Neural Comput & Applic (2018) 29:151–172
https://doi.org/10.1007/s00521-016-2527-x

ORIGINAL ARTICLE

Language discrimination by texture analysis of the image corresponding to the text

Darko Brodić[1] · Alessia Amelio[2] · Zoran N. Milivojević[3]

Received: 7 March 2016 / Accepted: 8 August 2016 / Published online: 19 August 2016

Abstract The manuscript provides a novel method for language identification using the texture analysis of the script. The method consists of mapping each letter from the text with certain script type. It is made according to characteristics concerning the position of the letter in the baseline area. In order to extract features, the co-occurrence matrix is computed. Then, the texture features are calculated. Extracted measures show meaningful differences due to dissimilarities in the script and language characteristics. It represents a basis in a decision-making process of the language identification. Feature classification is performed by the extension of a state-of-the-art method called genetic algorithms image clustering for document analysis. The proposed method is tested on an example of documents given in English, French, Slovenian and Serbian languages and compared to other well-known classification methods and feature representations in the state of the art. The results of experiments show the superiority of the proposed approach.

✉ Darko Brodić
dbrodic@tf.bor.ac.rs

Alessia Amelio
aamelio@dimes.unical.it

Zoran N. Milivojević
zoran.milivojevic@vtsnis.edu.rs

[1] Technical Faculty in Bor, University of Belgrade, Vojske Jugoslavije 12, Bor 19210, Serbia

[2] DIMES University of Calabria, Via P. Bucci Cube 44, 87036 Rende, CS, Italy

[3] College of Applied Technical Sciences, Aleksandra Medvedeva 20, Nis 18000, Serbia

Keywords Coding · Classification · Language recognition · Pattern recognition · Statistical analysis · Genetic algorithm68T05 · 68U10 · 68U15

Abbreviations

ASR	Automatic speech recognition
CM	Confusion matrix
DE	Differential evolution
GA-IC	Genetic algorithms image clustering
GA-ICDA	Genetic algorithms image clustering for document analysis
GLCM	Gray-level co-occurrence matrix
GMM	Gaussian mixture model
HF	Hash function
IR	Information retrieval
LI	Language identification
NLP	Natural language processing
NMI	Normalized mutual information
OCR	Optical character recognition
PDF	Portable document format
PKC	Public key cryptography
PSO	Particle swarm optimization
SKC	Secret key cryptography
SOM	Self-organizing map
WOI	Window of interest

1 Introduction

1.1 Problem statements

Language identification (LI) represents the process of determining which language is given in the text. It generally represents an important preprocessing step in many specific tasks such as [1]:

- Identification of the language or encoding of the web pages,
- Text mining,
- Optical character recognition (OCR),
- Automatic speech recognition (ASR) and
- Information retrieval (IR) systems.

The types of text documents may be scanned or electronic documents (PDF, email messages, web pages, and other electronic archives). The volume growth of digital and digitalized documents is the main reason because this process represents an important research area.

LI is a fundamental requirement prior to any language-based processing. Typically, it is an initial step in a text processing system that may involve machine translation, semantic understanding, categorization, searching, routing or storage for IR [2]. Incorrectly identifying the language results in garbled translations, faulty or no information analysis and poor precision or recall in searching [3]. Existing methods can produce reasonable results, but they usually manage large lists and matrices. Hence, they are computationally expensive [4, 5]. Furthermore, the document type defines the suitability of the applied LI method. If the document is scanned, meaning that it is given as an image, then the script recognition techniques are usually applied. On the contrary, if an electronic document is under consideration, then the language recognition techniques are used. However, script recognition techniques are typically extended in order to be used for LI.

1.2 Related work

Many methods have been proposed in order to solve the LI problem. They can be grouped as follows:

- Word-based approach,
- *N*-gram approach,
- Markov model approach and
- Hybrid approach.

Word-based approach uses only words up to a specific length in order to make the language model. Sometimes, it is called short word-based approach [6]. The extension of this method is based on the most frequent word's appearance. It establishes the language model using the most frequent words only. These words represent the set of words, which has the highest frequency of appearance in the text [7].

Another technique establishes a language *n*-gram model [8]. *N*-grams are substrings, which are generated from a larger piece of text. Such a text is divided into smaller text parts, whose maximum length is *n*. Then, the frequency of characters' occurrence [8] or encoded bytes [9] is counted. Such a model is compared to the known language model. The language is identified if it is similar to the already known language model. Unfortunately, this approach requires a very large piece of text for training. Furthermore, it cannot solve the LI problem if the pieces of input text are composed of various languages, which is very common on the Web.

An interesting approach is given in [10], where a letter based scoring method is used. This method can be qualified as modified *n*-gram method, where *n* is equal to 1, i.e., unigram method. It calculates letter distributions of texts. Then, letter distributions are multiplied by average letter distributions of languages obtained from training set. Classification is made according to centroid values of the algorithm, which creates the text. It is performed to identify the language of each text document.

Techniques from the next group use Markov models in combination with Bayesian decision rules [9]. They are employed to produce models for each language in the training data. It is carried out by segmenting the strings and entering them into a transition matrix, which contains the occurrence probability of all character sequences. These language models are then used to determine the likelihood of test data generated by a particular model.

Also, an hybrid approach based on the selection of system elements by several classifiers and discriminant analysis is proposed [11]. Furthermore, it is improved by using methods for the estimation of robust regularized covariance matrix oriented to under-resourced languages and by adopting stochastic methods for speech recognition tasks. Still, the method is primarily oriented toward ASR. The latter differs from the language identification techniques because it is based on the phoneme instead of grapheme research. Similar approach is given in [12].

Among the others, the extended version of script recognition techniques connected to OCR is classified as hybrid method. An efficient technique for discrimination of Bangla and English scripts in multilingual documents is given in [13]. It is based upon the analysis of connected component profiles extracted from the scanned images.

Indian documents represent a good basis for script and language differentiation due to their mixing of script and language. For such a problem, a method which introduces two evaluation parameters, i.e., average accuracy rate and model building time is proposed [14]. Then, obtained features are used for script and language discrimination by well-known classifiers. The results are promising.

One of the most universal approaches, which can be used to differentiate Latin and non-Latin scripts as well as the languages in degraded document images is proposed in [15]. It performs document vectorization, which transforms the image into a vector. The vector characterizes the shape and frequency of the contained character and word images. Then, the scripts are identified by the density and distribution of vertical runs between character strokes and a

vertical scan line. Furthermore, the Latin-based languages are discriminated using a set of word shape codes, which are constructed by horizontal word runs and character extremum points.

Still, a method that combines OCR and LI approaches is proposed in [16]. It converts a document image into the character shape codes and word shape tokens. Using a natural language processing (NLP) method such as 3-gram applied to the character shape codes, it is able to discriminate 23 languages with overall accuracy >90 %.

Comprehensive examination of different LI methods is given in [17–20]. The approach in the last reference differs from the others, because it examines different LI techniques on the example of very short texts sometimes called query style texts. The results showed that Naive-Bayes classifiers perform the best in such circumstances. However, errors tend to occur within language when some text is given from the same language families.

1.3 Our approach

In this paper, we propose a novel algorithm for the language identification, which consists of the following steps: language modeling, statistical analysis of the texture, extraction of texture features, and classification of features and language discrimination. The core of the proposed approach is the original way of generating the features, which are then used for classification and language discrimination, as well as the extension of a classification technique of the state of the art which is suitable to the proposed features.

Script recognition methods extended for LI process work on scanned documents. They typically extract the features connected to OCR process, such as connected component profiles, shape of characters, density and distribution of vertical runs between character strokes and a vertical scan line. Hence, their application to electronic documents is limited.

Furthermore, NLP-based methods need a clearly identified text from document images. It means that the OCR should be performed prior to the process of language recognition. Actually, the OCR should recognize each character properly, because it is an input to the language recognition method. Hence, any degradation in the OCR process will lead to language recognition errors.

On the contrary, our approach is a universal one. Its versatility is connected with its application to either scanned or electronic documents. If we apply our method to the scanned documents, it is more robust to errors than other methods. It is mainly because it doesn't require each character to be properly recognized. Instead, it classifies each connected component (representing a character) according to a proposed set of four elements. In this way, it maps all connected components from a scanned document into four different elements creating a mapped image with four levels of gray. Furthermore, if it is applied to an electronic document, then each letter is mapped according to its unicode value to one of the four set elements. In this way, the initial unicode electronic document is mapped into the equivalent gray-scale image with the four levels as in the case of scanned documents. As a consequence, the number of variables is considerably reduced. It leads to a computer non-intensive algorithm. Up to now, our approach is similar to the one given in [16]. But, instead of applying typical NLP procedure to extract the features such as n-gram and then to discriminate the languages, we introduce the processing of the image by texture analysis. Hence, the mapped image is subjected to texture analysis extracting the features needed for classification. In this way, we used the co-occurrence combination of the aforementioned four elements. The main point represents the neighborhood's order of these elements creating a variety of combinations given by co-occurrence matrix with dimension 4×4. Accordingly, the vector of 12 co-occurrence first- and second-order features of the texture is extracted.

Then, an extension of a state-of-the-art classification tool called genetic algorithms image clustering for document analysis (GA-ICDA) is introduced to establish solid criteria for language discrimination and later identification. Considering the multiple and well-known advantages of the evolutionary strategies on the other optimization methods [21], the proposed tool uses a population-based genetic procedure as a basis for clustering, which is more robust to errors in the input than the other point-based optimization techniques [22]. In particular, it is to prefer to artificial neural networks, for which errors in the input data produce biased network parameters in the training phase. Although a robust cost function is introduced to mitigate the problem, it has not been totally solved, because the bias does not totally disappear [23]. For this reason, the application of a genetic procedure on a robust-to-noise document representation becomes of great importance in the context of scanned documents which are usually subjected to errors due to the OCR process. The new GA-ICDA clustering approach introduces three modifications, making the algorithm particularly apt to deal with language discrimination in text documents. This context is completely different from a context of pure image analysis. We do not present a new genetic clustering procedure for texture analysis optimization. Our aim consists in presenting an extension of a genetic approach previously adopted in image analysis and shows that it is able to successfully solve a language discrimination problem, as well as the utility of the introduced modifications in solving this task. Genetic algorithms have been successfully applied in many contexts of text document clustering [24–26]. Furthermore,

the advantages of the genetic approach with respect to other evolutionary approaches are demonstrated from the introduction of genetic operators in particle swarm optimization (PSO) strategy for improving the clustering results of PSO [27]. On the other hand, genetic procedure has been also used to improve the global search ability of the differential evolution (DE) algorithm by introducing the crossover in DE to explore more solutions in the search space and to avoid local minima [28].

The method is tested on a training and test databases of documents written in English, French, and closely related Slovenian and Serbian languages. From all aforementioned, the proposed approach is universal, and also can be used to distinguish among a wider set of languages or very similar languages.

The remainder of the paper is organized as follows. Sections 2, 3 and 4 provide detailed information on how the language discrimination problem is solved by using texture analysis and clustering. Specifically, Sect. 2 proposes the language model established according to the text line definition. Section 3 gives the elements of the co-occurrence analysis. Section 4 explains the classification algorithm. Section 5 defines the experiment. Section 6 shows the results of the experiment and discusses them. Furthermore, the criteria for language discrimination are proposed. Section 7 makes conclusions and points out further research work direction.

2 Language model

Cryptography is a mathematically oriented part of science, which uses cryptographic algorithms to convert ordinary data into unreadable ones [29]. There exist three different types of cryptographic algorithms [30]: (1) Secret key cryptography (SKC), (2) Public key cryptography (PKC) and (3) hash functions (HF).

In the proposed approach, HF is used. It encrypts data irreversibly, which means that original data cannot be retrieved from encrypted data [31]. In our case, the decryption of the cipher text is not significant. Also, HF creates shorter output than input, which is very suitable for deeper analysis due to lower computer time consuming.

