EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

A novel approach for handedness detection from off-line handwriting using fuzzy conceptual reduction

Somaya Al-Maadeed*, Fethi Ferjani, Samir Elloumi and Ali Jaoua

Abstract

A challenging area of pattern recognition is the recognition of handwritten texts in different languages and the reduction of a volume of data to the greatest extent while preserving associations (or dependencies) between objects of the original data. Until now, only a few studies have been carried out in the area of dimensionality reduction for handedness detection from off-line handwriting textual data. Nevertheless, further investigating new techniques to reduce the large amount of processed data in this field is worthwhile. In this paper, we demonstrate that it is important to select only the most characterizing features from handwritings and reject all those that do not contribute effectively to the process of handwriting recognition. To achieve this goal, the proposed approach is based mainly on fuzzy conceptual reduction by applying the Lukasiewicz implication. Handwritten texts in both Arabic and English languages are considered in this study. To evaluate the effectiveness of our proposal approach, classification is carried out using a K-Nearest-Neighbors (K-NN) classifier using a database of 121 writers. We consider left/right handedness as parameters for the evaluation where we determine the recall/precision and F-measure of each writer. Then, we apply dimensionality reduction based on fuzzy conceptual reduction by using the Lukasiewicz implication. Our novel feature reduction method achieves a maximum reduction rate of 83.43 %, thus making the testing phase much faster. The proposed fuzzy conceptual reduction algorithm is able to reduce the feature vector dimension by 31.3 % compared to the original ⊠BEST OF ALL COMBINED FEATURES⊠algorithm.

Keywords: Index terms-handwriting, Fuzzy binary relation, Left/right identification, Feature, Lukasiewicz implication, Galois connections, Closure of Fuzzy Galois connections

1 Introduction

Handwriting recognition is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens, and other devices. The image of the written text may be sensed “off-line” from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition. Alternatively, the movements of the pen tip may be sensed “on-line”, for example by a pen-based computer screen surface. Handwriting recognition principally entails optical character recognition. However, a complete handwriting recognition system also handles formatting, performs correct segmentation into characters and determines the most plausible words.

Handwriting recognition has been one of the fascinating and challenging research areas in the field of image processing and pattern recognition [1, 2]. It contributes immensely to the advancement of an automation process and can improve the interface between human beings and machines in numerous applications. Several research works have been focusing on new techniques and methods that would reduce the processing time while providing higher recognition accuracy [3].

In general, handwriting recognition is classified into two types of methods: off-line and on-line handwriting recognition methods. In off-line recognition, the writing is usually captured optically by a scanner, and the complete text is available as an image. In the on-line system, the two-dimensional coordinates of successive points are

*Correspondence: s_alali@qu.edu.qa
Department of Computer Science and Engineering, Qatar University, Doha, Qatar

represented as a function of time, and the order of strokes made by the writer is also available. The on-line methods have been shown to be superior to their off-line counterparts in recognizing handwritten characters due to the temporal information available with the former [4, 5]. Several applications, including mail sorting, bank processing, document reading, and postal address recognition, require off-line handwriting recognition systems. As a result, off-line handwriting recognition continues to be an active area of research toward exploring newer techniques that would improve recognition accuracy [6].

In our current study, we focus on off-line handwriting and use a K-Nearest Neighbors (K-NN) classifier to make classifications based on many parameters, such as gender, age, handedness, and nationality, to measure the performance of our proposed algorithm. This type of classification has several applications. For example, in the forensic domain, handwriting classification can help investigators to focus on a certain category of suspects. Additionally, processing each category separately leads to improved results in writer identification and verification applications.

There are few studies in the literature that investigate the automatic detection of gender, age, and handedness from handwritings. Bandi et al. [7] proposed a system that classifies handwritings into demographic categories using the "macro-features" introduced in [8]. These features focus on measurements such as pen pressure, writing movement, stroke formation, and word proportion. The authors reported classification accuracies of 77.5, 86.6, and 74.4 % for gender, age and handedness classification, respectively. However, in this study, all the writers produced the same letter. Unfortunately, this is not always the case in real forensic caseworks. Moreover, the dataset used in this study is not publicly available.

Liwicki et al. [9] also addressed the classification of gender and handedness in the on-line mode (which means that the temporal information about the handwriting is available). The authors used a set of 29 features extracted from both on-line information and its off-line representation and applied support vector machines and Gaussian mixture models to perform the classification. The authors reported a performance of 67.06 % for gender classification and 84.66 % for handedness classification. In [10], the authors separately reported the performance of the off-line mode, the on-line mode, and their combination. The performance reported for the off-line mode was 55.39 %, which is slightly better than chance.

