Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA, July 31 – August 5, 2011

A MLP-SVM Hybrid Model for Cursive Handwriting Recognition

Washington W. Azevedo and Cleber Zanchettin

Abstract— **This paper presents a hybrid MLP-SVM method for cursive characters recognition. Specialized Support Vector Machines (SVMs) are introduced to significantly improve the performance of Multilayer Perceptron (MLP) in the local areas around the surfaces of separation between each pair of characters in the space of input patterns. This hybrid architecture is based on the observation that when using MLPs in the task of handwritten characters recognition, the correct class is almost always one of the two maximum outputs of the MLP. The second observation is that most of the errors consist of pairs of classes in which the characters have similarities (e.g. (U, V), (m, n), (O, Q), among others). Specialized local SVMs are introduced to detect the correct class among these two classification hypotheses. The hybrid MLP-SVM recognizer showed improvement, significant, in performance in terms of recognition rate compared with an MLP for a task of character recognition.**

I. Introduction

Handwriting is one of the most natural communication method among people, being particularly able to generate a great amount of data on paper. Often it is necessary to process the data contained in these documents automatically, making it extremely desirable that computers have the ability to 'read' and interpret handwritten documents. The recognition of handwritten characters has been a concern of the scientific community [50]. Applications for a system that makes that sort of recognition are many, for instance, automatic bank checks readers, automatic machines for processing of postal codes, automatic machines to process any type of form filled out manually, and others [31]. The fact that the characters were in handwritten form makes the recognition task complex. This is due to variations of existing styles in handwriting, or the personal style of each writer, since for each writer the form of a character can be done in different ways and even the same writer can represent the character in different ways [31]. Another problem in recognizing handwritten characters is the similarities between different characters, for example, U and V, Q and O, among others. In this article, we demonstrate the advantage of using Support Vector Machines (SVMs) [6] to improve the performance of a ICR system (Intelligent Character Recognition) based on MLP neural networks [53]. In section 2, there is a brief description of character recognition and we point out the main problems of handwriting recognition. In Section 3, we present the most common types of classifiers used for character recognition found in the literature. Section 4 describes the database C-Cube. In Section 5, the feature extraction and some results are briefly introduced. In Section 6, we explain the hybrid architecture MLP-SVM. In section 7 experiments and the results are analyzed. The conclusion is given in the final section.

Washinton Azevedo and Cleber Zanchettin are with the Federal University of Pernambuco, Center of Informatics, 50.732-970, Cidade Universitária, Recife - PE, Brazil (email: {wwas, cz}@cin.ufpe.br).

II. Recognition of characters

Character recognition is something that has been studied extensively by the scientific community since the invention of the computer [32] and consists of the features extracted from a set of characters, separating them into 10 classes, in the case of digits, or 26 classes, in the case of letters of the Western alphabet.

The first commercial systems that emerged were the so-called OCR (Optical Character Recognition). In OCR characters to be recognized are printed on a particular source. The next step was taken toward ICR systems (Intelligent Character Recognition). Unlike the ORBs, the ICR deals with a far more complex problem, since the characters to be recognized are handwritten and not printed.

A. Major problems with the task of Character Recognition

There are some problems that hinder the implementation of character recognition. It can occur, for example, that a scanned image is of low quality due to some inconvenience during the process of scanning the document, thus being necessary to perform a preprocessing to eliminate noise in the image. Another problem that hinders this step is the existence of distorted characters, especially when dealing with handwritten documents, due to the characteristics of the writer's handwriting, which can hinder the recognition of characters by a person.

It may also be highlighted as a possible hindrance the fact that a document may have several different types of characters, for example, uppercase and lowercase letters, numbers, special characters, Greek letters used in mathematical formulas, among others, which implies the development of a more comprehensive and complex character recognition system. Moreover, another difficulty to consider is the similarity between some characters, such as "I" and "J", "Q" and "O ", "U" and "V", among others, that may hinder the classification of recognized characters.

The work [30] describes a set of tests for the evaluation of an extraction technique of features proposed in [27] which uses the database as a set of handwritten letters. In [29] an evaluation of this technique was carried out, but the database was composed only of digits. This technique was developed primarily to treat the problem of recognition of handwritten characters and is based on the projection of the image outline on the sides of a regular polygon built around each character. The feature vector is formed by the perpendicular

978-1-4244-9637-2/11/$26.00 ©2011 IEEE

distances taken from each side of the polygon to the contour of the image. In the evaluation process the proposed approach is compared using two polygons (square and hexagon) and two different amounts of projection lines taken from each side of the polygon, with two versions of coding by bit maps (standard and tuned). The discriminatory power of each case is examined through the use of a MLP neural network (Multi-Layer-Perceptron). The database used in the current experiment was obtained from the digitization of preprinted tender forms (formed only by uppercase letters) duly filled by several people.

In [28], a methodology for recognizing handwritten characters is presented. The proposed methodology is based on a new extraction technique feature based on recursive subdivision of the character image so that the result of sub-images in each iteration has a balanced number (approximately equal) of pixels in the foreground, as far as it is possible. In the experiments two databases of handwritten characters (CEDAR and LIC) and two databases of handwritten digits were used (MNIST and CEDAR). The classification step was performed using Support Vector Machine (SVM) with Radial Basis Function (RBF).