In this paper, the cryptography is used solely as a basis for modeling and analyzing documents written in English, French, Slovenian and Serbian languages. Accordingly, HF is used for converting English, French, Slovenian and Serbian documents into a cipher. Then, the cipher is subjected to the co-occurrence analysis. It shows a noticeable feature distinction between cipher obtained from English, French, Slovenian and Serbian texts. At the end, the classification tool is used for language differentiation. Figure 1 shows the flow of the algorithm.

The text line in the printed documents is determined by four virtual reference lines establishing three vertical zones as in Fig. 2 [32].

All letters, diacritics, ligatures or signs can be classified on behalf of aforementioned zones. Hence, Fig. 2 illustrates script characteristics according to its position in the text line.

Accordingly, all letters are divided into four different script types: (1) short letter, (2) ascender letter, (3) descendent letter, and (4) full letter.

Taking into account the above classification, all letters from the alphabet can be replaced with the cipher. It is built in the manner like the HF, which can be given as:

$$f(x) = y, \text{ where } x \in X \text{ and } y \in Y \tag{1}$$

Thereby, X consists of the following elements:

$$X = \left\{ \begin{array}{l} a,\ c,\ e,\ i,\ m,\ n,\ o,\ r,\ s,\ u,\ v, \\ w,\ x,\ z,\ \text{œ},\ \text{æ},\ b,\ d,\ f,\ h,\ k,\ l, \\ t,\ A,\ B,\ C,\ D,\ E,\ F,\ G,\ H,\ I,\ J, \\ K,\ L,\ M,\ N,\ O,\ P,\ R,\ S,\ T,\ U,\ V, \\ W,\ X,\ Y,\ Z,\ \grave{a},\ \hat{a},\ \acute{e},\ \grave{e},\ \hat{e},\ \ddot{e},\ \hat{\imath}, \\ \ddot{\imath},\ \hat{o},\ \hat{I},\ \ddot{I},\ \hat{O},\ \grave{U},\ \hat{U},\ \ddot{U},\ \acute{Y},\ \acute{c},\ \check{c}, \\ \check{s},\ \check{z},\ đ,\ \acute{C},\ \check{C},\ \check{C},\ \check{Z},\ Đ,\ Æ,\ Œ,\ g, \\ j,\ p,\ q,\ lj,\ Lj,\ Nj \end{array} \right\} \tag{2}$$

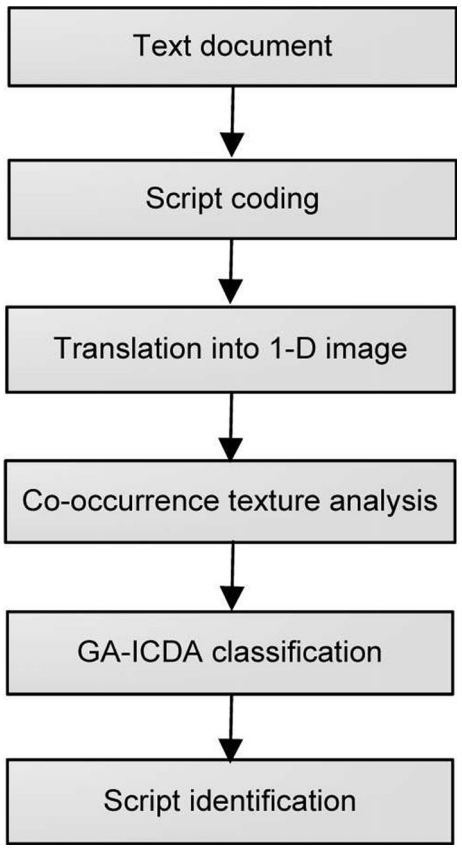


Fig. 1 Flow of the algorithm

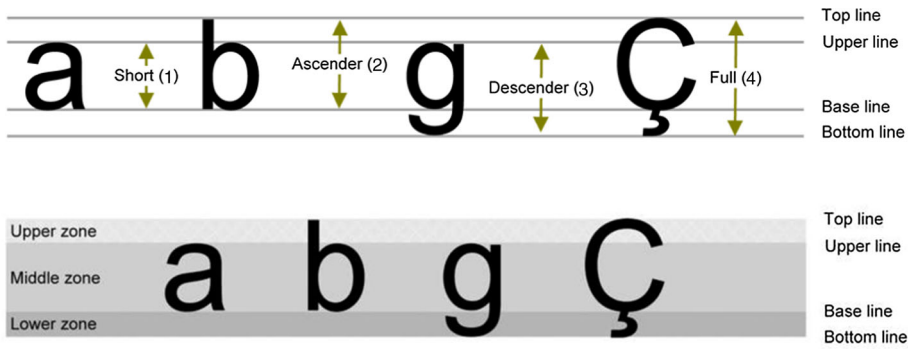


Fig. 2 Definition of the script characteristics: **a** four virtual reference lines, **b** three vertical zones

Table 1 English, French, Slovenian and Serbian alphabet mapping according to baseline characteristics

Script type	Letters, diacritics and ligatures
S(1)	Small letters: a, c, e, m, n, o, r, s, u, v, w, x, z
	Ligatures: æ, œ
A(2)	Small letters: i, b, d, f, h, k, l, t
	Capital letters: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, R, S, T, U, V, W, X, Y, Z
	Diacritics: à, â, é, è, ê, ë, î, ï, ô, Î, Ï, Ô, Ù, Û, Ü, Ÿ, ć, č, š, ž, đ, Ć, Č, Š, Ž, Đ
	Ligatures: Æ, Œ
D(3)	Small letters: g, p, q, y
	Diacritics: ç
	Ligatures: dž, Dž
F(4)	Capital letters: Q
	Small letters: j
	Diacritics: Ç, ÿ
	Ligatures: lj, Lj, nj, Nj

while Y is given as:

$$Y = \{1, 2, 3, 4\} \tag{3}$$

Each alphabet contains small and capital letters. English has 52 different letters, French exhibits 52 different letters with additional 28 diacritics and 4 ligatures, Slovenian contains 50 different letters and Serbian includes 60 different letters. All these elements are the members of X. Each letter occupies a certain zone(s) in text line area. Accordingly, each of them is replaced with the cipher, which is taken from the set of four counterparts given by Y [33]. The mapping is made according to Table 1. This way, original documents in English, French, Slovenian and Serbian are mapped into cipher text. Furthermore, each element in cipher, i.e., 0, 1, 2 and 3 is taken as the corresponding gray-scale level in the image. Hence, starting from the cipher text, a 1-D gray-scale image is created. This approach is very convenient because texture analysis can be applied to such an image in order to extract features. Also, the maximum number of different elements in the cipher is up to 4 [see Eq. (3)] leading to computer non-intensive texture analysis. Figure 3 illustrates the result of such mapping from cipher to an image on the example of English, French, Slovenian and Serbian documents.

3 Texture analysis

The co-occurrence probabilities provide a second-order method for generating texture features [34]. These texture features are extracted from an image in two steps. First, the pairwise spatial co-occurrences of pixels separated by a particular angle θ and distance d are tabulated using a gray-level co-occurrence matrix (GLCM). Second, the GLCM is used to calculate a set of scalar quantities that characterize different aspects of the underlying texture. The GLCM is a tabulation of how often different combinations of gray levels co-occur in an image or image section [34]. In this stage, it is necessary to point that in our problem, 1-D image is only considered [see Fig. 3]. It means that instead of 2-D image, we have one very long 1-D image. This 1-D image includes the information where is the end of some text row r_l and the beginning of the next text row r_{l+1}. Hence, it does not represent the text row information without any connection, i.e., completely separated text lines. Accordingly, our 1-D image mimics the elements of initial 2-D image.

Gray-scale image which will be under consideration is given as $\mathbf{I}$. It features M rows and N columns with a total number of grays T. The spatial relationship of gray levels in the image $\mathbf{I}$ is given by co-occurrence matrix (GLCM) $\mathbf{C}$.

Biscuits for breakfast containing more sugar and fat such as cornflakes or whole grain bread.

(a)

22111221 211 211122112 1112121213 1111 11311 112 212 1112 11 1111221211 11 12121 31121 21112

(b)

2211122121121112211211121212131111113111122121112111111221211111212131121211 12

(c)

(d)

Biscuits pour le petit déjeuner contenant plus de sucre et de graisse tels que céréales ou pain de grains entiers.

(e)

22111221 3111 21 31212 22411111 111211112 3211 21 11111 12 21 3112111 2121 311 12121211 11 3121 21 311211 1122111

(f)

2211122131112131212224111111112111123211211111112221311211212131112121211113121213112111122111

(g)

(h)

Piškoti za zajtrk vsebujejo več sladkorja in maščob kot na primer koruzni kosmiči ali polnozrnat kruh.

(i)

2222122 11 114212 111214141 112 121221141 21 112212 212 11 312111 2111112 2111222 122 3121111112 2112

(j)

22221221111421211121414111212122114121112212212113121112111112211122212231211111122112

(k)

(l)

Keks za doručak sadrži više šećera i masti nego kukuruzne pahuljice ili integralni hleb.

(m)

2121 11 2111212 112122 1221 212111 2 11122 1131 212111111 31214211 222 2121311212 2212

(n)

212111211121211212212212121112111221131212111111312142112222121311212 2212

(o)

(p)

Fig. 3 Mapping and translating into 1-D image: **a**–**d** English document, **e**–**h** French document, **i**–**l** Slovenian document and **m**–**p** Serbian document

GLCM **C** is a $T \times T$ square matrix. To compute GLCM **C**, a central pixel $I(x, y)$ with a neighborhood defined by the Window of interest (WOI) has been taken. WOI is defined by inter-pixel distance d and orientation θ. If d is equal to 1, then it means 1 pixel distance, i.e., first neighbor pixel. Furthermore, if 8-connected neighborhood is considered, then the angles $\theta = 0°, 45°, 90°, 135°$ are possible. For the given image **I**, GLCM **C** is defined as [35]:

$$C(i,j) = \sum_{x=1}^{T}\sum_{y=1}^{T}\begin{cases} 1 & \text{if } I(x,y) = i, \text{and } I(x+\Delta x, y+\Delta y) = j, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where i and j are the intensity values of the image $\mathbf{I}$, x and y are the spatial positions in the image $\mathbf{I}$, the offset $(\Delta x, \Delta y)$ is the distance between the pixel-of-interest and its neighbor. It should be noted that offset depends on the direction θ that is used and the distance d at which the matrix is computed. Taking into account the nature of text, the neighborhood is given as 2-connected. Accordingly, θ is given as 0°, while d is typically used as first neighborhood, i.e., $d = 1$. Also, T represents the number of gray levels which is given as 4. This leads to quite small dimension of GLCM, i.e. 4×4. Hence, the calculation of GLCM is not computer time intensive.

Normalized matrix $\mathbf{P}$ obtained from the GLCM $\mathbf{C}$ is calculated as [36]:

$$P(i,j) = \frac{C(i,j)}{\sum_{i=1}^{T}\sum_{j=1}^{T} C(i,j)}. \tag{5}$$

Furthermore, using Eq. (5) mean values μ_x and μ_y and standard deviations σ_x and σ_y (of $\mathbf{P}$) are calculated as [34, 37]:

$$\mu_x = \sum_{i=1}^{T} i \sum_{j=1}^{T} P(i,j), \tag{6}$$

$$\mu_y = \sum_{j=1}^{T} j \sum_{i=1}^{T} P(i,j), \tag{7}$$

$$\sigma_x = \sqrt{\sum_{i=1}^{T}(i-\mu_x)^2 \sum_{j=1}^{T} P(i,j)}, \tag{8}$$

$$\sigma_y = \sqrt{\sum_{j=1}^{T}(j-\mu_y)^2 \sum_{i=1}^{T} P(i,j)}. \tag{9}$$

These texture semi-features will be used in forthcoming statistical analysis as well. While GLCM provides a quantitative description of a spatial pattern, it is too unwieldy for practical image analysis. A set of scalar quantities for summarizing the information contained in a GLCM is proposed in [34]. Although, a total of 14 quantities, i.e., features were originally proposed, only subsets of them are used [38]. They are the following eight GLCM-derived features of interest: (1) correlation, (2) energy, (3) entropy, (4) maximum, (5) dissimilarity, (6) contrast, (7) inverse difference moment and (8) homogeneity.

