Writer verification based on simple graphemes and extreme learning machine approaches

A Vásquez¹, M Mora¹, V Aubin², E Salazar¹, R Barrientos³, R Hernandez¹, and K Vilches¹

¹ Laboratorio de Investigaciones Tecnológicas en Reconocimiento de Patrones, Universidad Católica del Maule, Talca, Chile
² Departamento de Ingeniería e Investigación Tecnológica, Universidad Nacional de la Matanza, Buenos Aires, Argentina

E-mail: marcomoraco@gmail.com

Abstract. Traditional literature presents complex biometric sources, descriptors, and classifiers to solve the writer’s verification problem. The simple graphemes have been studied recently considering classifiers such as multilayer perceptron, support vector machine and convolutional neural network, which allow a high level of performance but with high computational cost in the training. In this paper, we propose the use of extreme learning neural networks to verify the writer identity based on simple graphemes with the aim of achieve a better descriptor performance in a less training time. The proposal allows verify peoples identity through the analysis of handwritten text in order to fakes detect, authorship identification, fakes, threats and thefts in documents. The experimental results show that this type of classifiers achieve a rate of success greater to the 95% for all five characters in the problem addressed, but with significantly less training times than traditionally used techniques.

1. Introduction

There are different biometric features that allow the people recognition, among them is find the writing. The rhythm of writing is unrepeatable and unique, incorporating in the text particular graphic characteristics that allow the author identification. The people recognition through the analysis of handwritten texts is very used in different tasks such as: authorship identification, fakes detect, frauds and theft, in documents of types different such as wills, letters, checks, etc.

The writer’s verification problem through handwritten traces has been approached traditionally analyzing complex sources such as text pages [1], words [2] and signatures [3]. The structural complexity of the sources used also implies a great complexity of the descriptors developed. In addition of the above, have been used classifiers such as the multilayer perceptron (MLP), support vector machine (SVM) and convolutional neural networks (CNN), all them report high levels of verification performance but with a high computational cost in training.

With the objective of reducing the global complexity of the writer’s verification problem, recently has been proposed to extract the biometric information of the handwritten text from very simple individual graphemes such as “C”, “∩”, “∪”, “S” and “∪” [4, 5]. In these works are used classifiers of the SVM type, known by their high levels of adjustment in classification and regression problems, but with high computational cost in the training algorithm and the estimation of the model’s hyperparameters.
The extreme learning machines (ELM) [6] have begun to be used massively in complex pattern recognition problems due to the low computational cost of their training algorithm. In this paper is proposed the use of ELM neural networks to address the writer verification problem by analysis of simple graphemes. The experiments are carried out with single layer ELM networks, and with a multilayer model (ML-ELM). The results show high levels of verification performance, but achieved with the reduced times that are characteristic of ELM training.

2. Simple graphemes for writer’s verification

The simple graphemes database used in this work corresponds to the proposed in [4, 5]. This database contains 50 writers, 5 types of simple graphemes (“C”, “∩”, “∼”, “S” and “∪”), 50 samples by grapheme, reaching a total of 12500 images. The proposal of the article is interesting because the characters mentioned are much simpler than the signatures, which is the writing stroke most studied in the literature. It allows the use of classifiers to fakes detect, authorship identification, fakes, threats and thefts in documents.

Such as shown the Figure 1(a), the grapheme image contains many white pixels (pixels of the background) which contain no information. For obtain an image that considers only the pixels of the grapheme, is constructed a rectified image that consists of a “stretched” version of the grapheme. The rectification procedure is made up of a sequence of simple operations of image processing that are graphically represented in Figure 1(b). The sequence of operations are explained in detail in [5], but can summarized as follows:

- Convert the color image of the grapheme to a grayscale image using the V channel of the HSV model.
- Binarize the grayscale image of the V channel through the well-known algorithm of Otsu.
- Obtain the morphological skeleton of the binary image of the H channel (white line in Figure 1(b)).
- Obtain the lines perpendicular to the morphological skeleton (black lines in Figure 1(b)).
- Build an image with the pixels of the grayscale image that are found over the perpendicular lines.

![Figure 1](image-url)

**Figure 1.** Simple graphemes rectification (a) graphemes database, (b) graphemes rectification process, (c) rectified image of the grapheme.
The image resulting of the rectification process is shown in Figure 1(c). It is important point out that this rectified image, to the be in grayscale and not include the pixels of the background, drastically reduces the dimension of the color image of the original grapheme. To represent the rectified image of the grapheme in a vector format necessary for the neural classifiers, is made use of the well-known texture descriptor local binary pattern (LBP) [7] such as is propose in [5].