Paper [33] presents a new approach for recognizing cursive characters using multiple feature extraction algorithms and a classifier ensemble. Several feature extraction techniques, using different approaches, are extracted and evaluated. Two techniques, Modified Edge and Multi Zoning Maps, are proposed. Based on the results, a combination of feature sets is proposed in order to achieve high recognition performance. This combination is motivated by the observation that the sets of characteristics are independent and complementary. The ensemble is conducted by combining the outputs generated by the classifier in each set of features separately. The database used for the experiments was the C-Cube, a three-layer MLP network, trained with the Resilient Backpropagation.

In the paper [41], is presented an original hybrid MLP-SVM method for unconstrained handwritten digits recognition. This hybrid architecture is based on the idea that the correct digit class almost systematically belongs to the two maximum MLP outputs and that some pairs of digit classes constitute the majority of MLP substitutions (errors). Specialized local SVMs are introduced to detect the correct class among these two classification hypotheses. The hybrid MLP-SVM recognizer achieves a recognition rate of 98.01%, for real mail zipcode digits recognition task, a performance better than several classifiers reported in recent researches.

The study [13] presents a cursive character recognizer that performs the character classification using SVM and neural gas. The neural gas is used to verify whether lower and upper case version of a certain letter can be joined in a single class or not. Once this is done for every letter, the character recognition is performed by SVMs. SVMs compare notably better, in terms of recognition rates, with popular neural classifiers, such as learning vector quantization and multi-layer-perceptron. SVM recognition rate is among the highest presented in the literature for cursive character recognition.

In the works [30] and [28], an error trend was observed between letters that have similarities (eg (B, D), (H, N) and (O, Q)). In [33] detected a high error rate in characters that have two completely different ways of writing (eg, (a, A) and (f, F) see figure 1). While in [13] the power of network classification SVMs to the task of handwriting recognition it is shown.

Fig. 1. Different shapes for the uppercase version of the letters A and F [33].

Hybrid Intelligent Systems (HIS) has been widely studied in recent years as an alternative to increase efficiency and accuracy. The main motivation for using HIS is that a single technique, because of their limitations and/or disabilities, may not be able, by itself, solve a given problem. For that the combination of several techniques can lead to a more robust and efficient. In this paper we show that by combining simple classifiers (eg MLP-SVM) to the task of character recognition, we can achieve similar or better performance than technical and methods complex of hard trainings presented in literature, in addition to treating the problem of similarity between letters (eg (A, N), (J, S) (M, N) (O, Q) among others) pointed in [30] and [28].

III. TYPES OF CLASSIFIERS

In character recognition systems, the classifier must be chosen appropriately to the type and format of the extracted features [7]. The statistical classifiers and artificial neural networks have been the most widely used classification methods [25] [21].

In statistical classifiers, the features are of the form *n*-tuples or vectors. The purpose of these classifiers is to estimate the probability of belonging to a character analysis of the possible classes. The techniques used for classification can be divided into parametric and nonparametric. Parametric classifiers include *Linear Discriminant Function* (*LDF*) [42] and *Quadratic discriminant Function* (*QDF*) [42], for example. Among the nonparametric classifiers, one can cite the neighbor-next 1-NN, the rule of nearest k-NN neighbors (*K-Nearest Neighbor*) and decision trees. The hidden Markov chains HMM (*Hidden Markov Models*) are statistical classifiers that perform a double stochastic process, being a fundamental stochastic process is not observable (hidden), but it can be observed through another set of stochastic processes that yields the following observations [12]. Applications of HMM initially appeared in speech recognition [4] and were later extended to model characters [23] [17] and words [5] [16].

Artificial neural networks include the multilayer perceptron, networks of radial basis function, probabilistic neural networks and polynomial classifiers [21]. The features that enter the neural classifiers are also the form of *n*-tuples or vectors. Besides the classifier, neural networks can also be used to extract features and to do so the images of its patterns are directly inserted in its entry.

MLPs generally achieve good performance in terms of correct recognition rate in handwritten character classification. Unfortunately, there are limits when using MLPs in classification tasks:

1) First, there is no theoretic relationship between the MLP structure (ex: hidden layers number and neurons number per layer) and the classification task.
2) The second limitation is due to the fact that MLP derives separating hyperplane surfaces, in feature representation space, which are not optimal in terms of the margin (for the margin notion, see [19]) area between the examples of two-different classes. This is because the Backpropagation algorithm, used in the training process, converges as soon as one separating hyperplane solution is reached (local or global minimum), even if this solution is not optimal in terms of the separating margin.

The main difference between statistical classifiers and artificial neural networks is that the network parameters are optimized in a process supervised discriminative learning that seeks to separate the patterns into different classes [34]. When the network structure is properly assigned and the training set has many elements, the artificial neural networks are able to provide a high accuracy in pattern classification of unknown test set, however, training is slow and runs the risk of network loses its ability to spread and become over-specialized [21] [25].