Correlation measures the linear dependency of gray levels on those of neighboring pixels. It is calculated as:

$$\text{Correlation} = \sum_{i=1}^{T}\sum_{j=1}^{T} \frac{(i\cdot j)\cdot P(i,j) - (\mu_x \cdot \mu_y)}{\sigma_x \cdot \sigma_y}. \tag{10}$$

Energy measures the textural uniformity of the pixel pair repetitions. Hence, it detects disorders in textures. Energy can reach the maximum value equal to one. High energy values occur when the gray-level distribution has a constant or periodic form. Energy is defined as:

$$\text{Energy} = \sum_{i=1}^{T}\sum_{j=1}^{T} P(i,j)^2. \tag{11}$$

Entropy measures the disorder or complexity of an image. It reaches large values when the image is not texturally uniform or when many GLCM elements have very small values. Complex textures tend to have high entropy. Entropy is strongly, but inversely correlated to energy. Entropy is given as:

$$\text{Entropy} = -\sum_{i=1}^{T}\sum_{j=1}^{T} P(i,j)\cdot \log P(i,j). \tag{12}$$

Maximum (probability) extracts the most probable difference between gray-scale values in pixels. It is calculated as:

$$\text{Maximum} = \max\{P(i,j)\}. \tag{13}$$

Dissimilarity measures the variation in gray-level pairs of the image. It depends on distance from the diagonal weighted by its probability. Dissimilarity is given as:

$$\text{Dissimilarity} = \sum_{i=1}^{T}\sum_{j=1}^{T} P(i,j)\cdot |i-j|. \tag{14}$$

Contrast measures the spatial frequency of an image. It is the difference between the highest and the lowest values of a contiguous set of pixels. Hence, it measures the amount of local variations present in the image. A low-contrast image presents GLCM concentration term around the principal diagonal and features low spatial frequencies. Contrast is defined as:

$$\text{Contrast} = \sum_{i=1}^{T}\sum_{j=1}^{T} P(i,j)\cdot (i-j)^2. \tag{15}$$

Inverse difference moment (Invdmoment) measures a sort of image homogeneity. Due to the weighting factor $1/(1+(i-j)^2)$ it gets only small contributions from the areas in the image that are inhomogeneous $(i-j)$. It has a low value for inhomogeneous images, while a relatively higher value for homogeneous images. It is given as:

$$\text{Invdmoment} = \sum_{i=1}^{T}\sum_{j=1}^{T} \frac{1}{1+(i-j)^2} P(i,j). \tag{16}$$

Homogeneity measures the image homogeneity as it assumes larger values for small gray tone differences in pair elements. It is more sensitive to the presence of near diagonal elements in the GLCM. It has maximum value when all elements in the image are the same. Its definition is given as:

$$\text{Homogeneity} = \sum_{i=1}^{T}\sum_{j=1}^{T} P(i,j)^2. \tag{17}$$

GLCM contrast and homogeneity are strong, but inversely, correlated in terms of equivalent distribution in the pixel pair's population. It means homogeneity decreases if contrast increases while energy is held constant. At the final stage, the energy and contrast are the most significant parameters in terms of visual assessment and computational load to discriminate between different textural patterns.

4 Classification

The GLCM features are used as a basis for the automatic language discrimination of documents using a classifier method. Here, we propose a new approach for the document classification in different languages which is called genetic algorithms image clustering for document analysis (GA-ICDA). This algorithm is an extension of the genetic algorithms image clustering (GA-IC) method [39] for the clustering of image databases. GA-ICDA is used for clustering the documents in different classes, each representing the language (i.e., English, French, Slovenian, Serbian) of the text inside the document. Starting from the traditional GA-IC technique, three main modifications are introduced in terms of feature representation, graph construction and cluster detection.

Differently from [39] where images are processed, here we have documents. Each document is represented as a feature vector codifying only texture content. In fact, each vector is composed of the GLCM-derived features of interest, obtained from the previously considered statistical analysis.

Similarly to [39], the document database is modeled as a weighted graph $G = (V, E, W)$, where V are the n nodes of the graph, E are the m edges among the nodes and W are the weights associated to the edges. A single document is a graph node. In GA-IC algorithm, an edge between two nodes p and q is present only if the two corresponding images are "quite" similar to each other. Here, we extend this concept by introducing an ordering between the nodes. Consequently, an edge between two nodes p and q is allowed not only if the two corresponding documents are similar, but also if p and q are not far to each other with respect to the given ordering. The weight on the edge $\langle p, q\rangle$ realizes the strength of the similarity between p and q.

The document graph is built in different steps. First of all, the L_1 distance is computed for each pair of document feature vectors. This phase is useful to realize the distance matrix of the documents. After that, from the obtained distance values, the similarity scores are computed, like in [39]. In particular, given two documents α and β, the similarity between them is:

$$w(\alpha,\beta) = e^{-\frac{d(\alpha,\beta)^2}{a^2}}, \tag{18}$$

where $d(\alpha,\beta)$ is the L_1 distance between α and β and a is a local scale parameter, computed as the average L_1 distance of each document to its first nearest neighbor document. For each document α, the similarity is evaluated only between α and its k h-nearest neighbors $nn_\alpha^h = \{nn_\alpha^h(1), nn_\alpha^h(2), \ldots, nn_\alpha^h(k)\}$, otherwise it is zero. The h-nearest neighbors of α represent those documents exhibiting the h-lowest distance values from α inside the distance matrix, where h is a parameter. Recall from [39] that the number k of h-nearest neighbors can be greater than h. At the end of this step, the similarity matrix for the documents is obtained. This sparse matrix represents the adjacency matrix of the document database graph. In fact, each similarity score $w(\alpha,\beta)$ computed between the document α and its h-nearest neighbor β traces a weighted link between α and β.

In GA-IC, this kind of graph exhibits stronger components with dense intra-connections, representing groups of very similar images. These components are linked to each other by a small number of inter-connections. However, when the documents are the nodes of the graph, fixing a high h-neighborhood of the nodes causes the introduction of noisy edges, due to the complexity of the document representation. They are edges connecting nodes not strongly similar to each other. On the other hand, choosing an extremely low number of h-neighbors for the nodes determines the loss of the document graph components. In order to overcome this limitation, we add another criterion of neighborhood selection, derived from the concept of Matrix Bandwidth [40], to break edges linking nodes too far to each other with respect to an established node ordering.

Consider the vertex ordering induced by the graph adjacency matrix. It is a one-to-one function mapping the nodes of the graph to integers $f : V \to \{1, 2, \ldots, n\}$. Let $f(v)$ be the label of the node $v \in V$, where each node has been assigned to a different label. For each node $v \in V$, we calculate the difference between $f(v)$ and the labels $F = \{f(nn_v^h(1)), f(nn_v^h(2)), \ldots, f(nn_v^h(k))\}$ of its adjacent nodes corresponding to the k h-nearest neighbors. This difference is expressed as:

$$\forall v \in V, \left|f(v) - f(nn_v^h(z))\right|, 1 \le z \le k. \quad (19)$$

Then, for each node v, we choose to eliminate those edges between v and its adjacent nodes whose corresponding label difference is greater than or equal to a threshold value Γ. Consequently, given two documents α and β, the element $w_{\alpha,\beta}$ inside the graph adjacency matrix becomes:

$$w_{\alpha,\beta} = \begin{cases} e^{-\frac{d(\alpha,\beta)^2}{a^2}} & \text{if } \beta \in nn_\alpha^h, \alpha \neq \beta, |f(\alpha) - f(\beta)| < \Gamma \\ 0 & \text{otherwise .} \end{cases} \quad (20)$$

Thus, if the documents α and β have a high distance $d(\alpha, \beta)$ to each other, the similarity score $w_{\alpha,\beta}$ between them will be quite low. On the contrary, if the two documents have a small distance $d(\alpha, \beta)$ between them, provided that they are near to each other with respect to the given node ordering ($|f(\alpha) - f(\beta)| < \Gamma$), their similarity score $w_{\alpha,\beta}$ will be high.

Clustering of the documents is performed by finding the dense node components inside the graph with intra-edges exhibiting high similarity weight. In order to find the graph components, we adopt the genetic algorithm used in [39]. Each individual I represents a possible solution to clustering problem, which is a partition of nodes in clusters, and consists of n genes, $\{g_1, g_2, \ldots, g_n\}$. A gene g_i represents a node, while its value is one of the node neighbors. Initialization randomly associates to each gene g_i in I a value corresponding to one of its neighbors. Uniform crossover and mutation are used as variation operators. Let I_1 and I_2 be two parent individuals. Uniform crossover determines a child individual I_3 by adopting a binary mask. Each gene g_i in I_3 will contain the value of the corresponding gene g_i in I_1 or I_2, based on the binary mask value at gene position i. Given an individual I, mutation randomly selects a gene g_i from I and modifies its value with a randomly selected neighbor. The employed fitness function is the weighted modularity, based on the edge weights in G. Genetic algorithm initializes the individuals, uses the variation operators and computes the fitness function on individuals. Procedure is repeated for a fixed number of generations. At the end, the individual $I_{\max}$ with the best weighted modularity, corresponding to the best partitioning of nodes in clusters, is chosen.

However, due to the complexity of the language discrimination, in order to mitigate the problem of local optima, we extend the clustering by a refinement step at the end of the genetic procedure. The clusters produced from the genetic algorithm are considered as initial clusters. Starting from these clusters, we apply a merging of cluster pairs until a fixed cluster number is reached. For each cluster s_i, the distance values between s_i and the other clusters $\{s_j\}, j = 1, \ldots, d, j \neq i$ are computed, where d is the number of clusters detected from the genetic algorithm. Then, the two clusters s_i and s_h with the minimum mutual distance are merged in a single cluster. Given two clusters $s_i = \{s_i^1, \ldots, s_i^a\}$ and $s_h = \{s_h^1, \ldots, s_h^b\}$, composed of a and b number of document feature vectors, the distance $D(s_i, s_h)$ between them is evaluated as the L_1 distance between those two document feature vectors, one for each cluster, which are farthest away from each other:

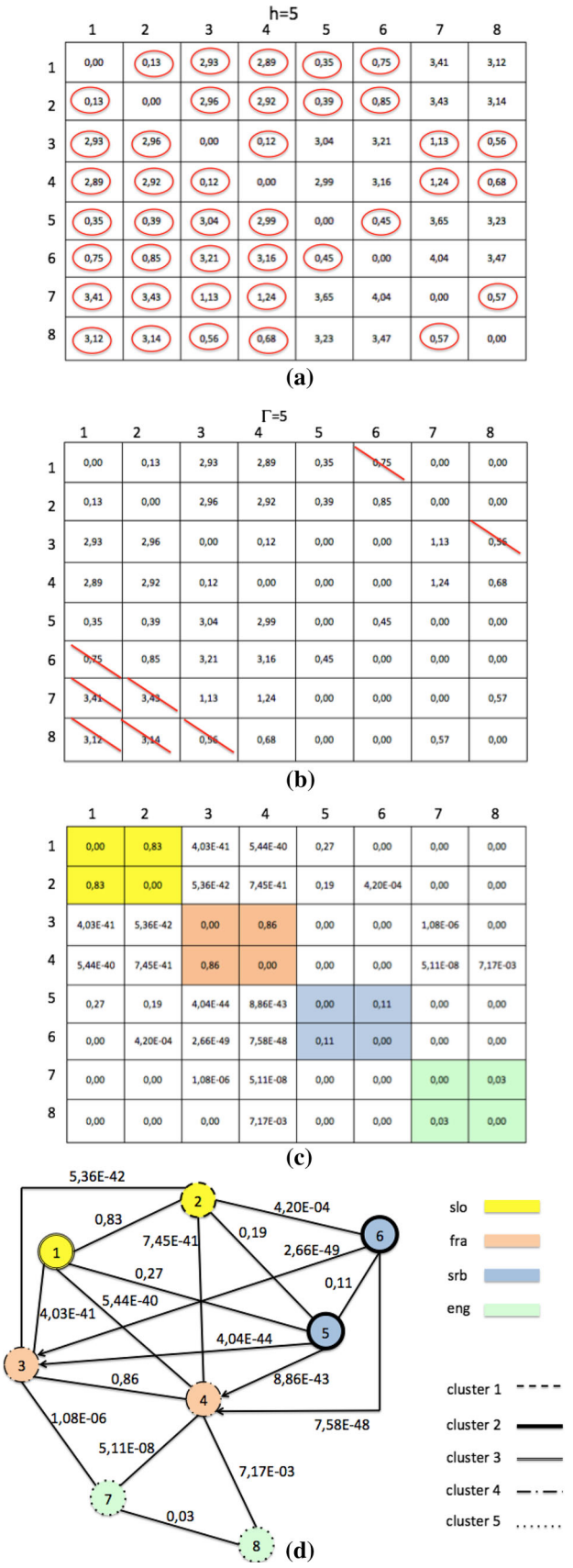
$$D(s_i, s_h) = \max_{s_i^x, s_h^y} \{d(s_i^x, s_h^y) | s_i^x \in s_i, s_h^y \in s_h\}, \quad (21)$$

where $d(s_i^x, s_h^y)$ is the L_1 norm between the two document feature vectors s_i^x and s_h^y.