3. Neural networks for writer verify through simple graphemes

3.1. Extreme learning machine

Let \( \{ \{x_i, t_i\} : x_i = (x_{i1}, x_{i2}, \ldots, x_{im})^T \in \mathbb{R}^n, t_i = (t_{i1}, t_{i2}, \ldots, t_{im})^T \in \mathbb{R}^m \} \) with \( i = 1, \ldots, N \) be a set of \( N \) training samples for an ELM neural network. The characteristics learning for the input data space of a single hidden layer feedforward neural network with activation function \( g : \mathbb{R}^n \rightarrow \mathbb{R}^m \) and \( L \) hidden neurons was introduced in [6] through the Equation (1).

\[
\sum_{i=1}^{L} \beta_i g(w_i \cdot x_j + b_i) = t_j, \quad j = 1, \ldots, N, \tag{1}
\]

where \( w_i = (w_{i1}, w_{i2}, \ldots, w_{im})^T \) and \( b_i \) are weights and biases of the hidden layer randomly generated, \( \beta_i = (\beta_{i1}, \beta_{i2}, \ldots, \beta_{im})^T \) are output weights and \( w_i \cdot x_j \) is the usual inner product of the vectors \( w_i \) and \( x_j \) [6]. The Equation (1) can be written in matrix form \( H\beta = T \) with \( H \), \( \beta \) and \( T \) defined in [6] through the Equation (2).

\[
H = \begin{bmatrix}
g(w_1 \cdot x_1 + b_1) & \cdots & g(w_L \cdot x_1 + b_L) \\
\vdots & \ddots & \vdots \\
g(w_1 \cdot x_N + b_1) & \cdots & g(w_L \cdot x_N + b_L)
\end{bmatrix}, \quad \beta = \begin{bmatrix}
\beta_1^T \\
\vdots \\
\beta_L^T
\end{bmatrix}
\text{ and } T = \begin{bmatrix}
t_1^T \\
\vdots \\
t_N^T
\end{bmatrix}. \tag{2}
\]

The matrix \( H \) defined in Equation (2) is called output matrix of the hidden layer of the neural network; the \( i \)-th column of \( H \) is the \( i \)-th output vector of the hidden neuron with respect to the input \( (x_1, \ldots, x_N) \). The output weights are calculated analytically by \( \beta = H^\dagger T \), where \( H^\dagger \) is the Moore-Penrose generalized inverse of \( H \) and corresponds to the least squares solution for a regression problem [8]. In [9], have been proposed various ELM algorithms in base to the calculation of Moore-Penrose inverse matrices with the aim of achieving better results. In particular, in this article will take into account the ridge regression theory to add a positive value to the diagonal of the matrix \( H^T H \) to obtain the solution given by Equation (3) [10,11].

\[
\beta = \left( \frac{I}{C} + H^T H \right)^{-1} H^T T. \tag{3}
\]

The solution resulting by Equation (3) is more stable and tends to have better generalization results [10,11]. Figure 2(a) shows the schematic of an ELM network with the following summarized structure: the parameters \( (w_i, b_i) \) are randomly initialized, \( g_i(x) = g(w_i, b_i, x) \) is the activation function for each hidden neuron in terms of the input \( x \) and the output weights \( \beta \) are obtained by regularized least squares.

3.2. Multilayer extreme learning machines

In [12] they study the construction of a based on ELM autoencoder (ELM-AE), consists in altering the ELM to perform unsupervised learning as follows: the input data is used as output data, the weights and biases of the hidden layer are randomly generated and then orthogonalized through the Gram-Schmidt orthogonalization method. The ELM-AE projects the input data to a space of different or equal dimension, furthermore is the accountable of learning representations.
of characteristics via singular values. Figure 2(b) establishes the structure of the ELM-AE of summarized way: the output label is the same that the input \(x\), the parameters of the hidden layer \((a_i, b_i)\) are randomly initially generated and then are orthogonalized and \(g_i(x) = g(a_i, b_i, x)\) is the activation function for the hidden neuron \(i\)-th with respect to the input \(x\).

![Figure 2. Neural networks (a) ELM learning algorithm (b) ELM-based autoencoder.](image)

The multilayer extreme learning machine architecture is obtain by the stacking of ELM-AE [12]. ML-ELM hidden layer weights are initialized with ELM-AE through a unsupervised training by layers. The Equation (4) introduced in [12] establishes the general relationship for \(k\) hidden layers.

\[
H^k = g\left((\beta^k)^T H^{k-1}\right),
\]

where \(H^k\) is the output matrix of the hidden layer \(k\)-th, \(\beta^k\) are the output weights of the hidden layer \(k\)-th and \(g\) is the activation function that can be chosen as a linear function if the number of neurons \(L^k\) in the hidden layer \(k\)-th is equal to the number of neurons in the hidden layer \((k - 1)\)-th, otherwise, may be piecewise nonlinear. The input layer \(x\) can be considered as the hidden layer 0 \((k = 0)\). Finally, the weights between the hidden layer last and the output labels \(t\) are obtained by regularized least squares through supervised training. In particular, Figure 3 shows the construction of a neural network with three hidden layers by the autoencoders stacking, the process is as follows: \(\beta^1\) are weights of layer \(h^1\) obtained as output weights of ELM-AE with respect to the input data \(x\), \(\beta^k\) \((k = 2, 3)\) are weights of layer \(h^k\) obtained as weights of ELM-AE with respect to the output of layer \(h^{k-1}\) and the weights between the \(h^3\) layer and the \(t\) output labels are obtained by regularized least squares through supervised training.