The classifier LVQ (*Learning Vector Quantization*), used in some studies [21] [25] is considered hybrid since it uses the 1-NN rule for classification and adjusts its parameters in the same way that neural networks. A new type of classifier, SVM (*Support Vector Machine*) has emerged in this area and therefore appears in some recent works [36] [35] [51] [52]. SVM is based on statistical learning theory and optimization of quadratic programming, and a binary classifier, so that multiple SVMs may be combined to form a classification system [21] [51] [52]. Systems that employ SVMs have outperformed the results obtained using traditional techniques such as artificial neural networks, however, they require more memory and have higher computational cost. Therefore, hybrid approaches have been employed, using cascading classifiers with simpler classifiers at the beginning and more complex in later levels. Thus, SVM would be responsible for resolving the most difficult cases that are rejected by the simpler classifiers [37].

In recent years, to achieve an optimal recognition rate, much research has resulted in the design of classification systems using different methods of combining multiple classifiers [18] [24] [11] [2]. The idea is to compensate for the weakness of a classifier in a local specific area of the feature space, the robustness of other classifiers as they are properly optimized. The combination method can use local performance estimates [3], Local Learning Algorithms [1], Adaptive Mixtures of Local Experts [8] or to aggregate the decisions obtained from individual classifiers to obtain better final decisions from a standpoint statistics [10]. But the disadvantage of most of these methods is the complexity of optimization for each classifier and the definition of the local area in terms of K-nearest neighbors, which requires storage in system memory of all the training examples. Another feature used to increase the effectiveness of the systems is the use of a modular classifier, which consists of multiple classifiers, each one specializes in a class of problem [38].

The original idea of our method is based on the observation that, when using MLP as a system for recognizing handwritten characters, some pairs of classes constitute the majority of errors (substitutions) made by the MLP (eg ("M", "N") or ("J", "S")) due to the similarity of distinct characters. In order to improve the performance of MLP, our approach is the introduction of Support Vector Machines (SVMs) [22] [6] experts in the local areas around the surface of separation between each pair of characters that make up the majority of the errors of the MLP (confusion).

IV. C-Cube Database

The database C-Cube is a public database available for download on the Cursive Character Challenge website (http://ccc.idiap.ch). The database consists of 57.293 files, including uppercase and lowercase letters, manually extracted from the CEDAR and United States Post Service (USPS) databases. As reported by Camastra [14], this database has three advantages:

1) The database is already divided into training and test sets, so the results of different researchers can be compared rigorously;
2) The database contains not only images but also their feature vectors extracted using the algorithm proposed by Camastra [13];
3) The results obtained using the methods of the state of the art still leave significant space for significant improvement.

The dataset is divided into 38.160 (22.274 lower case and 15.886 upper case) images for training and 19.133 (11.161 lower case and 7.972 upper case) images for test. All image are binary and with variable size. For each image, additional information are provided such as distance between baseline and upper line, distance of the upper extreme from the baseline and distance of the lower extreme from the baseline. The number of samples for each class is variable and was selected according to its frequency in documents extracted from the CEDAR and USPS datasets. Figures 2 and 3 show the distribution of the letters in the lower and upper case versions, respectively. It can be seen that there is a big difference in the number of pattern among the letters.

Thornton [20] observed, through reverse engineering, that the image files (*test.chr* and *training.chr*) do not correspond to the feature vectors (*test.vec* and *training.vec*) available on the C-Cube website. The feature vectors are from a different split of the database (i.e., some images are on the training set in one split and in the test set in the other). For this reason Thornton [20] denominated the dataset of the feature vectors files (*training.vec* and *test.vec*) as *Split A* and the dataset of the image files (*test.chr* and *training.chr*) as *Split B*. It was also proved that the *Split B* consists in a more difficult division of the database, with results around 3 percentile points lower when compared to *Split A*.

In this work only the Split B was used for the experiments because the image files of the Split A are not available.

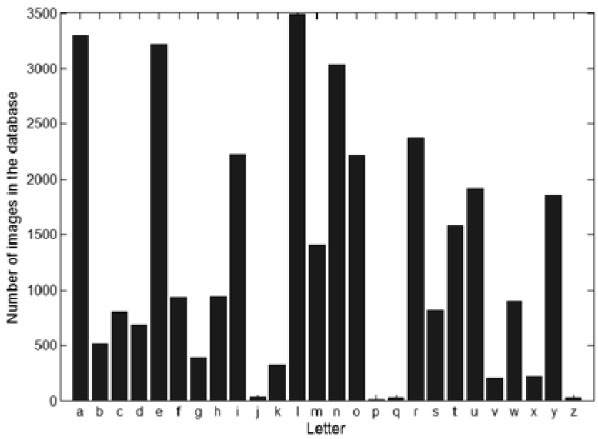

Fig. 2. Lower case letter distribution in the C-Cube Database [33].