In this paper, we propose a novel approach to the detection of the handedness of the writer of a handwritten document based on the Lukasiewicz implication where a set of features was proposed and evaluated to predict the handedness of the writer. These features are combined using a K-Nearest Neighbors (K-NN) classifier under the *Rapidminer* platform [11]. This method is evaluated using the QUWI database, which is the only publicly available dataset containing annotations regarding gender, age range, and nationality.

The rest of the paper is organized as follows. In section 2, we review some basic definitions from relational algebra, the mathematical background related to fuzzy set theories and useful for this research paper. In section 3, the state-of-the-art in writer identification for the English and Arabic languages is presented in detail. The evaluation was made using larger amounts of text and may not produce acceptable results when limited amounts of text are available. Writer recognition from short handwritten texts is therefore an interesting area of study. Section 4 gives a description of the system overview and the database used for carrying out the experimental evaluations. Next, we describe the proposed features and the method in which they are extracted. Then, we present the utilized (K-NN) classifier followed by the detailed results and an analysis of the experimental evaluations. Finally, we conclude the paper with some discussion on future research directions on the subject.

2 Key settings and new definitions

The domains of computer science, relational algebra, formal concept analysis, and lattice theory have seen important advances in research [12]. This research has contributed enormously to the search for original solutions for complex problems in the domains of knowledge engineering, data mining, and information retrieval. Relational algebra and formal concept analysis may be considered as useful mathematical foundations that unify data and knowledge in information retrieval systems.

2.1 Binary relations

In the following, we review some basic definitions from relational algebra. Let us consider two sets $\mathcal{X}$ and $\mathcal{Y}$ and two elements e and e', where $e \in \mathcal{X}$ and $e' \in \mathcal{Y}$.

- A relation $\mathcal{R}$ is a subset of the Cartesian product of two sets $\mathcal{X}$ and $\mathcal{Y}$.
- An element $(e, e') \in \mathcal{R}$, where e' denotes the image of e by $\mathcal{R}$.
- A binary relation identity $\mathcal{I}(\mathcal{A}) = \{(e, e)|e \in \mathcal{A}\}$.
- The relative product or composition of two binary relations $\mathcal{R}$ and $\mathcal{R}^{'}$ is $\mathcal{R} \circ \mathcal{R}^{'} = \{(e, e')|\exists t \in \mathcal{Y} : ((e, t) \in \mathcal{R})\&((t, e') \in \mathcal{R}')\}$.
- The inverse of the relation $\mathcal{R}$ is $\mathcal{R}^{-1} = \{(e, e')|(e', e) \in \mathcal{R}\}$.
- The set of images of e is defined by $e.\mathcal{R} = \{e'|(e, e') \in \mathcal{R}\}$.
- The set of antecedents of e' is defined by $\mathcal{R}.e' = \{e|(e, e') \in \mathcal{R}\}$.

- The cardinality of $\mathcal{R}$ is defined by $Card(\mathcal{R})$ = the numbers of pairs $(e, e') \in \mathcal{R}$.
- The complement of the relation $\mathcal{R}$ is $\overline{\mathcal{R}} = \{(e, e') | (e, e') \notin \mathcal{R}\}$.
- The domain of $\mathcal{R}$ is defined by $Dom(\mathcal{R}) = \{e | \exists e' : (e, e') \in \mathcal{R}\}$.
- The range or codomain of $\mathcal{R}$ is defined by $Cod(\mathcal{R}) = \{e' | \exists e : (e, e') \in \mathcal{R}\}$.

2.2 Formal concept analysis

Formal Concept Analysis (FCA) is the mathematical theory of data analysis using formal contexts and concept lattices [12–14]. It was introduced by Rudolf Wille in 1984 and builds on applied lattice and order theory, which were developed by Birkhoff et al. [15]

Definition 1. A formal context
A formal context (or an extraction context) is a triplet $\mathcal{K} = (\mathcal{X}, \mathcal{Y}, \mathcal{R})$, *where* $\mathcal{X}$ *represents a finite set of objects,* $\mathcal{Y}$ *is a finite set of attributes (or properties), and* $\mathcal{R}$ *is a binary (incidence) relation, (i.e.,* $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$*). Each couple* $(x, y) \in \mathcal{R}$ *expresses that the object* $x \in \mathcal{X}$ *verifies property* y *belonging to* $\mathcal{Y}$.

Definition 2. Formal concept in fuzzy binary relation
Let $\mathcal{X}$ *be a set called the universe of discourse. Elements of* $\mathcal{X}$ *are denoted by lowercase letters. A fuzzy set* $E = \{x_1/v_1, x_2/v_2, \cdots, x_n/v_n\}$ *is defined as a collection of elements* $x_i \in \mathcal{X}, i = 1 : n$, *which includes a degree of membership* v_i *for each element* x_i [16, 17].

