Published at: Lecture Notes in Engineering and Computer Science -Proceedings of the 2010 Word Congress on Engineering and Computer Science (WCECS 2010) – International Conference on Soft Computing and Applications (ICSCA 10), San Francisco, USA. Vol. 1, pp. 525 - 528. Eds. Ao SI, Douglas C, Grundfest WS, Burgstone J. International Association of Engineers. ISBN: 978-988-17012-0-6. ISNN: 2078-0958 (print) ISNN: 2078-0966 (online).

Unconstrained Handwritten Word Recognition Using a Combination of Neural Networks

Rodolfo Luna-Pérez, Pilar Gómez-Gil

Abstract- **Automatic offline recognition of text handwritten by different writers is still an open problem, due to several challenges including strong variability in writing styles, noise embedded in the environment, segmentation issues and others. In order to avoid errors during character segmentations, systems based on recognition of whole words have been developed lately. In this paper we present a novel method for classification of isolated handwritten words based on three components: a self organizing map (SOM) for non-supervised classification of segments of a word, a function measuring probabilities of each segment belonging to a specific cluster and a simple recurrent network (SRN) for temporal classification of a sequence of feature vectors obtained from segments forming the world. The experiments showed that the combination of these three components significantly improved the classification of words obtained from the benchmark IAM when compared with a multi-layer perceptron (MLP) and a plain combination of SOM and MLP. The proposed classifier obtained a mean word accuracy of 78.2% over a test set, compared to 66.2% obtained by a SOM combined to a MLP and to 32.1% obtained by MLP.**

Index Terms— **Offline word handwritten recognition, temporal classification, Simple recurrent network, Self organizing maps.**

I. Introduction

Recognition of manuscript texts refers to the activity of transforming a set of marks to symbols [1]. This task may be executed either online or offline, being the first easier to execute than the second, due to temporal information captured by the system as handwritten takes place online [2]. However, offline recognition is very important nowadays, because there is a huge amount of non-digital documents that are required to be interpreted in digital form. For this reason, research related to offline handwritten recognition is looking for the most effective ways to obtain meaningful text from document images. These studies include several strategies to recognize text, based on recognition by character, by word and even by line. Word recognition consists on finding the word that is most compatible to a specific image, from a previously defined lexical set [3]. According to Namane et. al [4] word recognition techniques may be classified as analytical or global. Analytical techniques consist on dividing an image in segments that may be characters or pseudo-characters, that is, parts of characters with no meaning for humans. Then each segment is identified in some way and using context information and a dictionary, a word is assigned to the image. Global techniques handle an image of a word as a whole entity, with no segmentation involved. In this case, recognition is carried out using characteristics obtained from the complete image. Analytical techniques have the advantage that errors generated by character segmentation are avoided, but in the other hand, in most cases they require to define a specific model for each word involved in the lexical context. For example, the most popular of these analytical techniques use hidden Markov models (HMM). Besides, these models have the disadvantage that they assume that the probability of each observation depends only on the current state, with no use of contextual information [5]. It is well known that human beings use contextual information as an important aid in the reading tasks, therefore, models able to represent context could produce better results.

In the last years, interesting results have been obtained on handwritten line and word recognition with systems based on recurrent neural networks (RNN). Due to feedback connections included at RNN neurons, they are able to memorize temporal information embedded in the input training data [6], allowing contextual information to be involved in a recognition system. Other very powerful neural networks are the Self-Organizing Maps (SOM), which are able to generate groups with no supervision. Once trained, the output layer of a SOM forms a two-dimensional map, where each node contains a prototype of a cluster. This map is topologically related, that is, each neuron represents a cluster similar to clusters represented by its neighbors. This allows to known not only what is the best cluster a sample could belong to, but also what other clusters contain similar patterns. Due to its self-organizing abilities and topological relations, SOM has been used to build a vast number of recognizers where a-prior exact classification is not available or desired, as in the case of off-line handwritten recognition [7]. In this work we present a new analytical model for off-line classification of handwritten words, which takes advantage of the topological information represented in a SOM trained with segments of words, and using a metric to represent the probability of a segment to belong to a specific cluster defined by the SOM and to its neighbors. This

Manuscript received July 14, 2010. R. Luna thanks the National Council of Science and Technology (CONACYT) for the financial support received during this work by the scholarship # 27156. R. Luna-Pérez and P. Gómez-Gil are with the National Institute of Astrophysics, Optics and Electronics, Department of Computational Science Tonantzintla Puebla, México. e-mail:
pgomez, rodolfo.luna}@inaoep.mx, pgomez@acm.org .