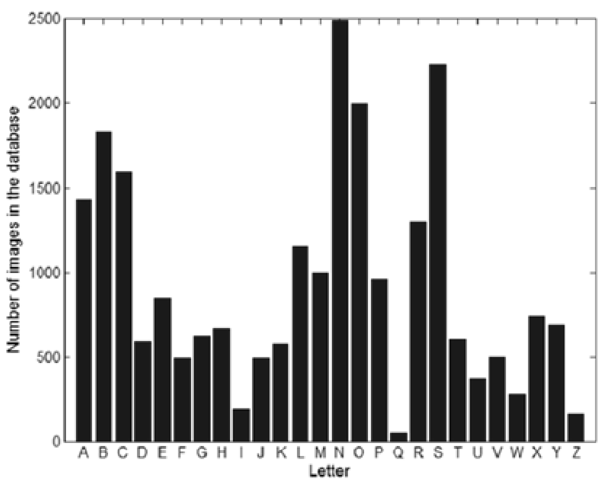

Fig. 3. Upper case letter distribution in the C-Cube Database [33].

V. Feature Extraction

Feature extraction can be defined by the extraction of the most important information to perform the classification for a given task [47]. There are several feature extraction techniques proposed and its choice can be considered the most important factor to achieve high accuracy rates [48].

In [33] two different experiments were made (All the experiments were conducted using a three layers MLP trained using the Resilient Backpropagation [19] algorithm): First splitting upper and lower cases and other experiment with both. For the latter, as some letters present the same shape in both upper and lower case versions, they were joined into a single class. Camastra [13] used a clustering analysis to verify whether the upper and lower case versions of the same letters are similar in shape. The letters (c, x, o, w, y, z, m, k, j, u, n, f, v) presented the highest similarity between the two versions and were joined into a single class.

The classification results for the split and joined cases are shown in Tables I and II, respectively. The results are ordered by the recognition rates. The proposed Modified Edge Maps algorithm presented the overall best result. Most feature sets presented better accuracy for the upper case letters with the exception of the method proposed by Camastra that performed better for lower case letters. This feature set also presented the best accuracy (84.37%) for the lower case letter. It can be seen that the methods based on gradients and the modified edge maps presented the best results. These methods have in common the use of directional information. The Camastra 34D feature set also uses directional features.

VI. An Hybrid Architecture Combining MLP-SVM

One of the problems hindering the task of handwritten character recognition, is the similarity between different characters (for example, S and B, M and H, among others) causing confusion at moment of classifying such patterns. Figure 4 shows some similarities between characters. This observation motivates the search for a suitable method that can detect the correct classification of greater confusion between patterns having a maximum level of confidence in the classification decision. Once the MLP decision is made, the problem is choosing the right class between two ways of classification. This choice results in a binary problem.

One of the most effective methods to solve a binary classification problem, with the utmost confidence in the decision, is the introduction of SVMs. The effectiveness of these methods is due to its ability to construct an optimal separating hyperplane between examples of two different classes. The margin of separation between these examples is maximal.

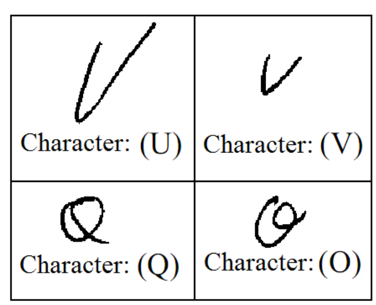

Fig. 4. Similarity between distinct characters.

This combination of methods results in the specialization of SVMs in the local areas around the surface of separation between each pair of characters that constitute the majority of the errors of the MLP (confusion). Thus, SVMs are introduced only for pairs of classes that constitute the greatest confusion of MLP. The MLP-SVM hybrid model is shown in Figure 5.

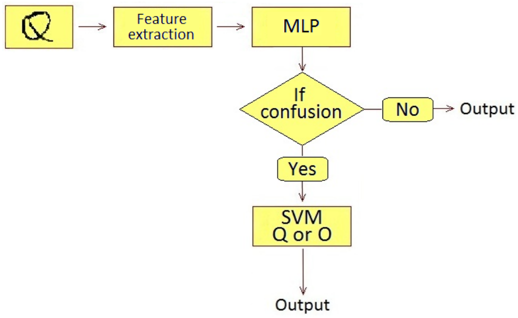

Fig. 5. MLP-SVM Hybrid Model: One pattern is presented to MLP. If the output classified by the MLP network is part of the classes of confusion, this output is presented to the SVM expert to detect the correct class for the classification of these two hypotheses.