A fuzzy binary context (or fuzzy binary relation) is a fuzzy set defined on the product of two sets $\mathcal{O}$ *(set of objects) and* $\mathcal{P}$ *(set of properties). Hence,* $\mathcal{X} = \mathcal{O} \times \mathcal{P}$.

Definition 3. Galois connection
Let $(\mathcal{X}, \mathcal{Y}, \mathcal{R})$ *be the formal context, and let* $A \subseteq \mathcal{X}$ *and* $B \subseteq \mathcal{Y}$ *be two finite sets. We define two operators* $f(A)$ *and* $g(B)$ *on A and B as follows:*

1. $f(A) = \left\{e' | \forall e \in A, (e, e') \in \mathcal{R}\right\}$,
2. $g(B) = \left\{e | \forall e' \in B, (e, e') \in \mathcal{R}\right\}$.

Operator f defines the properties shared by all elements of A, and operator g defines objects sharing the same properties included in set B. The operators f and g define a Galois connection between the sets $\mathcal{X}$ *and* $\mathcal{Y}$ *with respect to the binary context* $(\mathcal{X}, \mathcal{Y}, \mathcal{R})$ [12, 18].

Definition 4. Fuzzy set
In classical set theory, elements fully belong to a set or are fully excluded. However, a fuzzy set $\mathcal{A}$ *in universe* $\mathcal{X}$ *is the set whose elements also partially belong to* $\mathcal{X}$. *The grade of belonging of each element is determined by a membership function* μ_A *given by* [16]

- $\mu_A : \mathcal{X} \rightarrow [0, 1]$,
- *A finite fuzzy set can be denoted as* $A = \{\mu_A(x_1)/x_1, \mu_A(x_2)/x_2, \ldots, \mu_A(x_n)/x_n\}$, *for any* $x_i \in \mathcal{X}$.

Example 1. *Let us consider the fuzzy relation* $\mathcal{F}_{BR}$ *depicted in Table 1.* $\mathcal{F}_{BR}$ *contains five objects* O_1, O_2, O_3, O_4, *and* O_5 *and six properties {**a**, **b**, **c**, **d**, **e**, **f**}, where the values have been set randomly.*

Definition 5. *Fuzzy Galois connection*
Let $\mathcal{F}_{BR}$ *be a fuzzy binary relation defined on* $\mathcal{X}$. *For two sets A and B such that* $A \subseteq \mathcal{O}$, *B is a fuzzy set defined on* $\mathcal{P}$, *and* $\delta \in [0, 1]$ [17, 19–21]. *We define the operators* $\mathcal{F}$ *and* $\mathcal{H}_\delta$ *as follows:*

- $\mathcal{F}(A) = \left\{d/\alpha | \alpha = min\{\mu_{\mathcal{F}_{BR}}(g, d) | g \in A, d \in \mathcal{P}\}\right\}$
- $\mathcal{H}_\delta(B) = \left\{g | d \in \mathcal{P} \Rightarrow (\mu_{\mathcal{B}}(d) \rightarrow_L \mu_{\mathcal{F}_{BR}}(g, d) \geqslant \delta\right\}$,

where $\rightarrow_L$ *denotes the Lukasiewicz implication. For example, for* $x_i, x_j \in [0, 1]$,

$$x_i \rightarrow_L x_j = min(1, 1 - x_i + x_j). \tag{1}$$

Note that $\mu_{\mathcal{F}_{BR}}(g, d)$ *denotes the weight of the pair (g,d) in the fuzzy relation* $\mathcal{F}_{BR}$.

Definition 6. A fuzzy closure operator
For two sets A and B such that $A \subseteq \mathcal{O}$, *B is a fuzzy set defined on* $\mathcal{P}$, *and* $\delta \in [0, 1]$. *We define* $Closure(A) = \mathcal{H}_\delta(\mathcal{F}(A)) = A'$ *and* $Closure(B) = \mathcal{F}(\mathcal{H}_\delta(B)) = B'$.

The composition $f \circ g$ defines the closure of the Galois connection. Let A_i , A_j be subsets of objects $\mathcal{O}$, and B_i , B_j fuzzy subsets defined on $\mathcal{P}$. The operators f and g have the following properties [12]:

1. $A_i \subseteq A_j \Rightarrow f(A_i) \supseteq f(A_j)$;
2. $B_i \subseteq B_j \Rightarrow g(B_i) \supseteq g(B_j)$;
3. $A_i \subseteq g \circ f(A_i)$ and $B_i \subseteq f \circ g(B_i)$;
4. $A \subseteq g(B) \Leftrightarrow B = f(A)$; and
5. $f = f \circ g \circ f$ and $g = g \circ f \circ g$.

Fuzzy data reduction To manage the large amount of features, it is important to select the most pertinent ones.