![Figure 3. Extreme learning neural network with three hidden layers based on ELM-AE.](image)
4. Writer verification experiments using extreme learning machines
In this section is performed a numerical evaluation of the writer’s verification considering the simple graphemes “C”, “∩”, “∼”, “S” and “∪” of repository [5]. ELM and ML-ELM networks train themselves to classify the Local Binary Patterns of each of the simple graphemes of the 50 writers of repository. Specifically, the repository consists of 5 matrices, each matrix has 2500 rows that correspond to the 50 samples of the 50 writers. In addition, the matrices have 256 columns that correspond to the 256 elements of the LBP vector of each rectified image. The designed classifiers have 256 inputs and 50 outputs. The inputs correspond to the elements of the LBP and the outputs correspond to the 50 classes (1 class by person).

4.1. Individual characters
A series of experiments are conducted for each of the characters with the purpose of verify the effectiveness and efficiency of the ML-ELM framework in comparison with the ELM. Initially, we do a search of optimal parameters for the grapheme “∼” and through these results is done an analysis for the remaining characters. The selection of the ridge parameter $C$ for the ELM is set as $\{10^{-10}, 10^{-9}, \ldots, 10^{0}, 10^{10}\}$ and for the ML-ELM is done the variations with respect to the two hidden layers and the output layer to obtain a grid of order $9261 \times 3$. The sets $\{10^{-2}, 10^{-1}, 1\}$ and $\{(10^5, 10^8), (10^6, 10^{-8}, 10^4), (10^3, 10^7, 10^5)\}$ are parameters obtained with highest percentage of acceptance for the ELM and ML-ELM, respectively. The Figure 4(a) and Figura 4(b) shows the testing accuracy ELM and ML-ELM in function of the neurons number $L$, where the optimal parameters $C$ are fixed. We are observed that ML-ELM follows a convergence rule similar to ELM but with higher accuracy and fewer amount of hidden neurons.

![Figure 4](image.png)

**Figure 4.** Testing accuracy in $L$ subspace for ELM and LM-ELM: (a) accuracy of ELM in terms of $L$; (b) accuracy of ML-ELM in terms of $L$.

The Table 1 shows the acceptance rate and the testing time for each grapheme. The experiments were repeated 20 times and the results are consequence of an average value. It can be seen that ML-ELM obtains an improvement in compared with the ELM. The average success percentage with the higher number of neurons for the ML-ELM is 95.44 %, while for the ELM is 91.92 %.
Table 1. Accuracy and testing time comparison for individual characters.

| Dataset | $C = 10^{-2}$ | $C_1 = 10^5$, $C_2 = 10^5$, $C_3 = 10^4$ | $C_1 = 10^{-2}$ | $C_2 = 10$, $C_3 = 10^7$ |
|---------|---------------|----------------------------------|----------------|------------------|
| ~       | 91.45 0.1620  $L = 8000$ | 96.08 0.1083  $L_1 = L_2 = 1200$ |
| C       | 91.98 0.1975  $L = 10000$ | 96.40 0.1525  $L_1 = L_2 = 1500$ |
| C       | 93.16 0.2640  $L = 15000$ | 96.44 0.1903  $L_1 = L_2 = 1800$ |
| ∩       | 88.88 0.1655  $L = 8000$ | 94.93 0.1161  $L_1 = L_2 = 1200$ |
| ∩       | 89.82 0.2025  $L = 10000$ | 95.04 0.1854  $L_1 = L_2 = 1500$ |
| ∩       | 91.42 0.3035  $L = 15000$ | 95.05 0.2378  $L_1 = L_2 = 1800$ |
| ∪       | 89.41 0.1610  $L = 8000$ | 95.39 0.1924  $L_1 = L_2 = 1500$ |
| ∪       | 90.07 0.1880  $L = 10000$ | 95.59 0.2356  $L_1 = L_2 = 1800$ |
| ∪       | 91.38 0.2870  $L = 15000$ | 95.80 0.2730  $L_1 = L_2 = 2000$ |

4.2. Characters combination

In this subsection we are proposed the combination of characters in order to improve the people recognition. The training set consists of 10000 samples, while the testing set 2500 descriptors. The search of optimal parameters was done of similar way to the grapheme “~”, giving rise to the sets $\{10^{-2}, 10^{-1}, 1\}$ and $\{(10^2, 10, 10^{10}), (10^4, 10^{-3}, 10^7), (10^4, 10, 10^7)\}$ as parameters with higher accuracy for ELM and ML-ELM. The Figure 5(a) and Figure 5(b) shows the behavior of the hit percentage in terms of the neurons number $L$ for the ELM and ML-ELM, fixing the optimum parameters. It is observed that the accuracy percentage ML-ELM is more stable than the ELM in a wide range of $L$.