TABLE I
RECOGNITION RATE BY FEATURE SET FOR THE UPPER AND LOWER CASE SEPARATED [33]

Method	No. Nodes	Upper Case(%)	Lower Case(%)
Edge	490	**86.52**	81.13
Binary Grad.	490	86.35	79.89
MAT Grad.	300	85.77	79.22
Median Grad.	360	85.10	79.48
Camastra 34D	400	79.63	**84.37**
Zoning	450	84.46	78.07
Structural	320	81.94	77.70
Concavities	530	73.35	81.89
Projections	500	71.73	79.90

TABLE II
RECOGNITION RATE BY FEATURE SET FOR THE JOINT CASE [33]

Method	No. nodes	Recognition Rate(%)
Edge	490	**82.49**
Binary Grad.	490	81.46
MAT Grad.	300	80.83
Median Grad.	360	79.96
Camastra 34D	400	79.97
Zoning	450	78.60
Structural	320	77.07
Concavities	530	74.90
Projections	500	73.85

VII. EXPERIMENTS AND RESULTS

A. *MLP training process*

All experiments were performed using an MLP with three layers (an input layer, a hidden layer and an output layer) and trained with the Resilient Backpropagation algorithm (Rprop) [19]. The algorithm was chosen because Rprop showed faster convergence and better results for this case, when compared to conventional Backpropagation. The values of the weights were adjusted based on the error of the result with the expected result of the network. The error measure used was the SSE (sum of squared errors). Thus it is expected that at each iteration the error decreases. Training is terminated when the error of validation set grows by five iterations in a row or even the thousandth iteration, which is the stopping criterion used.

The architecture of the MLP used for all cases (uppercase, lowercase and uppercase + lowercase) was as follows:

- input layer containing 34 neurons, regarding the number of features extracted from each character using the algorithm proposed by Camastra [13];
- A hidden layer, with the number of nodes based in preliminary tests, choosing the architecture with the best rate of classification. Assuming the following form: 530 nodes for the uppercase, 450 nodes for the lowercase and 400 nodes for the uppercase + lowercase. The number of nodes in the hidden layer for each case was based on preliminary tests to determine the best configuration to solve the problem;
- An output layer containing 26 neurons for uppercase and lowercases ("A" to "Z" and "a" to "z" respectively) and 52 neurons for the uppercase + lowercase ("A" to "Z" + "a" to "z").

Two different experiments were performed: First cases dividing upper and lower case, and another experiment with both cases, the results for the separate and combined cases are presented in Tables III and IV.

TABLE III
RATE OF RECOGNITION FOR THE UPPER AND LOWER CASE SEPARATED

Classifier	Nº of nodes	Case	Recognition rate(%)
MLP	530	Uppercase	86,83
MLP	450	Lowercase	82,12

TABLE IV
RECOGNITION RATE FOR THE JOINT CASE

Classifier	Nº of nodes	Case	Recognition rate(%)
MLP	400	Uppercase + Lowercase	76,74

B. *Analysis of confusion*

With the MLP networks trained for each case (uppercase, lowercase and uppercase + lowercase) their confusion matrices were generated (Figure 6 illustrates the part referring to major confusion occurred in experiments with uppercase letters. Due to the large amount of classes, 26 classes for the case uppercase/lowercase and 52 classes for the uppercase + lowercase, it is impractical to display the complete confusion matrices for all cases) so that we can analyze pairs of characters from which the network obtained the highest number of confusion.

	A	B	C	H	K	L	N	O	R	S	U
A	308	8	4	0	1	3	8	4	5	4	0
C	1	5	370	0	0	9	1	5	1	0	0
D	1	8	2	2	1	1	3	17	0	3	0
E	1	2	9	0	2	0	0	0	9	1	0
G	0	11	5	1	0	1	1	3	1	4	0
H	4	5	0	132	3	0	11	0	0	4	0
J	1	3	0	0	0	0	6	0	0	15	1
L	0	0	4	3	0	249	1	0	4	17	0
M	1	1	0	8	1	0	25	0	2	3	0
N	6	7	1	4	4	1	545	2	3	3	0
R	11	5	5	1	1	1	4	0	277	0	1
S	1	11	4	0	0	2	2	4	2	517	1
V	0	0	0	0	1	1	4	0	0	0	14
W	0	0	1	0	0	0	10	0	0	1	1
X	2	1	0	0	6	2	4	0	7	1	0

Fig. 6. Part of the confusion matrix related to major confusion occurred in experiments with uppercase letters.

C. *SVM training process*

In SVMs different derived from pairs of the classes (e.g. (U, V), (m, n), (N, n) ... etc.) constituting the majority of the confusions of MLP, as shown in figure 6, different kernel functions (linear, polynomial and RBF) were tested and the best performances were obtained by trained SVMs with the RBF kernel function.

D. *Hybrid architecture MLP-SVM*

Based on the confusion matrices analyzed, the hybrid architecture for each case was constructed as follows:

- Uppercase: 1 MLP + 21 SVMs;
- Lowercase: 1 MLP + 32 SVMs;
- Uppercase+ Lowercase: 1 MLP + 44 SVMs.

The choice of pairs of classes was based on the amount of errors taking as minimum 10% the size of the test set.

Table V presents the results obtained by our hybrid MLP-SVM recognizer.