Table 1 Fuzzy binary relation

	a	b	c	d	e	f
O_1	0.5	0.2	0.6	0.4	0.7	0.5
O_2	0.7	0.3	0.2	0.3	0.2	1
O_3	1	0.4	0.6	1	0.7	0.5
O_4	0.5	0.2	0.6	0.4	0.8	0.6
O_5	0.7	0.3	0.7	0.4	0.6	0.7

In this paper, we use fuzzy conceptual reduction applied to the original data. Fuzzy conceptual data reduction methods have the main objective of minimizing the size of data while preserving the content of the original document. Unfortunately, most of the methods presented in the literature are based on heuristics and are not accurate. Moreover, reducing fuzzy data becomes a difficult problem because the handling of imprecision and uncertainty may cause information loss and/or deformation. In this work, we develop a fuzzy conceptual approach based on Lukasiewicz fuzzy Galois. This method is based on fuzzy formal concept analysis, which has been recently developed by several researches and applied for learning, knowledge acquisition, information retrieval, etc. The Lukasiewicz implication based on the fuzzy Galois connection is mainly used in this paper. It allows one to consider different precision levels according to the value of δ in the definition of fuzzy formal concepts.

The advantage of reduced data is that it can be used directly as a prototype for making decisions, for supervised learning, or for reasoning. For that purpose, we first prove that some rows can be removed from the initial fuzzy binary context at a given precision level (value given to δ by application of the Lukasiewicz implication). It is primordial to assess that there is an equivalence between an object and a set of objects. Second, we define a solution for data reduction in the case of fuzzy binary relations.

Equivalence between an object and a subset of other objects An object x is equivalent (for a given value of δ for δ varying from 0 to 1) to a set of objects S_x, relative to a fuzzy binary context $\mathcal{F}_{BR}$, if and only if $\{x\} \cup S_x$ is a domain of a concept of $\mathcal{F}_{BR}$, and the closure $(S_x) = \{x\} \cup S_x$, where $x \notin S_x$. As intuitive justification, x is equivalent to S_x means that $S_x \rightarrow x$ within some precision δ.

3 A review of related works

Handwriting refers to the style of writing textual documents with a writing instrument such as a pen or pencil by a person. Characteristics of handwriting include the following: (1) specific shapes of letters, e.g., their roundness or sharpness; (2) regular or irregular spacing between letters; (3) the slope of the letters; (4) the rhythmic repetition of the elements or arrhythmia; (5) the pressure to the paper; and (6) the average size of letters. Because each person's handwriting is unique, it can be used to verify a document's writer. Therefore, writer identification has been recently studied in a wide variety of applications, such as security, financial activity, and forensics and has been used for access control. Writer identification is the task of determining the writer of a document among different writers. Writer identification methods can be categorized into two types: text-dependent methods and text-independent methods. In text-dependent methods, a writer has to write the same fixed text to perform identification, but in text-independent methods, any text may be used to establish the identity of the writer. These methods can be performed on-line, where dynamic information about the writing is available, or off-line, where only a scanned image of the writing is available. Recently, different approaches for writer identification have been proposed. A scientific validation of the individuality of handwriting was performed in [22]. In that study, handwriting samples from 1500 individuals, representative of the US population with respect to gender, age, ethnic groups, etc., were obtained. The writer can be identified based on macro-features and micro-features that were extracted from handwritten documents. The authors in [23] proposed a global approach based on multi-channel Gabor filtering, where each writer's handwriting is regarded as a different texture. Bensefia et al. [24] used local features based on graphemes extracted from segmentations of cursive handwriting. In addition, writer identification has been performed by a textual-based information retrieval model. The work in [25] presented a new approach using a connected-component contour codebook and its probability density function. In addition, combining connected-component contours with an independent edge-based orientation and curvature PDF yields very high correct identification rates. Schlapbach and Bunke [26] proposed an HMM-based approach for writer identification and verification. In [27], the authors used a combination of directional, grapheme, and run-length features to improve writer identification and verification performance. Other studies used chain code and global features for writer identification [28]. Both proposed methods are applicable to cursive handwriting and have practical feasibility for writer identification. Other applications including such as off-line handwriting recognition systems for different languages reached up to 99 % for handwritten characters [29]. For handedness detection our earlier study proved that it can work [30]. Authors in [25] evaluated the performance of edge-based directional probability distributions as features in comparison to a number of non-angular features. [31] extracted a set of features from handwritten lines of text. The features extracted correspond to visible characteristics of the extracted feature score writing, such as the width, slant, and height of the three main writing zones. In [32], a new feature vector was employed by means of morphologically processing the horizontal profiles of the words. Because of the lack of a standard database for writer identification, a comparison of the previous studies is not possible. Because our purpose is to introduce an automatic method and because no limitation on handwriting is considered, methods that need no segmentation or connected-component analysis are regarded. Most previous studies are based on English documents with the assumption that the written text is fixed (text-dependent methods), and no research has been reported on English and Arabic texts or Arabic documents. In this paper, we propose a new method that is text-independent for off-line writer identification based on English/Arabic handwriting. Based on the idea that was presented in [23], we assume handwriting as a texture image, and a set of new features are extracted from preprocessed images of documents.