Published at: Lecture Notes in Engineering and Computer Science -Proceedings of the 2010 Word Congress on Engineering and Computer Science (WCECS 2010) – International Conference on Soft Computing and Applications (ICSCA 10), San Francisco, USA. Vol. 1, pp. 525 - 528. Eds. Ao SI, Douglas C, Grundfest WS, Burgstone J. International Association of Engineers. ISBN: 978-988-17012-0-6. ISNN: 2078-0958 (print) ISNN: 2078-0966 (online).

information is fed to a simple recurrent network (SRN), which memorizes the temporal relationships among all segments and their respective measures of similarity, and assigns the input image to the best matching word. This system is writer-independent and works with a pre-defined vocabulary.

The paper is organized as follows: Section II presents the main components of the proposed classifier; section III described the training scheme used for this system; section IV presents the results from experiments developed to test the model; section V summarizes some conclusions and ongoing work.

II. The Proposed Classifier

Similar to other handwritten classifiers, this one contains four main components: pre-processing of input images, segmentation, generation of feature vectors and word recognition (See Fig. 1). It must be pointed out that segmentation is used to obtain portions of the image that will feed the clustering, even though these segments may not correspond to a character. Next, each component is detailed.

1. Preprocessing includes noise elimination, binarization and slant correction; Fig. 2 shows an example of an image before and after preprocessing it.

2. After preprocessing, the resulting image is segmented to portions of variable length that we called "pseudo-characters." For the results presented here, the cutting position of each segment is decided manually. This process may be executed automatically based on column histograms, because the correspondence of each segment with a real manuscript character is not of relevance for this process. Each image of a segment is normalized to the same size, and represented by a binary one-dimensional vector storing the segment by rows. The number of segments in each word is variable.

3. The generation of feature vectors is based on a SOM and a probabilistic metric. During the training phase of the classifier, a SOM of 20x20 is trained to cluster the training segments in an unsupervised way. During the classification phase, each segment *t* is fed to SOM, getting a winning neuron c_t , which represents the cluster where that segment is best suited, plus the *k-1* neurons, each identified as m_{ti} $i=2..k$, with highest activations in the map. Let $m_{ti}= c_t$. In this way the SOM gives information of the *k+1* most representative clusters for that segment. Next, a measure of the probability of each neuron to represent the segment is calculated. This measure is proportional to the Euclidian distance in the map of each neuron to the winning neuron, as defined by the equation:

$$p_t(\mathbf{m_{ti}}) = \frac{\exp(-\|\mathbf{m_{ti}}-\mathbf{c_t}\|)}{\sum_{\mathrm{i=1}}^{\mathrm{k}}\exp(-\|\mathbf{m_{ti}}-\mathbf{c_t}\|)} \qquad (1)$$

where:

$\mathbf{c_t}$ is a 2D vector defined by the coordinates of the winning neuron at SOM map obtained when segment t is applied,

$\mathbf{m_{ti}}$ is the 2D vector defined by the coordinates of the *i*-neuron with highest activation at the SOM, *i= 1..k*, obtained when segment *t* is applied.

Notice that for each segment t:

a) $\sum_{i=1}^{k} p_t(\mathbf{m_i}) = 1$ (2)

b) if $\|\mathbf{m_i}-\mathbf{c}\| = \|\mathbf{m_j}-\mathbf{c}\|$ then $p(\mathbf{m_i}) = p(\mathbf{m_j})$ (3)

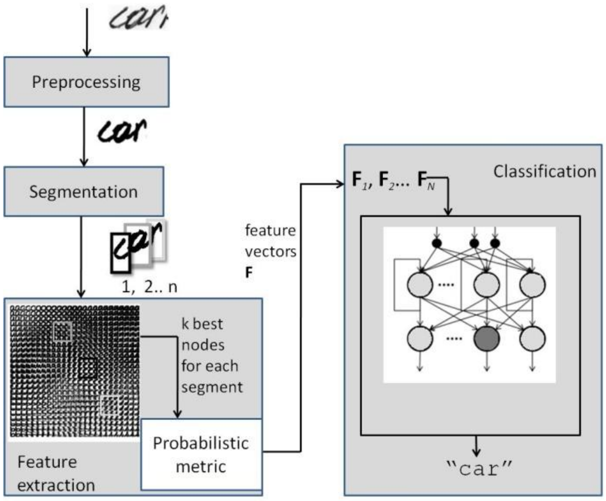


Fig. 1. The proposed method for classification of a word

In summary, feature vector Ft for each *t*- segment is defined as:

$$\boldsymbol{F_t} = (m_{t11}, m_{t12}, p(\boldsymbol{m_{t1}}),\; m_{t21}, m_{t22}, p(\boldsymbol{m_{t2}}), \dots m_{tk1}, m_{tk2}, p(\boldsymbol{m_{tk}})) \qquad (4)$$

Notice that $\mathbf{F_t}$ is *3k* –dimensional.