TABLE V
RESULTS OBTAINED BY HYBRID MLP-SVM RECOGNIZER

Case	MLP (%)	SVM+MLP (%)	Improvement (%)
Uppercase	86,83	90,48	3,65
Lowercase	82,12	88,84	6,72
Uppercase + Lowercase	76,74	82,53	5,79

These results show that our hybrid MLP-SVM recognizer improves significantly the performance in terms of recognition rate and error rate compared with one MLP network for one classification task of characters. One can also observe that the results obtained by our method were better than the recognition rates of all the technique of extracting features presented in Tables I and II. It is worth emphasizing the simplicity, the training speed and low computational cost of our method, compared with techniques of extracting features more complex [49] requiring, in most cases, high computational cost and longer training.

The best results obtained in recent years, for C-Cube the database, are displayed in Table VI.

TABLE VI
RECOGNITION RATES FOR THE C-CUBE DATABASE. RBF = RADIAL BASIS NETWORK, HVQ = HIERARCHICAL VECTOR QUANTIZATION, MDF = MODIFIED DIRECTIONAL FEATURES, SVM = SVM WITH RADIAL BASIS KERNEL

Algorithm	#Classes	Recognition Rate(%)
HVQ-32 [39]	52	84.72
HVQ-16 [39]	52	85.58
MDF-RBF [40]	52	80.92
34D-RBF [40]	52	84.27
MDF-SVM [40]	52	83.60
34D-SVM + Neural GAS [13]	52	86.20
34D-MLP [13]	52	71.42
Proposed method	**52**	**82.53**

In [39] The HVQ with temporal pooling algorithm is a partial implementation of the work of [43] on hierarchical temporal memory (HTM). This biologically-inspired model places emphasis on the temporal aspect of pattern recognition, and consequently parses all images as 'movies'. The hierarchy itself is a full 4 level tree of degree 4 that processes a 32 × 32 pixel input character image. During training, each node receives input from the layer below, with leaf nodes receiving a 4 × 4 raw pixel image that is moved one pixel at a time across the node's receptive field, in a process known as sweeping. As the sweep progresses, we count how frequently one pattern follows another. This information is then used to create temporal groups that collect together patterns that have most frequently succeeded another during training. The same process of temporal pooling is repeated at each level up to the root node, where images are classified according to their character values. During recognition, an image is again swept across the leaf node sensors, as each non-root node estimates the membership probability of its input for each of its temporal groups. This information is propagated up to the root, which then outputs the most probable character classification.

Paper [40] the modified direction feature extraction technique combines the use of direction features (DFs) [44] and transition features (TFs) [45] to produce recognition rates that are generally better than either DFs or TFs used individually. MDF extraction proceeds as follows: after initial preprocessing that leaves only the boundary of a character, direction features are used to encode the direction of each line as follows: 2 for vertical, 3 for right diagonal, 4 for horizontal and 5 for left diagonal (see Figure 9). Using this information direction transitions (DT) equal to the corresponding direction feature divided by 10 are extracted for each row (left to right and right to left) and each column (top to bottom and bottom to top). In addition, any contiguous set of equal value direction features is replaced by a single value. Location transitions (LTs) are similarly calculated for each row and each column in both directions, with the relative start positions of each direction feature calculated as a proportion of the total width (in the case of a row) or height (in the case of a column). Given the initial set of LT and DT values corresponding to the actual number of rows and columns in the original character bitmap, the data is then normalised and locally averaged to fit into a space of 5 rows and 5 columns producing a final vector of 120 features [46].

Camastra [13] presented in this work presents a cursive character recognizer, a crucial module in any cursive word recognition system based on a segmentation and recognition approach. The character classification is achieved by using support vector machines (SVMs) and a neural gas. The neural gas is used to verify whether lower and upper case version of a certain letter can be joined in a single class or not. Once this is done for every letter, the character recognition is performed by SVMs. A database of 57 293 characters was used to train and test the cursive character recognizer. SVMs compare notably better, in terms of recognition rates, with popular neural classifiers, such as learning vector quantization and multi-layer-perceptron. SVM recognition rate is among the highest presented in the literature for cursive character recognition.

Observing the results obtained by the models presented in Table VI, we observe that despite the simplicity of our classifier, we achieved better results than MDF-RBF [40] and 34D-MLP [13] in addition to results that can be considered satisfactory in comparison with other models HVQ-32 [39], HVQ-16 [39], 34D-RBF [40], MDF-SVM [40] and 34D-SVM + Neural GAS [13] that are more complex, take longer to be trained and have high computational cost. Thus, we proved that by using a technique of extracting simple features + MLP-SVM, we have as main advantages:

1. Fast training process: Methods of high complexity that present a slow convergence, the training process can stop in regions of local minimum where the gradients are null;
2. Low computational cost: Complex methods, in turn, present calculations of high complexity, which may cause loss of system efficiency;
3. Similar or better results to models of high complexity;

VIII. Conclusions

In this paper, a method for increasing the rates of recognition of handwritten characters by combining MLP + SVM is proposed. The experiments demonstrated that the combination of MLPs networks with SVMs experts pairs of classes that constitute the greatest confusion of MLP, have improved performance in terms of recognition rate. The results showed a significant improvement from 3.65% to 5.79% in recognition rate for all cases tested (uppercase, lowercase and uppercase + lowercase).