4 System overview of proposed generic approach for combined features

In this section, we present a system overview of our proposed approach. We then describe the dataset that was utilized to obtain the results. In the following, we give a description of the feature extraction and subsequently detail our proposed algorithm. The main experiment is discussed in this paper.

4.1 System overview

Figure 1 shows the handedness recognition system where the features are extracted from training and testing documents. The (K-NN) classifier [11] is then used to predict the handedness of the writer.

4.2 Description of the dataset

The dataset contains samples from 121 writers, which half of them are left-handed and the other half are right-handed. The dataset is a subset of the QUWI dataset, which was described in [33] in which 475 writers produced four handwritten documents: the first page contains an Arabic handwritten text that varies from one writer to another, the second page contains an Arabic handwritten text that is the same for all the writers, the third page contains an English handwritten text that varies from one writer to another, and the fourth page contains an English handwritten text that is the same for all the writers. left-handed writers were less represented in QUWI dataset. Only 121 writers were selected from this dataset to have evenly sampled data over the right- and left-handed writers. Figures 2 and 3 illustrate an example of the two pages. The images have been acquired using an EPSON GT-S80 scanner with a 600-DPI resolution. The images were provided in a JPG uncompressed format. The whole dataset of images and the corresponding features can be downloaded from the web site[1].