We run an experiment for finding the best Γ, h and genetic parameters on the training database in the given languages English, French, Serbian and Slovenian. Because the maximum neighborhood size and label difference are limited from the size of the database, it is not possible to learn absolute values of the h and Γ parameters. Nonetheless, we evaluated the classification accuracy of the method at different combinations of the parameters. Results of this experiment confirm that, at a fixed value of the genetic parameters, the classification accuracy reaches the best value when Γ and h are in a range between $0.15 \times n$ and $0.65 \times n$, where n is the number of graph nodes, representing also the number of the documents in the database. Hence, we selected the genetic parameter values determining the best accuracy in that range. Also, we fixed h and Γ in that range.

4.1 Example

Figure 4 shows an example of graph construction in GA-ICDA. It is performed on a toy database composed of 8 documents: 2 out of 8 are given in Slovenian (1 and 2), 2 out of 8 in French (3 and 4), 2 out of 8 in Serbian (5 and 6) and 2 out of 8 in English languages (7 and 8). According to the aforementioned parameter tuning, the h and Γ parameters have been fixed to 5. Starting from the distance matrix (Fig. 4a) where the L_1 distance is computed for each pair of documents, the h-nearest neighbors are computed for each document. For example, in correspondence with document 1, $nn_1^5 = \{2, 3, 4, 5, 6\}$, because they are the documents whose distance is included in the 5 lowest distance values from document 1. From the 5-nearest neighbors of each document, only the spatially closest ones with respect to Γ threshold are selected (Fig. 4b). For each document i, they are the only documents whose label difference with i is <5. For example, considering document 7 with 5-nearest neighbors $nn_7^5 = \{1, 2, 3, 4, 8\}$, we have that first and second neighbors are eliminated from the matrix (documents 1 and 2), because the label difference between 7 and 1 is 6



◀**Fig. 4** Example of graph construction for a toy database of eight documents in Slovenian, French, Serbian and English languages: **a** h-nearest neighbors selection, **b** Γ closest neighbors selection, **c** adjacency matrix construction, and **d** corresponding weighted graph definition

Table 2 Distance values $D(s_i, s_h)$, $i, h = 1 \ldots 5$, computed between the clusters $s_1, \ldots, s_5$ detected from the genetic procedure

	s_1	s_2	s_3	s_4	s_5
s_1	0.00	**0.13**	2.93	0.75	3.41
s_2	**0.13**	0.00	2.96	0.85	3.43
s_3	2.93	2.96	0.00	3.21	1.24
s_4	0.75	0.85	3.21	0.00	4.04
s_5	3.41	3.43	1.24	4.04	0.00

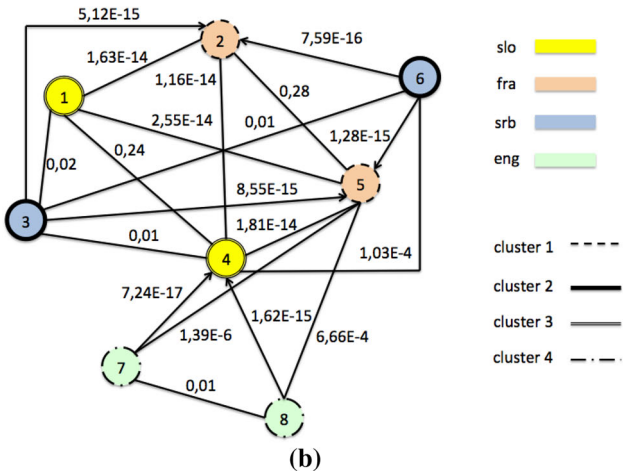
Bold value corresponds to the minimum value inside the table

and between 7 and 2 is 5, which are greater than or equal to $\Gamma = 5$. Similarity values are computed from the distance values corresponding to remaining neighbors (Fig. 4c), determining the adjacency matrix of a document database graph G (Fig. 4d). The true division of the documents in languages is colored and clearly visible inside the adjacency matrix and in G. Also, looking at the similarity values of the adjacency matrix, it is clear that the highest values are exhibited between the documents of the same language (1 and 2, 3 and 4, 5 and 6, 7 and 8), but that also high values are present between documents in Slovenian and Serbian languages (between 5 and 1, and 5 and 2, with values of 0.27 and 0.19). It demonstrates that discrimination of these languages is in general problematic. However, the value is higher between documents 1 and 2, which are both in Slovenian language, demonstrating the efficacy of the feature representation.

Considering the graph G in Fig. 4d, one of the solutions determined from the genetic algorithm is composed of 5 document clusters, $s_1 = \{1\}$, $s_2 = \{2\}$, $s_3 = \{3, 4\}$, $s_4 = \{5, 6\}$, $s_5 = \{7, 8\}$ (Fig. 4d). It is clear that it is not the correct solution, because Slovenian documents are split into two different clusters s_1 and s_2. Consequently, the merging procedure is applied on the found solution to determine the final solution of the correct 4 clusters. Table 2 reports the distance values $D(s_i, s_h)$ $i, h = 1, \ldots, 5$, computed for each cluster pair. For example, considering clusters s_3 and s_4, the distance $D(s_3, s_4)$ is computed by considering all the distance values between 3 and 5,6 and between 4 and 5,6 inside the original distance matrix (Fig. 4a). The final value is 3.21, because it corresponds to the distance between the two documents 3 and 6 which are the farthest away to each other. In Table 2, we observe that the minimum distance value is 0.13 (highlighted in bold) in correspondence of s_1 and s_2. Consequently, theyare merged

	1	2	3	4	5	6	7	8
1	0,00	1,63E-14	0,02	0,24	2,55E-14	0,00	0,00	0,00
2	1,63E-14	0,00	0,00	1,16E-14	0,28	0,00	0,00	0,00
3	0,02	5,12E-15	0,00	0,01	8,55E-15	0,01	0,00	0,00
4	0,24	1,16E-14	0,01	0,00	1,81E-14	1,03E-4	0,00	0,00
5	2,55E-14	0,28	0,00	1,81E-14	0,00	0,00	1,39E-6	6,66E-4
6	0,00	7,59E-16	0,0073	1,03E-4	1,28E-15	0,00	0,00	0,00
7	0,00	0,00	0,00	7,24E-17	1,39E-6	0,00	0,00	0,01
8	0,00	0,00	0,00	1,62E-15	6,66E-4	0,00	0,01	0,00

(a)



(b)

Fig. 5 Example of graph construction for a toy database of 8 documents in Slovenian, French, Serbian and English languages, when documents are not located consecutively in the distance matrix: **a** adjacency matrix construction and **b** corresponding weighted graph definition

in a single cluster including documents 1 and 2. Because the number of new obtained clusters is 4, the algorithm is terminated. The final clustering solution corresponds to the correct division of the documents in languages.

Figure 5 shows the graph construction when not all the documents are located consecutively in the distance matrix. It is performed on the same toy database composed of 2 Slovenian documents (1 and 4), 2 French documents (2 and 5), 2 Serbian documents (3 and 6) and 2 English documents (7 and 8). Figure 5a shows the obtained adjacency matrix derived from thresholding of the distance matrix by h-nearest neighborhood with $h = 5$ and by fixing the Γ parameter equal to 5, and by detecting the similarity values from the corresponding distance values. The true division of the documents in language classes is colored and visible inside the matrix. It is worth to observe that also in this case, the documents given in the same languages exhibit the highest similarity and that discrimination between Serbian and Slovenian languages is critical one. Hence, the adjacency matrix maintains the same properties as in the case of consecutive documents. Figure 5b illustrates the corresponding graph. In this case, the genetic algorithm determines the correct 4 clusters of documents, $s_1 = \{1, 4\}$, $s_2 = \{2, 5\}$, $s_3 = \{3, 6\}$, $s_4 = \{7, 8\}$. Consequently, the merging procedure will return this same solution of the 4 clusters.

5 Experiments

To test the proposed approach, the experiment is conducted on two custom-oriented databases of documents given in French, English, Slovenian and Serbian languages, extracted mainly from the web. Only few of them are printed and then scanned. French documents are extracted from Le Point Paris Match [41]. Furthermore, a part of English, Slovenian and Serbian documents are obtained from the web as well. However, for a few documents in different languages we give the equivalent translated ones. In this way, the database consists of original documents in different languages as well as the equivalent documents obtained from computer translator and also few scanned documents.

The first database consists of 85 documents, where 18 out of 85 are English documents, 10 are French documents, 32 and 25 are Serbian and Slovenian documents, respectively. English documents count from 600 to 3200 characters, while French documents consist from 800 to 4000 characters. Slovenian documents include from 600 to 3600 characters, while Serbian documents have from 600 to 3800 characters. It should be noted that Slovenian and Serbian languages are so-called closely related ones, which means that their discrimination is problematic. Furthermore, the co-occurrence analysis is sensitive to the low number of samples, which in our case represent characters. Hence, the documents of more than 500 characters can be used as an adequate means for valid statistical analysis, which is a premise of the factor analysis [42–44].

Then, the effectiveness of the proposed language identification tool is evaluated on the test database. It consists of 9 English documents, 3 French documents, and respectively, 6 and 5 Serbian and Slovenian documents, for a total of 23 documents. The documents count from 500 to 1000 characters.

6 Results evaluation

6.1 Analysis of the feature values and their discriminant capability

The results of the experiment that includes the four GLCM semi-features obtained from the database are given in Table 3.

Table 3 Four GLCM semi-features for English, French, Serbian and Slovenian documents from the database

GLCM semi-features	μ_x	μ_y	σ_x	σ_y
English				
Min	1.6705	1.6691	1.0101	1.0110
Max	1.7961	1.7961	1.0755	1.0764
Mean	1.7328	1.7320	1.0440	1.0439
French				
Min	1.6007	1.5999	0.9904	0.9901
Max	1.6634	1.6618	1.0303	1.0298
Mean	1.6293	1.6284	1.0081	1.0078
Serbian				
Min	1.3666	1.3657	0.6152	0.6151
Max	1.4382	1.4375	0.7141	0.7140
Mean	1.4033	1.4019	0.6586	0.6601
Slovenian				
Min	1.4013	1.3863	0.6489	0.6488
Max	1.4614	1.4608	0.7120	0.7120
Mean	1.4338	1.4315	0.6772	0.6768

Figure 6 illustrates the values of four GLCM semi-features: μ_x, μ_y, σ_x, and σ_y, according to their maximum, minimum and mean values.

Interpreting the results of four GLCM semi-features μ_x, μ_y, σ_x, and σ_y, it is easy to distinguish Slavic (Slovenian and Serbian) from non-Slavic languages (English and French). Furthermore, English and French show diversity in these features as well, which simplify their separation. However, closely related languages such as Serbian and Slovenian are characterized with overlapped GLCM semi-features. This leads to more complex process of differentiation.

Table 4 gives the results for eight GLCM full-features obtained from the database.