Figure 5. Testing accuracy in $L$ subspace for ELM and ML-ELM: (a) accuracy of ELM in function of $L$; (b) accuracy of ML-ELM in function of $L$. 

\[ \begin{align*}
\text{Testing Accuracy} & = \frac{\text{Number of correct classifications}}{\text{Total number of classifications}} \\
\text{Accuracy} & = \frac{\text{Correct responses}}{\text{Total responses}} \\
\text{Precision} & = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \\
\text{Recall} & = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \\
\text{F1 score} & = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\end{align*} \]
The testing results are shown in Table 2, it can be seen that the ML-ELM again achieves the highest accuracy in an accessible time with the least amount of neurons. The success percentage and time are consequence of an average value of 20 simulations for the parameter $C$ with higher accuracy, obtaining 92.19% with $L_1 = L_2 = 5000$ for ML-ELM and 89.86% with $L = 40000$ for ELM.

Table 2. Accuracy and testing time comparison for characters combination.

| Dataset                      | ELM (%) | Time | Neurons | ML-ELM (%) | Time | Neurons |
|------------------------------|---------|------|---------|------------|------|---------|
| All characters by person     | 88.70   | 2.0685 | $L = 25000$ | 91.07 | 0.5394 | $L_1 = L_2 = 1500$ |
|                              | 89.15   | 4.7575 | $L = 30000$ | 91.71 | 0.7269 | $L_1 = L_2 = 1800$ |
|                              | 89.50   | 4.7620 | $L = 35000$ | 91.74 | 0.8904 | $L_1 = L_2 = 2000$ |
|                              | 89.86   | 5.9565 | $L = 40000$ | 92.19 | 4.6605 | $L_1 = L_2 = 5000$ |

5. Conclusions
This study presented a new verification scheme of the writer through simple graphemes. To build the classifiers were adopted ELM neural networks, the which present a high adjustment level with low computational cost in the training. To determine the optimum architecture was evaluated a hyperparameters grid. The results show that the ML-ELM achieve a better performance than the ELM in the writer verification. Tests have shown that the ML-ELM descriptor and classifier achieve a success rate greater to the 95% for each one of the characters. In addition, it was presented a study for the simple Characters combination in order to improve the recognition capacity of the method, obtaining a accuracy percentage higher to the 91% for ML-ELM classifier.

References
[1] Kumar P, Sharma A 2020 Segmentation-free writer identification based on convolutional neural network Computers & Electrical Engineering 85 106707
[2] Vásquez J, Ravelo-García A, Alonso J, Dutta M, Travieso C 2018 Writer identification approach by holistic graphometric features using off-line handwritten words Neural Computing and Applications 32 15733
[3] Hafemann L, Sabourin R, Oliveira L 2017 Offline handwritten signature verification literature review Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA) (Montreal: IEEE)
[4] Aubin V, Mora M 2017 A new descriptor for person identity verification based on handwritten strokes off-line analysis Expert Systems With Applications 89 241
[5] Aubin V, Mora M, Santos-Peñas M 2018 Off-line writer verification based on simple graphemes Pattern Recognition 79 414
[6] Huang G, Zhu Q, Siew C 2004 Extreme learning machine: a new learning scheme of feedforward neural networks Neural Networks 2 985
[7] Ojala T, Pietikäinen M, Harwood D 1996 A comparative study of texture measures with classification based on featured distributions Pattern Recognition 29(1) 51
[8] Courrieu P 2008 Fast computation of Moore-Penrose inverse matrices Neural Information Processing - Letters and Reviews 8(2) 25
[9] Lu S, Wang X, Zhang G, Zhou X 2015 Effective algorithms of the Moore-Penrose inverse matrices for extreme learning machine Intelligent Data Analysis 19(4) 743
[10] Huang G, Zhou H, Ding X, Zhang R 2011 Extreme learning machine for regression and multiclass classification IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42(2) 513
[11] Hoerl A, Kennard R 1970 Ridge regression: Biased estimation for nonorthogonal problems Technometrics 12(1) 55
[12] Kasun L, Zhou H, Huang G, Vong C 2013 Representational learning with extreme learning machine for big data IEEE Intelligent Systems 28(6) 31