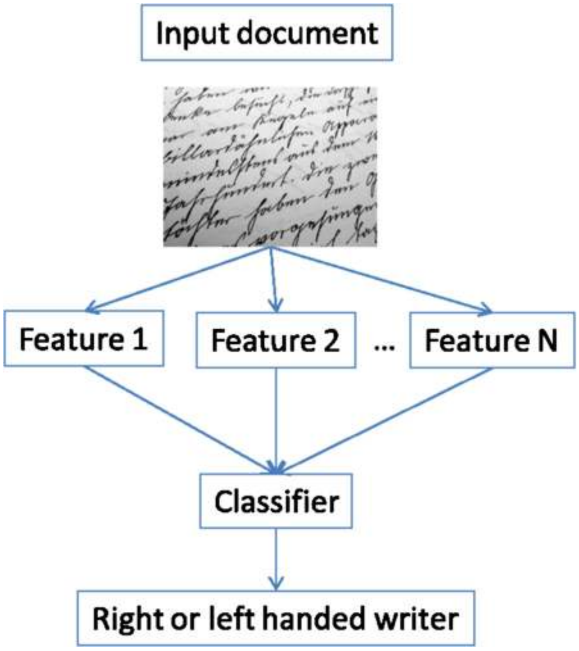


Fig. 1 System overview of the proposed handedness detection approach

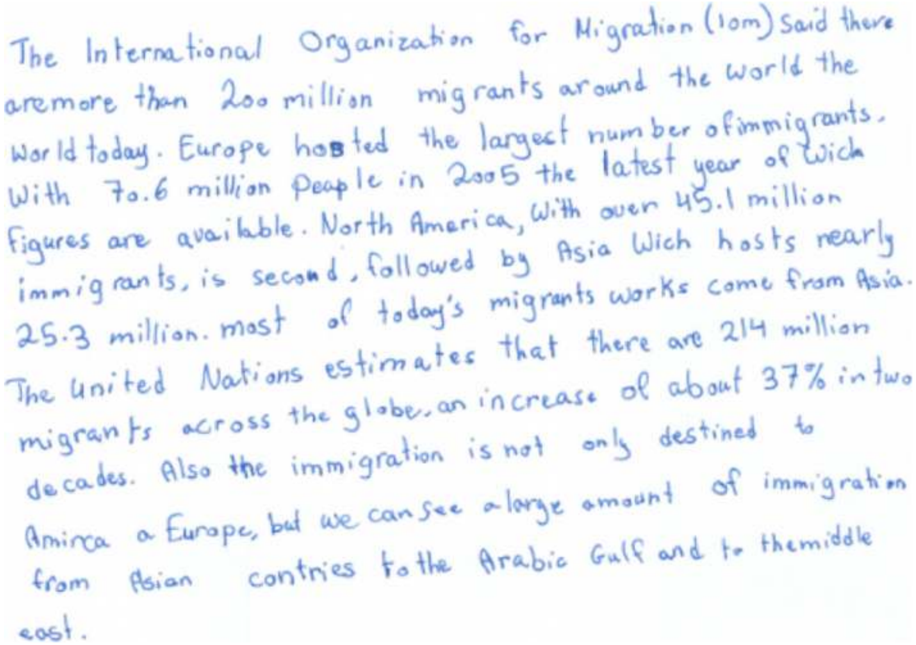
4.3 Feature extraction

In this step, the characterizing features are extracted from the handwriting data. As a preprocessing step for feature f1, we calculate the Zhang skeleton of the binarized image. This algorithm is popular for not creating parasitic branches compared to most skeletonization algorithms. For some other features, the contours were calculated using Freeman chain codes. We then continue the extraction of features, which measure the direction of writing (f1), curvature (f2), tortuosity (f3), chain code feature (f4, 6, and 16), and edge-based directional features (f16), in order to compare the results. In addition, edge-based directional features using the whole window computed for size 7 (f18, whose PDF size is 112) and size 10 (f26, whose PDF size is 220) are also extracted. These features enable us to discriminate between left- and right-handed writers as will be explained in the following sections. Figures 2 and 3 show handwriting examples of left- and right-handed writers. Figure 4 shows some examples of feature extraction from preprocessed handwritten character.

To make the system independent of the pen, images are first binarized using the Otsu thresholding algorithm [34]. The following subsections describe the features considered in this study. It is to be noted that these features do not correspond to a single value but are defined by a probability distribution function (PDF) extracted from the handwriting images to characterize the writer's individuality [35, 36]. The PDF describes the relative likelihood for a certain feature to take on a given value.

4.3.1 Directions (f1)

We move along the pixels of the obtained segments of the skeleton using a predefined order favoring the four connectivity neighbors. For each pixel *p*, we consider the



The International Organization for Migration (IOM) Said there
aremore than 200 million migrants around the World the
World today. Europe hosted the largest number ofimmigrants.
With 70.6 million People in 2005 the latest year of Wich
Figures are available. North America, With over 45.1 million
immigrants, is second, followed by Asia Wich hosts nearly
25.3 million. most of today's migrants works come from Asia.
The United Nations estimates that there are 214 million
migrants across the globe. an increase of about 37% in two
decades. Also the immigration is not only destined to
Amirca a Europe, but we can See a large amount of immigration
from Asian contries to the Arabic Gulf and to themiddle
east.

Fig. 2 Typical example of right-handed writer

2 × $N + 1$ neighboring pixels centered at p. The linear regression of these pixels is calculated to give the tangent at the pixel p [36].

The PDF of the resulting directions is computed as a probability vector for which the size has been empirically set to 10.

4.3.2 Curvatures (f2)

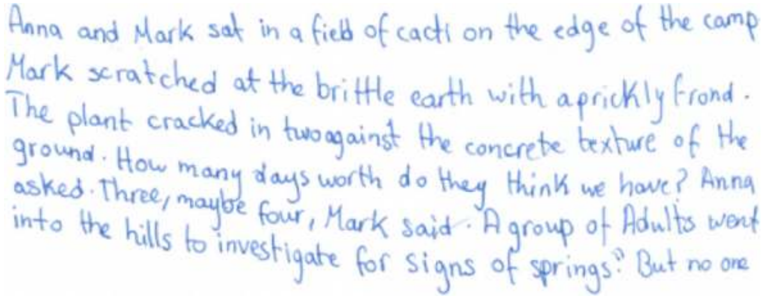
For each pixel p belonging to the contour, we consider a neighboring window, which has a size t. We compute the number of pixels n_1 inside this neighboring window belonging to the background and the number of pixels n_2 representing the foreground. Obviously, the difference $n_1 - n_2$ increases with the local curvature of the contour. We then estimate the curvature as being $C = \frac{n_1 - n_2}{n_1 + n_2}$. The PDF of the curvatures is computed as a vector whose size has been empirically set to 100 (s pixels in each side) [36].

4.3.3 Tortuosity (f3)

This feature makes it possible to distinguish between fast writers who produce smooth handwriting and slow writers who produce "ortuous"/twisted handwriting [36] by finding the longest line in the middle of the character shape. This feature has a PDF vector of 10. The PDF of the angles of the longest traversing segments are produced in a vector whose size has been set to 10.

4.3.4 Chain code features (f4-f7)

Chain codes are generated by browsing the contour of the text and assigning a number to each pixel according to its location with respect to the previous pixel. A chain code might be applied in different orders:



Anna and Mark sat in a field of cacti on the edge of the camp
Mark scratched at the brittle earth with aprickly frond.
The plant cracked in twoagainst the concrete texture of the
ground. How many days worth do they think we have? Anna
asked. Three, maybe four, Mark said. A group of Adults went
into the hills to investigate for signs of springs? But no one