As a proposal for future research, we intend to use different feature extraction techniques combined with classifier ensemble in our hybrid MLP-SVM, and motivated to achieve better recognition rates for the recognizer.

Another direction we would like to pursue is the use of clustering analysis [13] to verify whether the upper and lower case versions of the same letters are similar in shape. With this, we could reduce the number of confusions in the MLP and hence achieving better recognition rates.

Acknowledgment

This work was supported by FACEPE (Brazilian Research Agency).

References

[1] L. Bottou and V. Vapnik. Local learning algorithms. Neural Computation, 4(6):888–900, 1992.
[2] Rahman AFR, Fairhurst MC (2000) Multiple expert classification: a new methodology for parallel decision fusion. International Journal on Document Analysis and Recognition 3:40–55.
[3] K.Woods,W. P. K. Jr, and K. Bowyer. Combination of multiple classifiers using local accuracy estimates. IEEE Trans. on PAMI, 19(4):405–410, 1997.
[4] Fechine, J.; Madeiro, F.; Neto, B. G. A. Reconhecimento automático da identidade vocal baseado em quantização vetorial e HMM (In Portuguese). 2da. Conferencia Iberoamericana en Sistemas, Cibernética e informática - CISCI, 2003.
[5] Yacoubi, E. et al. An HMM-based approach for off-line unconstrained handwritten word modelling and recognition. IEE Transaction on Pattern Analysis and Machine Intelligence, v. 21, n. 8, p. 752-760, 1999.
[6] V. N. Vapnik. An overview of statistical learning theory. IEEE Trans. on Neural Networks, 10(5):988–999, 1999.
[7] Trier, O.D.; Jains, A. K.; Taxt, T. Feature extraction methods for character recognition - a survey. Pattern Recognition, v. 29, n. 4, p. 641-662, 1996.
[8] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. Neural Computation, 3(1):79–87, 1991.
[9] B. E. Boser, I.M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In Proc. Fifth Annual Workshop on Comput. Learn. Theory, Pittsburgh, USA, July 1992.
[10] Huang YS, Suen CY (1995) A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(1):90–94.
[11] J. Kittler. Combining classifiers: A theoretical framework. Pattern Analysis and Applic., 1(1):18–27, 1998.
[12] Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, v. 77, n. 2, p. 257-289, 1989.
[13] Francesco Camastra, "A svm-based cursive character recognizer,"Pattern Recognition, vol. 40, no. 12, pp. 3721–3727, 2007.
[14] Francesco Camastra, Marco Spinetti, and Alessandro Vinciarelli, "Offline cursive character challenge: a new benchmark for machine learning and pattern recognition algorithms.," Proceedings of the 18th International Conference on Pattern Recognition, pp. 913–916, 2006.
[15] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. Disponível em: http://www.csie.ntu.edu.tw/~cjlin/libsvm/. Acessado em: 09/2003, 2004.
[16] Freitas, C.; Bortolozzi, F.; Sabourin, R. Handwritten isolated word recognition: An approach based on mutual information for feature set validation. Proceedings 6th Internacional Conference on Document Analisys and Recognition, p. 665-669, 2001.
[17] Cai, J.; Liu, Z.-Q. Integration of structural and statistical information of unconstrained handwritten numeral recognition. IEE Transactions

on Pattern Analysis and Machine Intelligence, v. 21, n. 3, p. 263-270, 1999.

[18] Nadal C, Legault R, Suen CY (1990) Complementary Algorithms for the Recognition of Totally Unconstrained Handwritten Numerals. In: Proceedings of the tenth International Conference on Pattern Recognition 1990. Atlantic City, NJ, USA, pp 443–449.

[19] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," Proceedings of the IEEE International Conference on Neural Networks, pp. 586–591, 1993.

[20] John Thornton, Michael Blumenstein, Vu Nguyen, and Trevor Hine, "Offline cursive character recognition: A state-of-the-art comparison,"14th Conference of the International Graphonomics Society, 2009.

[21] Liu, C.-L. et al. Handwritten digit recognition: Benchmarking of state-of-the-art techniques. Pattern Recognition, v. 36, p. 2271-2285, 2003.

[22] V. N. Vapnik. Statistical Learning Theory. John Wiley and Sons, New York, USA, 1998.

[23] Brito, A. S. et al. Foreground and background information in an HMM-based method for recognition of isolated characters and numeral strings. Internacional Workshop on Frontiers in Handwriting Recognition, p. 371-376, 2004.

[24] Ho TK, Hull JJ, Srihari SN (1994) Decision Combination in Multiple Classifier Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(1):66–75.

[25] Liu, C.-L; Sako, H.; Fujisawa, H. Performance evaluation of pattern classifiers for handwritten character recognitio. Ineternacional Journal on Document Analysis and Recognition - IJDAR, v. 4, p. 191-204, 2002.

[26] A. J. Smola and B. Schölkopf. Learning with Kernels. TheMIT Press, Cambridge,MA, 2002.

[27] Rodrigues, R. J.; Kupac, G. V.; Thomé, A. C. G.; "Character Feature Extraction Using Polygonal Projection Sweep (Contour Detection)", IWANN2001, Granade - Spain, June 13 a 15, LINCS, 2001, pp 687-695, 2001.