Figures 7 and 8 present eight GLCM full-features: energy, entropy, maximum (maxi), dissimilarity, contrast, inverse difference moment (invdmoment), homogeneity and correlation, according to their maximum, minimum and mean values.

The values of eight full GLCM features obtained from the database are commonly overlapped between languages. However, there exists some exception. In this way, the acquired results from French documents can be distinguished by diverse energy, entropy and maxi values. Furthermore, the Slovenian and Serbian languages can be easily separated from other languages by dissimilarity, inverse difference moment, contrast and correlation. However, closely related languages such as Serbian and Slovenian are very difficult to distinguish without the strong classification algorithm.

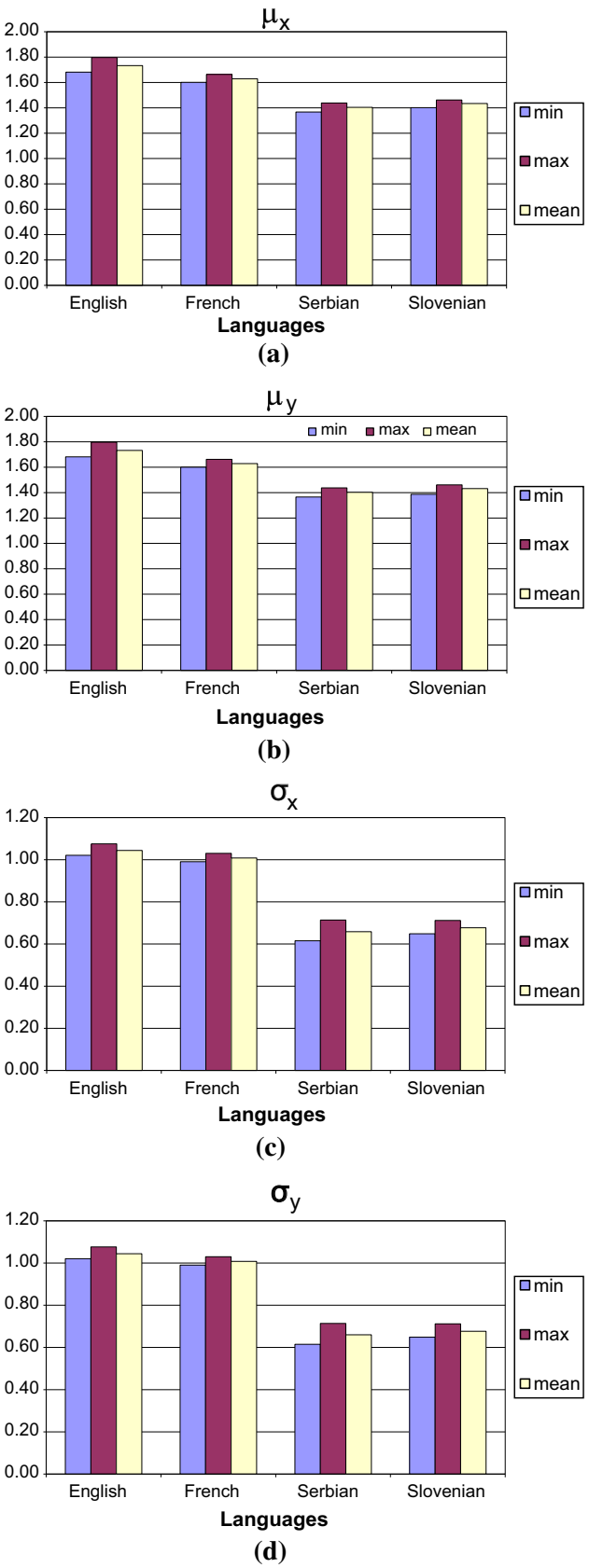


Fig. 6 Four GLCM semi-features for English, French, Serbian and Slovenian documents obtained from the database: **a** μ_x, **b** μ_y, **c** σ_x, and **d** σ_y

Table 4 Eight GLCM features for English, French, Serbian and Slovenian documents obtained from the database

GLCM features	Energy	Entropy	Maxi	Dissim.	Contr.	Invdm.	Homog.	Corr.
English								
Min	0.2454	−1.6475	0.3898	0.9385	2.0532	0.5882	0.6549	−0.1055
Max	0.2890	−1.5180	0.4636	1.0867	2.4019	0.6422	0.7005	0.0498
Mean	0.2695	−1.5730	0.4302	1.0258	2.2559	0.6101	0.6729	−0.0348
French								
Min	0.2928	−1.5151	0.4684	0.9300	2.0479	0.6141	0.6768	−0.1130
Max	0.3270	−1.4245	0.5127	1.0324	2.3350	0.6468	0.7038	−0.0122
Mean	0.3101	−1.4668	0.4901	0.9846	2.2024	0.6295	0.6898	−0.0838
Serbian								
Min	0.2395	−1.6726	0.3777	0.6593	0.9201	0.6431	0.6685	−0.2921
Max	0.2835	−1.5085	0.4425	0.8037	1.2756	0.6965	0.7134	−0.1944
Mean	0.2608	−1.5896	0.4128	0.7330	1.0863	0.6688	0.6899	−0.2529
Slovenian								
Min	0.2252	−1.7172	0.3333	0.6963	1.0662	0.6281	0.6507	−0.2908
Max	0.2734	−1.5657	0.4379	0.8216	1.3064	0.6673	0.7053	−0.2107
Mean	0.2419	−1.6574	0.3840	0.7695	1.1607	0.6524	0.6769	−0.2545

Table 5 shows the maximum dispersion of four GLCM semi-features obtained from the database. The French is characterized by the smallest average dispersion of four GLCM semi-features.

Table 6 shows the maximum dispersion of eight GLCM full-features obtained from the database. Again, French language is characterized by the smallest average dispersion of eight GLCM full-features.

The results of the experiment showing the values of four GLCM semi-features obtained from the test database are given in Table 7.

Similarly as in the database, it is easy to disseminate Slavic (Slovenian and Serbian) from non-Slavic languages (English and French) using the four GLCM semi-features μ_x, μ_y, σ_x, and σ_y. Furthermore, English and French can be easily distinguished by these measures. In contrast, closely related languages such as Serbian and Slovenian are characterized with slight overlapping GLCM semi-features. These results are similar to those previously obtained by the database.

Table 8 gives the results for eight GLCM full-features obtained from the test database.

The values of eight full GLCM features obtained from the test database show that tested languages cannot be easily discriminated. However, there exist measures that easily distinguish one language from others. Accordingly, all languages can be discriminated by energy values. Furthermore, Slavic languages can be separated from others by energy, entropy, maximum, dissimilarity, contrast and inverse difference moment values. French language can be distinguished from English one by energy, entropy and maximum values. Documents written in Slovenian are characterized by slightly lower values of energy than Serbian text. Nonetheless, closely related languages such as Serbian and Slovenian have values that are mainly overlapping. Hence, they are difficult to be distinguished without the strong classification algorithm.

Table 9 shows the maximum dispersion of four GLCM semi-features obtained from the test database.

Again, French is characterized by the smallest average dispersion of GLCM semi-features.

Table 10 shows the maximum dispersion of eight GLCM full-features obtained from the test database. Currently, Serbian language is characterized by the smallest average dispersion of eight GLCM full-features followed by the French language.

Taking into account all aforementioned, it is clear that the language differentiation is a challenging task. Hence, it is very difficult to establish a linear discrimination to separate each of them like in the script recognition [45]. It is even more complex if closely related languages are considered. Consequently, another experiment has been performed on the database by using the strong classification algorithm GA-ICDA.

6.2 Language identification results

In order to evaluate the classification performances, we use the traditional precision/recall and the normalized mutual information (NMI) measures. Precision, recall and f-measure are obtained from the confusion matrix **CM** representing the documents correctly or incorrectly classified in the different languages. In our case, the rows of the confusion matrix are the ground-truth languages, while the columns are the clusters found from the classification procedure. Each element $CM(i, j)$ inside the confusion

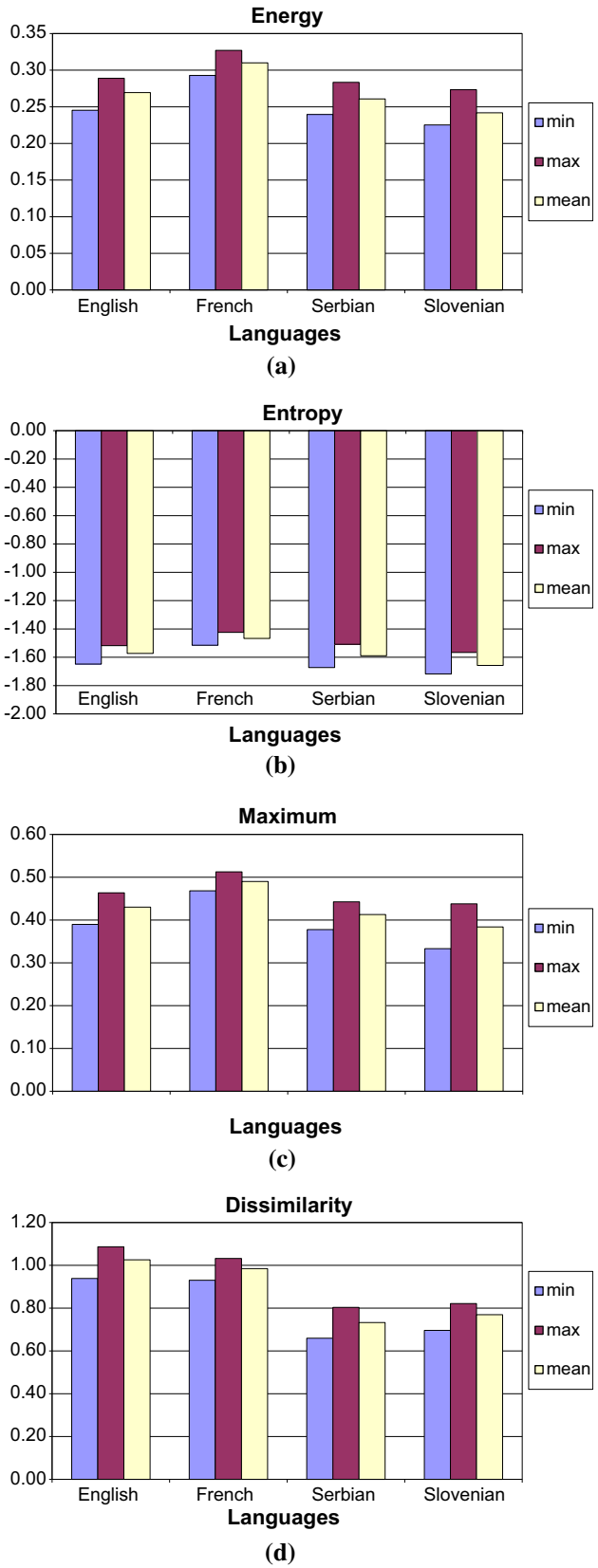


Fig. 7 Four out of eight GLCM features for English, French, Serbian and Slovenian documents obtained from the database: **a** energy, **b** entropy, **c** maximum, and **d** dissimilarity