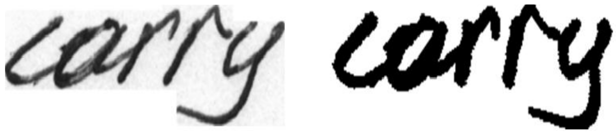
4. Word recognition is carried out using a Simple Recurrent Network (SRN), as defined by Elman at [8]. The network is fed with segments one at the time, up to the last segment of the input word is introduced. When the last segment of a word is introduced, the output neuron in the SRN with the highest output value represents the assigned class to such word. Outputs of the network before that last segment is introduced are not considered for classification.

The SRN network used here consists of 3 layers: an input layer with *3k* neurons, a hidden layer with recurrent connections and an output layer with as many nodes as the number of words to be recognized. The number of nodes in the hidden layer is chosen by experimentation. The output nodes use the activation function *Softmax*, defined as:

$$soft\max(x) = \frac{\exp(x)}{\sum_{i=1}^{n}\exp(i)} \qquad (5)$$

Where *x* is the output of involved output neuron and *n* is the total number of neurons at output layer. Notice that *Softmax* allows the sum of all output nodes to be 1, which allows interpreting the output of each neuron as the probability that the sequence input so far to the network could belong to the word represented by this neuron.

Published at: Lecture Notes in Engineering and Computer Science -Proceedings of the 2010 Word Congress on Engineering and Computer Science (WCECS 2010) – International Conference on Soft Computing and Applications (ICSCA 10), San Francisco, USA. Vol. 1, pp. 525 - 528. Eds. Ao SI, Douglas C, Grundfest WS, Burgstone J. International Association of Engineers. ISBN: 978-988-17012-0-6. ISNN: 2078-0958 (print) ISNN: 2078-0966 (online).



(a) (b)

Fig. 2. (a) Original image, (b) image after pre-processing

III. Training

Two networks are trained: SOM and SRN. SOM is trained in an unsupervised way in order to cluster the segments generated using words in the training set. From this process prototypes of segments or "pseudo-characters" are generated. The original algorithm proposed by Kohonen [9] was used to train SOM, using the Neural Net Library V 5.02 in Matlab V7.4.

SRN is trained to receive each segment $\mathbf{F_t}$ (equation 4) of each word *w* and to output a probability that such segment is the last segment at word *w*. SRN contains as many output nodes as possible words (classes) are in the lexical represented in the training set. The desired output of each output node *i* when segment $\mathbf{F_t}$ is input, is calculated as:

$$d_i = \begin{cases} x/n & if\ i = corresponding\ word\ of\ \boldsymbol{F_t} \\ 0 & otherwise \end{cases} \qquad (6)$$

where:
x is the position of segment $\mathbf{F_t}$ in the word,
n is the number of segments at the word.

The algorithm back propagation through time is used to train the network. SRN is trained with the prototypes of each neuron at SOM, which represent the clusters. In order to avoid that the network learns the whole training set as a unique sequence, weights are adjusted after presenting to the network a word.

IV. Experiments and Results

The proposed method was tested using a lexical with 10 words taken from database IAM [10]. Training set contained twenty images of each word, and testing set contained five images of each word. The lexical is shown at table I. It is important to pointed out, that, for the results presented here, the database was manually segmented. This is because we decided to isolate the noise produced by automatic segmentation in order to test the performance of the proposed method by itself. The proposed method was compared with other two neural classifiers, therefore three cases were analyzed:

1) *A feed-forward (FF) network.* The input to this network is made with all segments of a word, without any kind of processing using all bits in the image. Images were normalized to a fixed size, requiring then 1,200 input nodes; the number of output nodes is 10 (one for each class).

2) *FF-SOM network.* Here, a feed-forward network is also used, but its inputs are made of all vectors $\mathbf{F_t}$ (equation 4) obtained when applying step 3, described at section II, to all segments obtained from the input image. This feature extractor uses a SOM network, and a parameter $k = 5$. All vectors are given at once as input, which requires fixing a maximum number of segments by word to be handled, that for these experiments was three. Therefore, the number of input nodes for this case is 45 (3 segments*5 neighbors*3 values for each segment, as described by equation 4). As in case one, the number of output nodes is 10.

3) *SRN-SOM.* These networks correspond to the proposed method. As in case 2, inputs are obtained as described at section II. However, given the fact that a recurrent network is able to memorize, in this case each segment of the work is presented at once to the network, which makes the number of inputs to be reduced to 15 (5 neighbor values* 3 segments).