[28] G. Vamvakas; B. Gatos; S. J. Perantonis; "Handwritten character recognition through two-stage foreground sub-sampling"; Pattern Recognition 43 (2010) 2807–2816.

[29] Rodrigues, R. J.; Silva, Eugênio; Thomé, A. C. G.; "Feature Extraction Using Contour Projection"; 5th World Multiconference on Systemics, Cybernetics and Informatics SCI2001 – Orlando – USA, July 22 – 25.

[30] Rodrigues, R. J.; Silva, Eugênio; Thomé, A. C. G. "Extração de Características para o Reconhecimento de Letras Manuscritas" (In Portuguese), 2002.

[31] Aires, S.B.K., Reconhecimento de caracteres manuscritos baseados em regiões perceptivas (In Portuguese). Dissertação de Mestrado. Pontifícia Universidade Católica do Paraná. Curitiba. 2005.

[32] Veloso, L. R., Reconhecimento de Caracteres Numéricos Manuscritos (In Portuguese). Dissertação de Mestrado, Universidade Federal da Paraíba, Campina Grande, 1998.

[33] Rafael M. O. Cruz, George D. C. Cavalcanti and Tsang Ing Ren. An Ensemble Classifier For Offline Cursive Character Recognition Using Multiple Feature Extraction Techniques. IEEE International Joint Conference on Neural Networks (IJCNN), Barcelona, p. 744-751, 2010.

[34] Haykin, S. Neural Networks - A Comprehensive Foundation. [S.I.]: Prentice Hall, 1996.

[35] Oliveira, L. S.; Sabourin, R. Support vector machines for handwritten numerical string recognition. 9th International Workshop on Frontiers in Handwiting Recognition (IWFHR-9), p. 39-44, 2004.

[36] Teow, L.-N.; Loe, K.-F. Robust vision-based features and classification schemes for off-line handwritten digit recognition. Pattern Recognition, v. 35, p. 2355-2364, 2002.

[37] Vuurpijl, L.; Schomaker, L.; Van, M. Architectures for detecting and solving conflits: Two-stage classification and support vector classifiers. International Journal on Document Analisys and Recognition, p. 213-223, 2003.

[38] Oh, I. -S.; Suen, C. Y. A class-modular feedforward neural network for handwriting recognition. Pattern Recognition, V. 35, p. 229-244, 2002.

[39] John Thornton, Jolon Faichney, Michael Blumenstein, and Trevor Hine, "Character recognition using hierarchical vector quantization and temporal pooling," Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence, pp. 562–572, 2008.

[40] John Thornton, Michael Blumenstein, Vu Nguyen, and Trevor Hine, "Offline cursive character recognition: A state-of-the-art comparison," 14th Conference of the International Graphonomics Society, 2009.

[41] Bellili A., Gilloux M., and Gallinari P., "An Hybrid MLP-SVM Handwritten Digit Recognizer," icdar, pp.0028, Sixth International Conference on Document Analysis and Recognition (ICDAR'01), 2001.

[42] Duda, Hart, and Stork, "Pattern Classification" Chapter 5, Wiley, 2000.

[43] George, D., Hawkins, J. (2005). A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. Proc. Int. Joint Conf. on Neural Networks - IJCNN-05, Montreal, Canada, pp. 1812-1818.

[44] Blumenstein, M., Verma, B. & Basli, H. (2003). A novel feature extraction technique for the recognition of segmented handwritten characters. Proc. Int. Conf. On Document Analysis and Recognition – ICDAR-03, pp. 137-141.

[45] Gader, P. D., Mohamed, M. & Chiang, J. H. (1997). Handwritten word recognition with character and inter-character neural networks, IEEE Trans. Syst. Man Cybernet.—part B: Cybernetics 27, 158–164.

[46] Blumenstein, M., Liu, X. Y. & Verma, B. (2007). An investigation of the modified direction feature vector for cursive character recognition. Pattern Recognition 40(2), 376-388.

[47] P.A. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, Londres, 1982.

[48] Oivind Due Trier, Anil K Jain, and Torfinn Taxt, "Feature extraction methods for character recognition: A survey," Pattern Recognition, vol. 29, no. 4, pp. 641–662, 1995.

[49] Oivind Due Trier, Anil K Jain, and Torfinn Taxt, "Feature extraction methods for character recognition: A survey," Pattern Recognition, vol. 29, no. 4, pp. 641–662, 1995.

[50] Plamondon R., Srihari S. N., "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.22, p. p. 63-84, 2000.

[51] Kecman V., "Learning and Soft Computing, Support Vector machines, Neural Networks and Fuzzy Logic Models", The MIT Press, Cambridge, MA, 2001.

[52] Wang L.P. (Ed.), "Support Vector Machines: Theory and Application", Springer, Berlin, 2005.

[53] Rumelhard, D.E., Weigend, S.A.; "Predicting the Future: A Connectionist Approach", International Journal of Neural Systems, pp. 193-209, 1990.