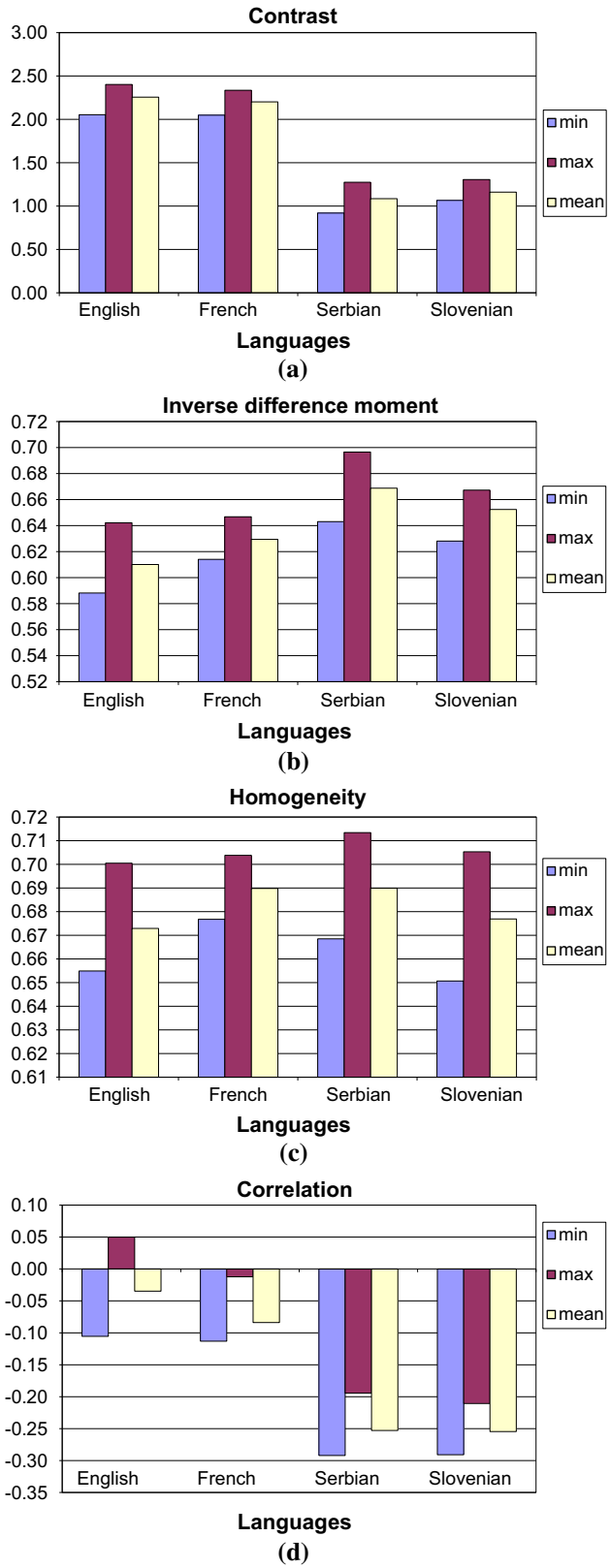


Fig. 8 Four out of eight GLCM features for English, French, Serbian and Slovenian documents obtained from the database: **a** contrast, **b** inverse difference moment, **c** homogeneity, and **d** correlation

Table 5 Maximum dispersion of four GLCM semi-features for English, French, Serbian and Slovenian documents from the database

Max dispersion	$\Delta\mu_x$	$\Delta\mu_y$	$\Delta\sigma_x$	$\Delta\sigma_y$	Avg. Δ
English	0.1152	0.1148	0.0546	0.0560	0.0852
French	0.0627	0.0619	0.0399	0.0397	0.0511
Serbian	0.0716	0.0718	0.0989	0.0989	0.0853
Slovenian	0.0601	0.0745	0.0631	0.0632	0.0652

Table 6 Maximum dispersion of eight GLCM features for English, French, Serbian and Slovenian documents from the database

Max dispersion	Δener.	Δentr.	Δmax.	Δdiss.	Δcontr.	Δinvdm.	Δhomog.	Δcorr.	Avg.Δ
English	0.0436	0.1295	0.0738	0.1482	0.3487	0.0540	0.0456	0.1553	0.1248
French	0.0342	0.0906	0.0443	0.1024	0.2871	0.0327	0.0270	0.1008	0.0899
Serbian	0.0440	0.1641	0.0648	0.1444	0.3555	0.0534	0.0449	0.0977	0.1211
Slovenian	0.0482	0.1515	0.1046	0.1253	0.2402	0.0392	0.0546	0.0801	0.1055

Table 7 Four GLCM semi-features for English, French, Serbian and Slovenian documents from the test database

GLCM semi-features	μ_x	μ_y	σ_x	σ_y
English				
Min	1.6927	1.6920	1.0231	1.0231
Max	1.7525	1.7517	1.0647	1.0647
Mean	1.7249	1.7244	1.0487	1.0486
French				
Min	1.6420	1.6403	1.0191	1.0185
Max	1.6825	1.6814	1.0347	1.0344
Mean	1.6618	1.6604	1.0249	1.0244
Serbian				
Min	1.5111	1.5106	0.7222	0.7222
Max	1.5391	1.5385	0.7695	0.7696
Mean	1.5277	1.5272	0.7450	0.7450
Slovenian				
Min	1.5335	1.5328	0.7506	0.7507
Max	1.5740	1.5733	0.7988	0.7988
Mean	1.5496	1.5491	0.7718	0.7718

matrix represents the number of documents of predicted class j appearing in the true language class i. Precision for a given ground language class i is the fraction of documents correctly classified as i (of language i) to the total number of retrieved documents in the predicted class j corresponding to i. Recall for a given ground language class i is the fraction of documents correctly classified as i (of language i) to the total number of relevant documents of class i. F-measure is the harmonic mean of precision and recall.

In the case of clustering algorithms, the correspondence between the found clusters and the true language classes is not known a priori. In order to identify which language corresponds to each cluster, we consider the confusion matrix **CM**. In the multi-class **CM**, each predicted class (cluster) j corresponds to the language class i sharing the maximum possible number of documents with it. Accordingly, the assignment of clusters to language classes is based on the Hungarian algorithm, which is a technique to be adoptable for optimal assignment of the clusters to the ground-truth classes in the **CM** [46].

The normalized mutual information (NMI) is an information-theoretic measure adopted for the clustering evaluation [47]. Specifically, let $O = \{o_1, o_2, \ldots, o_k\}$ and $D = \{d_1, d_2, \ldots, d_j\}$ be, respectively, the k clusters found from the algorithm and the j ground-truth language classes of documents. The NMI is defined as:

$$\mathrm{NMI}(O;D) = \frac{I(O;D)}{(H(O)+H(D))/2}, \tag{22}$$

where I is the mutual information and H is the entropy.

The mutual information $I(O; D)$ is expressed as:

$$I(O;D) = \sum_k \sum_j \frac{|o_k \cap d_j|}{N} \log \frac{N|o_k \cap d_j|}{|o_k||d_j|}, \tag{23}$$

where $|o_k \cap d_j|$ is the number of documents of language class d_j appearing in the cluster o_k, N is the total number of documents, $|o_k|$ and $|d_j|$ are, respectively, the number of documents inside the cluster o_k and inside the true language class d_j.

Entropy is computed as:

$$H(O) = -\sum_k \frac{|o_k|}{N} \log \frac{|o_k|}{N}. \tag{24}$$

$H(D)$ is computed analogously for D.

The mutual information I expresses the amount of information given from the found clusters useful to derive the membership of the documents in the true language classes. The minimum value of the mutual information I is obtained when the computed document clustering $\underline{O}$ is

Table 8 Eight GLCM features for English, French, Serbian and Slovenian documents obtained from the test database

GLCM features	Energy	Entropy	Maxi	Dissim.	Contr.	Invdm.	Homog.	Corr.
English								
Min	0.2600	−1.6141	0.4227	0.9655	2.0767	0.5998	0.6646	−0.0778
Max	0.2815	−1.5307	0.4539	1.0614	2.3807	0.6284	0.6884	0.0080
Mean	0.2708	−1.5784	0.4369	1.0315	2.2961	0.6107	0.6735	−0.0438
French								
Min	0.2834	−1.5315	0.4512	0.9756	2.1837	0.6046	0.6692	−0.1408
Max	0.3078	−1.4906	0.4903	1.0544	2.3766	0.6330	0.6923	−0.0519
Mean	0.2939	−1.5054	0.4668	1.0279	2.3111	0.6144	0.6773	−0.1004
Serbian								
Min	0.2012	−1.8758	0.2883	0.7702	1.2218	0.6532	0.6711	−0.1713
Max	0.2096	−1.8228	0.3381	0.7904	1.3321	0.6657	0.6848	−0.1186
Mean	0.2053	−1.8490	0.3145	0.7787	1.2722	0.6600	0.6791	−0.1465
Slovenian								
Min	0.1865	−1.9310	0.2905	0.7538	1.2513	0.6374	0.6624	−0.1743
Max	0.2018	−1.8536	0.3275	0.8540	1.4986	0.6728	0.6938	−0.0959
Mean	0.1969	−1.8827	0.3102	0.8021	1.3667	0.6554	0.6778	−0.1460

Table 9 Maximum dispersion of four GLCM semi-features for English, French, Serbian and Slovenian documents from the test database

Max dispersion	$\Delta\mu_x$	$\Delta\mu_y$	$\Delta\sigma_x$	$\Delta\sigma_y$	Avg. Δ
English	0.0598	0.0597	0.0416	0.0416	0.0507
French	0.0405	0.0411	0.0156	0.0159	0.0283
Serbian	0.0280	0.0279	0.0473	0.0474	0.0377
Slovenian	0.0405	0.0405	0.0482	0.0481	0.0443

random compared with the true partitioning D of documents in languages. The maximum value of I is realized when the computed document clustering $\underline{\underline{O}}$ is able to exactly reproduce the true language classes D. The normalized mutual information is a normalized version of I. and it is always bounded between 0 (minimum similarity between O and D) and 1 (maximum similarity between O and D).

In order to evaluate the superiority of the proposed framework for language classification, its results are compared with the results of other six unsupervised classifiers, well known in literature, adopted in some contexts for document classification and retrieval [48–52]. They are: k-means, complete linkage hierarchical clustering, DBSCAN [53], self-organizing map (SOM), Gaussian mixture models using expectation maximization (GMMs) and the clustering algorithm based on local search introduced in [54]. To demonstrate the significance and the utility of the modifications introduced in GA-ICDA, a comparison is also made with the GA-IC base approach. In terms of feature representation, a comparison is made between the proposed features and language n-gram model. In particular, the bi-gram frequency vectors are adopted as feature representation of the documents [55]. Hence, all the clustering algorithms are executed on the database of documents represented by the proposed feature vectors (see Table 11) and by the bi-gram frequency vectors (see Table 13). Then, we evaluate and analyze the best combination of feature representation and clustering approach.

For the competitor algorithms, a trial and error approach has been employed on the database for parameter tuning. The values of the parameters giving the best possible accuracy results on the database have been applied for clustering. This means that we perform a number of trials by considering different combinations of the parameter values and choose that combination determining the best possible solution [56]. Consequently, in k-means algorithm, the cluster number is fixed to 4, which is exactly the language number. In SOM algorithm, the dimension of a

Table 10 Maximum dispersion of eight GLCM features for English, French, Serbian and Slovenian documents from the test database

Max dispersion	Δener.	Δentr.	Δmaxi	Δdiss.	Δcontr.	Δinvdm.	Δhomog.	Δcorr.	Avg.Δ
English	0.0215	0.0834	0.0312	0.0959	0.3040	0.0286	0.0238	0.0858	0.0843
French	0.0244	0.0409	0.0391	0.0788	0.1929	0.0284	0.0231	0.0889	0.0646
Serbian	0.0084	0.0530	0.0498	0.0202	0.1103	0.0125	0.0137	0.0527	0.0401
Slovenian	0.0153	0.0774	0.0370	0.1002	0.2473	0.0354	0.0314	0.0784	0.0778

Table 11 Clustering results in terms of precision, recall, f-measure and NMI obtained from the following algorithms: GA-ICDA, GA-IC, k-means, complete linkage hierarchical clustering, DBSCAN, local search, SOM, GMMs, on the database of 85 documents represented by the proposed GLCM texture features