Twenty experiments were executed for each network case, using 150 network different configurations for each experiment. The difference in each configuration is the number of hidden nodes in the FF networks. Performance was measured in two ways: As the percentage of error of classification, defined as:

$$Error = \frac{\text{number of words incorrectly classified}}{\text{total number of words}} * 100 \qquad (7)$$

and using the word accuracy metric defined by Graves et al [3], defined as:

$$WA = 100 * (1 - \frac{\text{insertions} + \text{substitutions} + \text{eliminations}}{\text{set size}} \qquad (8)$$

Where insertions, substitutions and eliminations refers to the number of such changes that are required to make in each character of the word selected by the classifier, to become the right class. For example, if the classifier assigned the class "as", and the right class is "at", there is one substitution involved.

Tables II and III show the performance obtained by each case, using the metrics classification error and word accuracy respectively. Notice that, according to both metrics, the SRN-SOM method obtained the best results.

Table I. Lexical used for the experiments

a	and	are	as	at	be	but	bye	can	for

Table II. Classification error obtained by the three cases

Case	Error in training set	Error in testing set
1) FF	44.00% ±28.00%	71.00% ±15.00%
2) SOM-FF	8.00% ±2.00%	37.00% ±5.00%
3) SOM-SRN	**5.75%±1.34%**	**24.30%±5.12%**

Table III. Word accuracy obtained by the three cases

Case	Word Accuracy using training set	Word Accuracy using testing set
FF	53.03 ±27.42	33.12 ±17.25

Published at: Lecture Notes in Engineering and Computer Science -Proceedings of the 2010 Word Congress on Engineering and Computer Science (WCECS 2010) – International Conference on Soft Computing and Applications (ICSCA 10), San Francisco, USA. Vol. 1, pp. 525 - 528. Eds. Ao SI, Douglas C, Grundfest WS, Burgstone J. International Association of Engineers. ISBN: 978-988-17012-0-6. ISNN: 2078-0958 (print) ISNN: 2078-0966 (online).

SOM-FF	93.03 ±1.5	66.21 ±5.85
SOM-SRN	**95.42±1.21**	**78.25±3.25**

V. Conclusions

In this paper a new classifier of handwritten, isolated, writer-independent words is presented. A word is recognized avoiding the challenge of character segmentation by just segmenting it in small portions that may or may not correspond to a real character. Such segments are grouped in an unsupervised way and such information is used to generate features. The proposed classifier is based on the use of three main components: a feature extractor based on non-supervised clustering using a self organized map, a measure of the probability that a segment of a word belongs to the *k* most probable clusters and a simple recurrent network able to classify sequences of features representing the words, made of information given by the two previous components. The method showed to overcome two other neural classifiers when tested over a set of 10 words taken from the IAM benchmark database, that were written by different people and showing very different styles.

Currently, we are working in analyzing the use of other more powerful recurrent neural networks and other learning algorithms, in order to improve the classification performance. Also we are analyzing the performance of this classifier when fully automatic segmentation is carried out.

References

[1] R. Plamodon and S. Srihari. “On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 2000, pp. 63-84.

[2] P. Gómez-Gil, G. De los Santos-Torres, J. Navarrete-García and J.M. Ramírez-Cortés, “The Role of Neural Networks in the interpretation of Antique Handwritten Documents,” in *Hybrid Intelligent Systems. Analysis and Design Series: Studies at Fuzziness and Soft Computing,* O. Castillo, P. Melin, W. Kacprzyk, Ed., vol. 208. Berlin: Springer. 2007, pp. 269-281.

[3] A. Vinciarelli. “A survey on off-line Cursive Word Recognition”. *Pattern Recognition, The journal off pattern recognition society*, vol. 35 (7), 2002, pp. 1433 - 1446.

[4] A. Namane, A. Guessoum and P. Meyrueis. “New Holistic Handwritten Word Recognition and Its Application to French Legal Amount”. *Sprinnger-Verlag Berlin Heidelberg* , 2005, pp. 654-663.

[5] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke and J. Schmidhuber. “A Novel Connectionist System for Unconstrained Handwriting Recognition”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 31, 2009, pp. 855-868.

[6] S. Haykin. “Neural Networks: A Comprehensive Foundation”, Upper Saddle River, NJ; Prentince Hall, 1999.

[7] E. Cuevas-Farfán and P. Gómez-Gil. “PRISCUS: Reconocedor Óptico de caracteres manuscritos y antiguos” (“PRISCUS: Optical recognizer for antique handwritten characters”) *Proceedings of the 9º Research Workshop in the National Institute of Astrophysics, Optics and Electronics.* Tonantzintla, Puebla. Nov 2008 pp. 147-150

[8] J. Elman. “Finding Structure in Time”. *Congnitive Science , Vol. 14*, 1990 ,pp. 179-211.

[9] T. Kohonen. “Self-Organizing Maps”, *Springer Series in Information Science,* 3rd edition, 2001.

[10] U. V. Marti and H. Bunke, “The IAM-database: an English sentence database for offline handwritten recognition,” Int. Journal on Document Analysis and Recognition, vol. 5, pp. 39-46, 2002.