-modular architecture, and finally the application of the thresholds with the test set, where the pattern x is accepted or rejected, is based on (Eq. (4)).

9. Final experimental results

This section presents the results obtained with the conventional and class-modular MLP architectures using rejection schemes applying the whole feature set in these experiments. Tables 6–8 present recognition (Rec.), rejection (Rej.) and reliability (Rel.) rates at the three errors levels (1, 2, and 5%). All rejection thresholds used for the experiments on the test set (Table 8) were estimated on the validation set.

9.1. Discussions

The results obtained, in terms of recognition, with a rejection rate equal to 0% for both the conventional and class-modular architectures were 77.08 and 81.75%, respectively. We have observed in Tables 6 and 7 that, as in [12], all thresholds obtained

Table 6
Chow's rule and validation set

Error rates (%)	Conventional			Class-modular		
	Rec.	Rej.	Rel.	Rec.	Rej.	Rel.
1	21.34	77.66	95.52	23.00	76.00	95.83
2	26.75	71.25	93.04	29.75	68.25	93.70
5	40.25	54.75	88.95	46.08	48.92	90.21

Table 7
Multiple thresholds and validation set

Error rates (%)	Conventional			Class-modular		
	Rec.	Rej.	Rel.	Rec.	Rej.	Rel.
1	58.75	40.25	98.16	61.33	37.67	98.29
2	66.08	31.92	96.81	68.33	29.67	96.97
5	72.42	22.58	93.51	75.50	19.50	93.73

Table 8
Multiple thresholds estimated on the validation set and applied on the test set

Rates	Conventional			Class-modular		
	1%	2%	5%	1%	2%	5%
Rec.	57.17	67.00	75.42	68.33	74.17	79.75
Rej.	34.00	20.42	7.33	25.33	17.08	6.92
Error	8.83	12.58	17.25	6.33	8.75	13.33
Rel.	86.62	84.19	81.38	91.52	89.45	85.68

through CRTs were better than those obtained with Chow's rule for a conventional architecture MLP. Moreover, we have also observed that a class-modular architecture is superior to conventional one, see Table 7. These facts can be partly explained by a best mapping between the feature space and the classes provided by the class-modular architecture even prior to the rejection process, but mainly by the idea of multiple thresholds, obtained in local way in each module for each class and not globally as in Table 6. This superiority also extends to the test set, see Table 8. However, the error rates differ from those observed with the validation set due to the intrinsic variability of writing styles (pure cursive, uppercase, spaced discrete and mixed) and a different frequency of occurrence of these writing styles in both data sets.

10. Conclusions

The results indicate that this research is quite promising and proves to be worthy of further investigation of the class modularity paradigm. Consideration must be given to large-set classifications in order to test the effect of the number of classes on recognition capability (for example: legal amounts from Brazilian bank checks, with a lexicon composed of 39 words).

We proposed and implemented a new smaller feature set than that presented in [1]. It generates fewer parameters to be estimated in the ANN, decreasing the complexity computation without loss of recognition performance. We also observed that the class-modular network was superior in terms of convergence and recognition capability to the conventional network, such as in [8].

The conventional architecture has a *rigid* structure composed of an unstructured black box in which all the K classes intermingle. The modules cannot be modified or optimized locally for each class. However, the disadvantages of the class-modular architecture are that the training and validation set for assisting each class, as described in Section 6.2, and the training of K networks for the classes of the problem must be reorganized.

Based on the analysis of Tables 6–8, we can say that a better error-reject trade-off can be obtained with the rejection rule proposed in [12] mainly because it represents a local search of each class and not a global search as in [11]. Accordingly, this rejection mechanism also behaves better in the class-modular architecture.

The feature selection analysis presented in Section 7 shows the importance of the use of the structural and perceptual primitives selected in the proposed feature set, and it point for the additional features, because, in general, the feature set already used has been demonstrated to be fully necessary.

References

[1] J.J.O. Júnior, J.M. de C. Carvalho, C.O.A. Freitas, R. Sabourin, Evaluating nn and hmm classifiers for handwritten word recognition, in: Proceedings of the XV Brazilian Symposium on Computer Graphics and Image Processing, IEEE, 2002, pp. 210–217.
[2] M. Morita, R. Sabourin, F. Bortolozzi, C.Y. Suen, Segmentation and recognition of handwritten dates, in: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition, IEEE Computer Society Press, Los Alamitos, CA, 2002, pp. 105–110.
[3] L. Schomaker, E. Segers, A method for the determination of features used in human reading of cursive handwriting, in: Sixth International Workshop on Frontiers in Handwriting Recognition, IEEE Computer Society Press, Los Alamitos, CA, 1998, pp. 157–168.
[4] O.D. Trier, A.K. Jain, T. Taxt, Feature extraction methods for character recognition—a survey, Pattern Recognition 29 (4) (1996) 641–662.
[5] S. Madhvanath, V. Govindaraju, The role of holistic paradigms in handwritten word recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2) (2001) 149–164.
[6] M. Côté, Utilisation d'un modèle d'accès lexical et de concepts perceptifs pour la reconnaissance d'images de mots cursifs, PhD thesis, École Nationale Supérieure des Télécommunications, France, 1997.
[7] D. Ollivier, Une approche économisant les traitements pour reconnaître l'écriture manuscrite: application à la reconnaissance des montants littéraux de chèques bancaires, PhD thesis, Université de Paris XI Orsay, France, 1999.
[8] I.-S. Oh, C.Y. Suen, A class-modular feedforward neural network for handwriting recognition, Pattern Recognition 35 (2002) 229–244.
[9] J.C. Principe, N.R. Euliano, W.C. Lefebvre, Neural and Adaptive Systems: Fundamentals Through Simulations, Wiley, New York, 1999.
[10] L. Molina, L. Belanche, A. Nebot, Feature selection algorithms: a survey and experimental evaluation, in: International Conference on Data Mining, IEEE Computer Society Press, Los Alamitos, CA, 2002, pp. 306–313.
[11] C.K. Chow, On optimum error and reject tradeoff, IEEE Transactions on Information Theory 16 (1) (1970) 41–46.
[12] G. Fumera, F. Roli, G. Giacinto, Reject option with multiple thresholds, Pattern Recognition 33 (2000) 2099–2101.
[13] C.O.A. Freitas, F. Bortolozzi, R. Sabourin, Handwritten isolated word recognition: an approach based on mutual information for feature set validation, in: International Conference on Document Analysis and Recognition, International Association of Pattern Recognition, 2001, pp. 665–669.
[14] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37.
[15] R. Kohavi, G.H. John, Wrappers for feature subset selection, AIJ Special Issue on Relevance (1997) 1–43.
[16] R. Kohavi, Feature subset selection as search with probabilistic estimates, AAAI Fall Symposium on Relevance (1994) 122–126.
[17] D.B. Skalak, Prototype and feature selection by sampling and random mutation hill climbing algorithms, in: Eleventh International Machine Learning Conference (ICML-94), 1994, pp. 293–301.
[18] J. Moody, J. Utans, Principled architecture selection for neural networks: applications to corporate bond rating prediction, Advances in Neural Information Processing Systems 4 (1992) 683–690.
[19] L. Oliveira, R. Sabourin, F. Bortolozzi, C. Suen, A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition, International Journal of Pattern Recognition and Artificial Intelligence 17 (6), (2003), 903–929.
[20] B. Raman, T.R. Ioerger, Enhancing learning using feature and example selection, Master's thesis, Department of Computer Science, Texas A&M University, College Station, TX, USA, 2003. URL: citeseer.nj.nec.com/raman03enhancing.html