Fig. 3 Typical example of left-handed writer

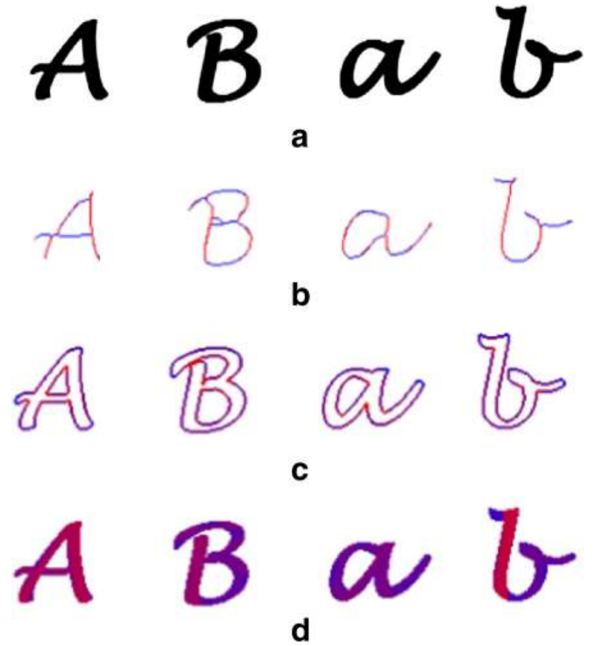


Fig. 4 Examples of some of feature extractions. **a** Original binary handwritten characters. **b** Direction feature f1 where the *red* color corresponds to a $\pi/2$ tangent, and the *blue* color corresponds to a zero tangent. **c** Curvature feature f2 highlighted in the binary image; *red* corresponds to the maximum curvature, and *blue* corresponds to the minimum curvature. **d** Tortuosity feature f3 where *red* corresponds to the maximum line segment that traverses a point and *blue* to the minimum length

f4: PDF of i patterns in the chain code list such that $i \in 0, 1, \ldots, 7$. This PDF has a size of 8.

f5: PDF of (i, j) patterns in the chain code list such that $i, j \in 0, 1, \ldots, 7$. This PDF has a size of 64. Similarly, **f6** and **f7** correspond to a PDF of (i, j, k) and (i, j, k, l) in the chain code list, where $i, j, k, l \in 0, 1, \ldots, 7$. Their respective sizes are 512 and 4096.

4.3.5 *Edge-based directional features (f8-f26)*

In this paper, this feature has been computed from size 1 (**f8**, whose PDF size is 4) to size 10 (**f17**, whose PDF size is 40). We have also extended these features to include the whole window. This feature has been computed from size 2 (**f18**, whose PDF size is 12) to size 10 (**f26**, whose PDF size is 220) (ref. Table 2).

Each contour $Contour_i$, being a sequence of consecutive boundary points, is computed as follows:

$Contour_i = \{p_j | j \leqslant M_i, p_1 = p_{Mi}\}$, where M_i is the length of $contour_i$.

In the following section, we will present the classifier used to predict the class of the set of features. Then, we define the different steps of the proposed algorithm.

Table 2 Overview of the implemented features

Feature	Description	Dimension
f1	Run-length distribution of white pixels in four directions	10
f2	Run-length distribution of black pixels in four directions	100
f3	Run-length distribution of white and black pixels in four directions	10
f4	Edge-direction distribution using 16 angles	8
f6	Polygon-based features	512
f16	Chain-code-based local features	36
f23	Codebook-based features	112
f26	AR-coefficient-based features	220

4.4 Proposed classifier

4.4.1 *(K-NN) Classifier*

The K-Nearest-Neighbor (K-NN) classifier is one of the most basic classifiers for pattern recognition and data classification. The principle of this method is based on the intuitive concept that data instances of the same class should be closer in the feature space. As a result, for a given data point x of an unknown class, we can simply compute the distance between x and all the data points in the training data and assign the class determined by the K nearest points of x. Due to its simplicity, K-NN is often used as a baseline method in comparison with other sophisticated approaches utilized in pattern recognition. The K-Nearest-Neighbor classification divides data into a test set and a training set. In our case, we choose $K = 5$ to be used in a sample of 128 writers for both English and Arabic texts.

The main task of classification is to use the feature vectors provided by the feature extraction algorithm to assign the object to a category [37]. In our work, we use the K-Nearest Neighbors (K-NN) for the classification of the

Table 3 Detailed accuracy for left and right handwriting: *fuzzy 2-combinations*

Left and right handedness					
# of Att	δ	% Reduction	Precision	Recall	F-measure
f1-f2	0.995	17.7 %	65.69 %	66.86 %	66.27 %
f1-f3	0.85	*31.3* %	69.47 %	69.40 %	*69.43* %
f1-f4	0.92	20.0 %	62.81 %	62.80 %	62.80 %
f1-f23	0.995	34.3 %	64.55 %	64.43 %	64.48 %
f1-f6	0.995	5.9 %	63.55 %	61.93 %	62.73 %
f1-f26	0.998	36.9 %	67.23 %	66.89 %	67.05 %
f1-f16	0.993	23.8 %	63.85 %	63.58 %	63.71 %

extracted features. K-NN running on the Rapidminer platform [11] classifier classifies an unknown sample based on the known classification of its neighbors [38, 39]. Given unknown data, the K-Nearest-Neighbor classifier searches the pattern space for the K training data that are closest to the unknown data. These K training tuples are the K "Nearest Neighbors" of the unknown data. Typically, we normalize the values of each attribute. This helps to prevent attributes with initially large ranges from outweighting attributes with initially smaller ranges (such as binary attributes).