Classes	Precision	Recall	F-measure	NMI
GA-ICDA				
English	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
French	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
Serbian	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
Slovenian	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
GA-IC				
English	0.6571 (0.0000)	0.9200 (0.0000)	0.7667 (0.0000)	0.5303 (0.0000)
French	0.5556 (0.0000)	1.0000 (0.0000)	0.7143 (0.0000)	
Serbian	0.9375 (0.0000)	0.4688 (0.0000)	0.6250 (0.0000)	
Slovenian	0.6250 (0.0000)	0.5556 (0.0000)	0.5882 (0.0000)	
K-means				
English	0.7357 (0.1582)	0.9333 (0.1209)	0.8002 (0.0423)	0.5987 (0.0385)
French	0.4614 (0.1855)	1.0000 (0.0000)	0.6128 (0.1506)	
Serbian	0.7567 (0.1650)	0.5575 (0.1397)	0.6267 (0.1001)	
Slovenian	0.7079 (0.1664)	0.6232 (0.1501)	0.6405 (0.0906)	
Hierarchical				
English	1.0000 (0.0000)	0.5556 (0.0000)	0.7143 (0.0000)	0.5963 (0.0000)
French	0.5556 (0.0000)	1.0000 (0.0000)	0.7143 (0.0000)	
Serbian	0.7429 (0.0000)	0.8125 (0.0000)	0.7761 (0.0000)	
Slovenian	0.7273 (0.0000)	0.6400 (0.0000)	0.6809 (0.0000)	
DBSCAN				
English	0.6429 (0.0000)	1.0000 (0.0000)	0.7826 (0.0000)	0.6527 (0.0000)
French	0.3571 (0.0000)	1.0000 (0.0000)	0.5263 (0.0000)	
Serbian	0.5614 (0.0000)	1.0000 (0.0000)	0.7191 (0.0000)	
Slovenian	0.4386 (0.0000)	1.0000 (0.0000)	0.6098 (0.0000)	
Local search				
English	1.0000 (0.0000)	0.5556 (0.0000)	0.7143 (0.0000)	0.6262 (0.0000)
French	0.5556 (0.0000)	1.0000 (0.0000)	0.7143 (0.0000)	
Serbian	0.7568 (0.0000)	0.8750 (0.0000)	0.8116 (0.0000)	
Slovenian	0.8000 (0.0000)	0.6400 (0.0000)	0.7111 (0.0000)	
SOM				
English	0.6429 (0.0000)	1.0000 (0.0000)	0.7826 (0.0000)	0.6382 (0.0276)
French	0.3571 (0.0000)	1.0000 (0.0000)	0.5263 (0.0000)	
Serbian	0.6040 (0.0810)	0.9494 (0.0966)	0.7286 (0.0186)	
Slovenian	0.4957 (0.1088)	0.9296 (0.1339)	0.6272 (0.0332)	
GMMs				
English	0.6868 (0.1824)	0.8756 (0.1956)	0.7384 (0.1158)	0.5585 (0.0785)
French	0.4552 (0.1966)	0.9520 (0.1147)	0.5861 (0.1418)	
Serbian	0.7036 (0.1575)	0.7000 (0.2044)	0.6671 (0.0891)	
Slovenian	0.5738 (0.1745)	0.7104 (0.1987)	0.6016 (0.1010)	

Classes is the name of the specific language ground class in the database

neuron layer is 1×4, the number of training steps for initial covering of the input space is 100 and the initial size of the neighborhood is 3. The distance between two neurons is calculated as the number of steps from each other. Hierarchical clustering employs a bottom-up agglomerative strategy using L_1 norm for distance computation. Complete linkage is used for cluster distance evaluation, in order to be compliant with the bottom-up strategy of GA-ICDA. The obtained dendrogram is "horizontally" cut to give a cluster number which is equal to 4. In GMMs, the cluster number is also fixed to 4. In DBSCAN, the ϵ distance value is fixed to 20 and the minimum number of points for the ϵ-neighborhood is fixed to 1.5. In local search algorithm, the parameters are automatically tuned from the

algorithm, based on data [54]. According to the parameter setting described in Sect. 4, the h value of the neighborhood and the Γ threshold value of GA-ICDA have been learned from the database to be equal to 15. Also, the same h parameter value has been fixed for GA-IC. Again, we fix population size equal to 700, number of generations equal to 200, probability of mutation equal to 0.7 and probability of crossover equal to 1. Furthermore, we choose elite reproduction equal to 10 % of the population size and roulette selection function. The learned parameter values have been adopted for the test database.

Because some of the proposed clustering algorithms can obtain different solutions for different runs on the same database, they have been run 50 times on the database and on the test database and the average values of precision, recall, f-measure and NMI have been computed for each language class. Average NMI is an overall measure of similarity; consequently, it is reported separately from the single language classes. Standard deviation is reported in parenthesis.

Table 11 shows the results of the experiment when the documents of the database are represented by the proposed GLCM texture features. They are two (instead of four) semi-features (μ_x, σ_x) and eight full texture features (correlation, energy, entropy, maximum, dissimilarity, contrast, inverse difference moment and homogeneity). The semi-features μ_y and σ_y are not considered for classification in this case. In fact, from the previous feature analysis, it is visible that they are correlated to μ_x and σ_x. It is worth to note that GA-ICDA is able to obtain the perfect identification of the languages in the database, while the other methods perform poorly. This means that the algorithm is able to partition the documents, perfectly recognizing the language of the document text. The obtained standard deviation is 0: It demonstrates the stability of our result. In fact, each time the algorithm is run, the genetic procedure obtains a solution which is corrected by the bottom-up final procedure. In this way, standard deviation is 0 because, in all the 50 runs, the genetic procedure finds a sub-optimal solution but the bottom-up strategy performs successfully. Also, GA-IC base approach is not able to well discriminate among documents written in different languages, obtaining very poor results with respect to the modified GA-ICDA approach.

Table 12 shows the results of the experiment when the documents of the database are represented by the bi-gram frequency vectors. Although the local search, GA-IC and hierarchical clustering perfectly identify the English and French languages, they are not able to correctly differentiate the Serbian and Slovenian languages. Also, GA-ICDA is able to correctly identify the Slovenian language, while it obtains f-measure values of around 0.60–0.70 for the other languages. It is mainly because GA-ICDA method is customized for the proposed feature representation. Finally, the other clustering algorithms perform poorly. It is worth to observe that the best combination is the proposed feature representation together with the GA-ICDA clustering method. It is followed by the bi-gram frequency vectors representation together with the local search algorithm. However, the proposed framework is the only combination which is able to perfectly discriminate the 4 languages.

The adopted dataset is quite complex in terms of document typology. Consequently, we expect to perform successfully also in other contexts of classification for these languages. Accordingly, we run the proposed language identification tool on the test database given in the same languages English, French, Serbian and Slovenian. Results of the experiment are reported in Table 13. Also in this case, we obtain the perfect discrimination of the languages, which is a very promising result. Finally, the goodness of the classification process confirms the validity of the document feature representation, which is totally new in the literature.

At the end, a brief comparison between proposed and other LI techniques will be made. First, we shall compare our method with the so-called hybrid method connected to OCR process. One type of these techniques deals with discrimination between Indian and English script, i.e., languages. It is based on different typographical features of aforementioned scripts. The best result of 0.989 for recall, precision and f-measure is obtained in [14]. Similarly, the connected component profiles as a basic feature to discriminate Bangla and English scripts are extracted in [13]. This method correctly recognized around 95 % of script documents. Furthermore, an approach that discriminates between Latin and non-Latin documents is proposed in [15]. Its complex procedure receives an average accuracy of 95 %. However, it is suitable for scanned documents only. In contrast, a new technique introduces character code shapes which can be used not only to scan documents as previous methods, but also with some modification to electronic documents [16]. The method experiments on 23 Latin-based languages. It receives an overall accuracy between 80 and 93 %. Its weakness represents the NLP approach to classification based on 3-gram method. Secondly, comparison will be made to so-called NLP-based methods incorporating word based, n-gram approach and Markov model approach. A technique that uses 1-gram approach has been proved to be efficient [10]. It receives an accuracy between 93 and 100 %. Still, it is usable only if the language is known and used in training procedure. The exploration of LI on the example of small texts is given in [20]. It receives up to 99.44 % success in headlines and 81.61 % in dictionaries using 5-gram method and Naive-

Table 12 Clustering results in terms of precision, recall, f-measure and NMI obtained from the following algorithms: GA-ICDA, GA-IC, k-means, complete linkage hierarchical clustering, DBSCAN, local search, SOM, GMMs, on the database of 85 documents represented by the bi-gram frequency vectors

Classes	Precision	Recall	F-measure	NMI
GA-ICDA				
English	0.6429 (0.0000)	1.0000 (0.0000)	0.7826 (0.0000)	0.8223 (0.0000)
French	0.3571 (0.0000)	1.0000 (0.0000)	0.5263 (0.0000)	
Serbian	1.0000 (0.0000)	0.7200 (0.0000)	0.7200 (0.0000)	
Slovenian	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
GA-IC				
English	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	0.9232 (0.0269)
French	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
Serbian	1.0000 (0.0000)	0.7812 (0.0932)	0.8745 (0.0574)	
Slovenian	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
K-means				
English	0.8091 (0.1808)	0.9911 (0.0342)	0.8802 (0.1123)	0.7261 (0.0884)
French	0.6609 (0.3210)	0.9930 (0.0537)	0.7463 (0.2339)	
Serbian	0.6267 (0.2295)	0.7553 (0.0755)	0.6662 (0.1303)	
Slovenian	0.6217 (0.1933)	0.9996 (0.0040)	0.7513 (0.1305)	
Hierarchical				
English	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	0.7484 (0.0000)
French	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
Serbian	0.4898 (0.0000)	0.7500 (0.0000)	0.5926 (0.0000)	
Slovenian	0.5102 (0.0000)	1.0000 (0.0000)	0.6757 (0.0000)	
DBSCAN				
English	0.2118 (0.0000)	1.0000 (0.0000)	0.3495 (0.0000)	0.0001 (0.0000)
French	0.1176 (0.0000)	1.0000 (0.0000)	0.2105 (0.0000)	
Serbian	0.3765 (0.0000)	1.0000 (0.0000)	0.5470 (0.0000)	
Slovenian	0.2941 (0.0000)	1.0000 (0.0000)	0.4545 (0.0000)	
Local search				
English	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	0.9609 (0.0000)
French	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
Serbian	0.9697 (0.0000)	1.0000 (0.0000)	0.9846 (0.0000)	
Slovenian	1.0000 (0.0000)	0.9600 (0.0000)	0.9796 (0.0000)	
SOM				
English	0.9272 (0.1534)	0.9944 (0.0176)	0.9538 (0.0976)	0.7261 (0.0563)
French	0.8728 (0.2683)	1.0000 (0.0000)	0.9067 (0.1968)	
Serbian	0.5502 (0.1311)	0.8031 (0.0211)	0.6449 (0.0644)	
Slovenian	0.5164 (0.0793)	0.9920 (0.0253)	0.6751 (0.0515)	
GMMs				
English	0.5962 (0.2585)	0.8528 (0.1851)	0.6661 (0.2110)	0.5105 (0.1928)
French	0.4872 (0.2976)	0.9320 (0.1294)	0.5925 (0.2497)	
Serbian	0.6087 (0.2028)	0.7597 (0.1573)	0.6467 (0.1152)	
Slovenian	0.4858 (0.1607)	0.8484 (0.1810)	0.6007 (0.1374)	

Classes is the name of the specific language ground class in the database

Bayes classifier. Furthermore, a comparative study of different techniques for the LI is given in [17]. The success rate is between 79.2 and 99.2 % on a sample of 135 documents. Still, the combination of the methods could slightly improve the correctness of the results. Our method with a success rate of 100 % on 2 samples of, respectively, 85 (training set) and 23 (test set) documents has proven its advantage. Furthermore, it is computer time non-intensive and robust to errors in writing and typos as well. The efficiency of the language discrimination depends also from the execution time of the adopted GA-ICDA method. It takes 40 seconds on the database of 85 documents, on a desktop computer quad core at 2.6 GHz, 8 GB RAM and Windows 7 operating system. In the cases of mistyping

Table 13 Clustering results in terms of precision, recall, f-measure and NMI obtained from the proposed language identification method on the test database

Classes	Precision	Recall	F-measure	NMI
Spanish	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)
Slovenian	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
English	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
Serbian	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	
French	1.0000 (0.0000)	1.0000 (0.0000)	1.0000 (0.0000)	

Classes is the name of the specific language ground class in the test database

text, the other techniques are error prone. The proposed algorithm does not need even the correct OCR preprocessing and recognition, which is mandatory to any other LI technique. But, the most important advantage of our method represents its versatility, i.e., suitability to be used for scanned and electronic documents. Also, it can be extended for wider language corpora. From all aforementioned, our approach is promising especially in the cases of closely related languages or variations of the same language.

7 Conclusions

The manuscript proposed a novel approach for identification of languages in document samples according to texture analysis of the coded document established according to the text line structure definition. The analysis of four and eight full texture features shows a diversity in documents written in different languages. However, a strong classification tool such as GA-ICDA is mandatory for successful identification of a certain language. The proposed algorithm is tested on two custom-oriented databases, which include documents given in French, English, Slovenian and Serbian languages. The obtained results show full distinction between documents written in different languages. These results are very promising. Hence, the proposed algorithm is suitable for the language recognition on the Web or incorporation in OCR system and can be extended for wider language corpora.

Further research direction will be toward the application of the proposed algorithm to much wider document database incorporating closely related languages. Also, the combination of different texture techniques such as co-occurrence and run-length texture analysis will be included in the process of the language evaluation.

Again, comparison of our language discrimination tool with the competitor algorithms will be further enriched by automatizing the parameter selection process. In the case of k-means, an intelligent variant called ik-means [57] will automatically find the suitable number of clusters by detecting *anomalous patterns*. In the hierarchical clustering, the *L* method [58] will determine the optimal number of clusters by finding the "knee" point in a graph depicting the clustering evaluation function in dependence of the clusters number. In the case of DBSCAN, binary differential evolution [59] will automatically find the suitable *Eps* and *MinPts* parameter values. Again, the parameterless PLSOM [60] will be employed instead of the traditional SOM, where the selection of the learning and annealing rates and of the neighborhood size are completely eliminated. In the GMMs, employing the self-adaptive differential evolution together with the EM algorithm [61] will eliminate the problem of choosing initial mixture parameters.

Finally, an optimized version of the framework source code in C programming language will be officially provided for encouraging the community to evaluate the performances of our technique and to favor comparisons with other existing methods.

Acknowledgments This study was partially funded by the Grant of the Ministry of Education, Science and Technological Development of the Republic of Serbia, as a part of the Project TR33037 within the framework of Technological development program. The receiver of the funding is Dr. Darko Brodić.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Kranig S (2006) Evaluation of language identification methods. In: Proceedings of ACM SAC
2. Chowdhury GG (2003) Natural language processing. Annu Rev Inf Sci Technol 37(1):51–89
3. Lewandowski D (2008) Problems with the use of web search engines to find results in foreign languages. Online Inf Rev 32(5):668–672
4. Jin H, Wong KF (2002) A Chinese dictionary construction algorithm for information retrieval. ACM Trans Asian Lang Inf Process 1(4):281–296
5. Botha G, Zimu V, Barnard E (2006) Text-based language identification for the South African languages. In; Proceedings of the 17th annual symposium of the pattern recognition association of South Africa. Parys, South Africa, pp 7–13

6. Grothe L, De Luca EW, Nürnberger A (2008) A comparative study on language identification Methods. In: Proceedings of the sixth international conference on language resources and evaluation (LREC), 28–30 May. Marrakech, Morocco, pp 980–985
7. Jurafsky D, Martin JH (2009) Speech and language processing, 2nd edn. Pearson-Prentice Hall, Upper Saddle River
8. Roark B, Saraclar M, Collins M (2007) Discriminative n-gram language modeling. Comput Speech Lang 21(2):373–392
9. Goodman J (2006) A bit of progress in language modeling: extended version. Technical report MSR-TR-2001-72, Machine Learning and Applied Statistics Group, Microsoft Research, Redmond, WA
10. Takci H, Sogukpimar I (2005) Letter based text scoring method for language identification. In: Yakhno T (ed) Advances in information systems, lecture notes in computer science 3261. Springer, New York, pp 283–290
11. Barroso N, Lopez de Ipina K, Grana M, Ezeiza A (2011) Language identification for under-resourced languages in the basque context. In: Corchado E et al (eds) Advances in intelligent and soft computing, vol 87. Springer, New York, pp 475–483
12. http://www.doc.ic.ac.uk/∼nd/surprise_96/journal/vol4/gac1/report.html
13. Zhou L, Lu Y, Tan CL (2006) Bangla/English script identification based on analysis of connected component profiles. In: Bunke H, Spitz AL (eds) Document analysis systems VII, lecture notes in computer science 3872. Springer, New York, pp 243–254
14. SkMd Obaidullah, Mondal A, Das N, Roy K (2014) Script identification from printed indian document images and performance evaluation using different classifiers. Appl Comput Intell Soft Comput 896128:1–12
15. Lu S, Tan CL, Huang W (2006) Bangla/English script identification based on analysis of connected component profiles. In: Bunke H, Spitz AL (eds) Document analysis systems VII, lecture notes in computer science 3872. Springer, New York, pp 232–242
16. Sibun P, Spitz AL (1994) Language determination: natural language processing from scanned document images. In: 4th applied natural language processing conference (ANLP). pp 15–21
17. Grothe L, De Luca EW, Nürnberger A (2008) A comparative study on language identification methods. In: Proceedings of the sixth international language resources and evaluation (LREC). Marrakech, Marocco, pp 980–985
18. Kranig S (2011) Evaluation of language identification methods. B.S. Thesis, University of Tubingen International Studies in Computational Linguistics, Tubingen
19. Do HV (2010) Natural language identification for OCR applications. B.S. Thesis, Freie Universität Berlin, Department of Mathematics and Computer Science, Berlin
20. Gottron T, Lipka N (2010) A comparison of language identification approaches on short, query-style texts. In: Gurrin C et al (eds) Advances in information retrieval, lecture notes in computer science 5993. Springer, New York, pp 611–614
21. Fogel DB (1997) The advantages of evolutionary computation. In: Proceedings of biocomputing and emergent computation BCEC97. World Scientific Press, pp 1–11
22. Arnold DV, Beyer HG (2002) Local performance of the (1 + 1)-ES in a noisy environment. IEEE Trans Evolut Comput 6(1):30–41
23. Van Gorp J, Schoukens J, Pintelon R (2000) Learning neural networks with noisy inputs using the errors-in-variables approach. IEEE Trans Neural Netw. 11(2):14–402
24. Liu C, Lu C, Lee W (2000) Document categorisation by genetic algorithms. In: Proceedings IEEE international conference on systems, man, and cybernetics, 08–11 October. IEEE CS Press, Nashville, TN, 5:3868-3872
25. Jian-Xiang W, Huai L, Yue-hong S, Xin-Ning S (2009) Application of genetic algorithm in document clustering. In: Proceedings IEEE international conference on information technology and computer science ITCS, 25–26 July. IEEE CS Press, Kiev, pp 145–148
26. Akter R, Chung Y (2013) An evolutionary approach for document clustering. IERI Proc 4:370–375
27. Abdel-Kader RF (2010) Genetically improved PSO algorithm for efficient data clustering. In: Proceedings second international conference on machine learning and computing (ICMLC), 9–11 February. IEEE CS Press, Bangalore, pp 71–75
28. Ali AF (2014) A novel hybrid genetic differential evolution algorithm for constrained optimization problems. (IJACSA) Int J Adv Comput Sci Appl 3(6):7–12
29. Hoffstein J, Pipher J, Silverman JH (2008) An introduction to mathematical cryptography. Springer, New York
30. Paar C, Pelzl J (2009) Hash functions, chapter 11 of understanding cryptography. A textbook for students and practitioners, Springer, New York
31. Yaksic VOC (2003) A study on hash functions for cryptography, global information assurrance certification paper, SANS Institute
32. Zramdini AW, Ingold R (1998) Optical font recognition using typographical features. IEEE Trans Pattern Anal 20(8):877–882
33. Brodić D, Milivojević ZN, Maluckov ČA (2013) Recognition of the script in Serbian documents using frequency occurrence and co-occurrence analysis. Sci World J 896328:1–14
34. Haralick RM, Shanmugan K, Dinstein I (1973) Textural features for image classification. IEEE Trans Syst Man Cybern 3(6):610–621
35. Eleyan A, Demirel H (2011) Co-occurrence matrix and its statistical features as a new approach for face recognition. Turk J Electr Eng Comput sci 19(1):97–107
36. Clausi DA (2002) An analysis of co-occurrence texture statistics as a function of grey level quantization. Can J Remote Sens 28(1):45–62
37. Conners RW, Trivedi MM, Harlow CA (1984) Segmentation of a high-resolution urban scene using texture operators. Comput Vis Gr Image Process 25:273–310
38. Newsam SD, Kamath C (2004) Retrieval using texture features in high resolution multi-spectral satellite imagery. In: SPIE conference on data mining and knowledge discovery: theory, tools, and technology VI
39. Amelio A, Pizzuti C (2014) A New evolutionary-based clustering framework for image databases. In: Elmoataz A et al (eds) Image and signal processing, lecture notes in computer science 8509. Springer, New York, pp 322–331
40. Marti R, Campos V, Laguna M, Glover F (2001) Reducing the bandwidth of a sparse matrix with tabu search. Eur J Oper Res 135(2):450–459
41. http://www.lepoint.fr
42. Comrey AL, Lee HB (1992) A first course in factor analysis. Psychology Press, Hillsdale
43. Cattell RB (1978) The scientific use of factor analysis in behavioral and life sciences. Plenum, New York
44. MacCallum RC, Widaman KF, Zhang S, Hong S (1999) Sample size in factor analysis. Psychol Methods 4(1):84–99
45. Brodić D, Milivojević ZN, Maluckov ČA (2015) An approach to the script discrimination in the Slavic documents. Soft Comput 19(9):2655–2665
46. Shrestha P, Jacquin C, Daille B (2012) Clustering short text and its evaluation. In: Proceedings of the 13th international conference, CICLing, March 11–17. Springer, New Delhi, India, LNCS 7182, pp 169–180
47. Manning CD, Raghavan P, Schütze H (2009) Introduction to information retrieval, Online edn. Cambridge University Press, Cambridge
48. Diem M, Kleber F, Fiel S, Sablatnig R (2013) Semi-automated document image clustering and retrieval. In: Proceedings SPIE, 9021, 0210M-90210M-10

49. Yuyu Y, Xu W, Yueming L (2013) A Hierarchical Method for Clustering Binary Text Image. In: Yuyu Y et al (eds) Trustworthy computing and services, CCIS 320. Springer, New York, pp 388–396
50. Weizhong Z, Qing H, Huifang M, Zhongzhi S (2012) Effective semi-supervised document clustering via active learning with instance-level constraints. Knowl Inf Syst 30(3):569–587
51. Marinai S, Marino E, Soda G (2008) Self-organizing maps for clustering in document image analysis. In: Simone M, Hiromichi F (eds) Machine learning in document analysis and recognition, studies in computational intelligence 90. Springer, New York, pp 193–219
52. Huaigu C (2008) Indexing and retrieval of low quality handwritten documents. Ph.D. Dissertation. State University of New York at Buffalo, Buffalo
53. Ester M, Kriegel H, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the second international conference on knowledge discovery and data mining (KDD-96). AAAI Press, pp 226–231
54. Zelnik-Manor L, Perona P (2004) Self-tuning spectral clustering. Adv Neur Inf 17:1601–1608
55. Turney PD, Pantel P (2010) From frequency to meaning: vector space models of semantics. J Artif Intell Res 37(1):141–188
56. Fodor JD, Sakas WG (2004) Evaluating models of parameter setting. in: Proceedings of the 28th annual Boston University conference on language development, October 31–November 2, Boston, MA
57. Chiang MM, Mirkin B (2010) Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. J Classif 27(1):3–40
58. Salvador S, Chan P (2004) Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: Proceedings of the 16th IEEE international conference on tools with artificial intelligence, ICTAI. pp 576–584
59. Karami A, Johansson R (2014) Choosing DBSCAN parameters automatically using differential evolution. Int J Comput Appl 91(7):1–11
60. Berglund E, Sitte J (2006) The parameterless self-organizing map algorithm. IEEE Trans Neural Netw 17(2):305–316
61. Kwedlo W (2014) Estimation of parameters of Gaussian mixture models by a hybrid method combining a self-adaptive differential evolution with the EM algorithm. Adv Comput Sci Res 11:109–123