In this step, the features previously presented are used to predict the handedness of the writer of each document. When performing the classification, each element of the feature vector will be used as a separate input for the classifier (for example, *f1* will be an input vector of 10 elements for the classifier, as shown in Table 2). We have combined these features using a K-Nearest-Neighbor classifier [40]. A description of the combination of features using the (K-NN) classifier is given below.

4.4.2 Proposed algorithm

In this section, we give a description of the proposed algorithm based on a new heuristic approach for obtaining the best feature combination by applying the Lukasiewicz implication with variations of different values of δ. Only the best value of δ that provides the highest F-measure score was retained. The proposed approach is split into three sub-modules : (1) the main algorithm, which takes into consideration the input fuzzy binary relation for the features (f1, f2, f3, f4, f6, f16, f23 and f26) denoted $\mathcal{F}_{BR}$ (these features are presented in excel files in the form of fuzzy binary relations, where the rows represent the different writers based on their left/right handedness and the columns represent the values of features); (2) the remaining feature module processing that identifies the features to be rejected and the features that will be maintained according to the best value of δ that provides a high score of the F-measure and a considerable improvement in the data reduction percentage (which is accomplished by computing the closure of the Galois connection); and (3) the third sub-module, which determines the computed closure of the remaining attributes. In this case, we consider the fuzzy binary matrix relevant to a given feature F_i, where $\mathcal{O}_i$ represents an object and $A_0, A_1, \ldots, A_n$ represent the corresponding attributes. The rows represent the different writers (left and right handedness). We use 121 writers. The columns represent the measured values of the features. For instance, feature f26 describes the measured values of the "Polygon-based features" using 512 measures (i.e., 0.019815 represents a measure).

Step 1: Performing a matrix transposition; transpose the rows to the attributes and columns to the objects. Each row represents a different feature F_i, and the objects corresponding to 121 different writers with both left and right handedness are represented as columns.

Step 2: Choosing different arbitrary values of δ, we compute the closure of the attributes by using the following formula: $h_\delta \circ f(A) = \{A_0, A_1, \ldots, A_n\}$. The discovered redundant attributes are removed. Intuitively, an attribute is redundant if we can regenerate it by association from other attributes. Finally, we keep the last subset that contains the reduced subset of *Objects* × *Attributes* with the highest values of δ in terms of precision, recall and F-measure.

```
Algorithm 1: MAIN ALGORITHM
  Input: FUZZY WORKING RELATION $\mathcal{F}_{BR}$
  Output: REMAINEDFEATURES
  begin
      Initialize the input fuzzy binary relation to the
      fuzzy working relation relation
      $\mathcal{F}_{BR} \leftarrow$ INITIAL CONTEXT
      We denote that $R_i$ is denoted by $x$
      for each feature $x$ in the domain $\mathcal{F}_{BR}$ do
          LISTCLOSE $\leftarrow$ COMPUTECLOSURE($x$)
          //Compute the closure of $x$
          $S_x \leftarrow$ LISTCLOSE.CLOSURE ({X}) $- \{x\}$
          //Compute the closure of $Sx - \{x\}$
          $CL_{Sx} \leftarrow$
          LISTCLOSE.CLOSURE ($S_x - \{x\}$) $- \{x\}$
          if ($CL_{S_x} \equiv S_x$) then
              //Remove feature $x$
```

```
Algorithm 2: COMPUTING CLOSURE
  Input: VECTOR LISTSX
  Output: LISTVECTOR
  begin
      Compute the Galois connection of ListSx
      According to different values of $\delta$
      VECTORMIN $\leftarrow$ GALOISF(ListSx)
      compute the closure of Galois connection of
      ListSx LISTVEC $\leftarrow$ GALOISH(VectorMin, $\delta$)
```

In general, the sub-modules are composed of the following steps:

1. In the main algorithm, we determine for each row
   - The closure list "ListClose", which is denoted S_x and computed using the following formula: $\mathcal{H} \circ \mathcal{F}(x)$.

- The next step consists of removing from Listclose the redundant feature: The values of $\mathcal{F}_{\mathcal{BR}}$ (*i.e.*, $\mathrm{CL}_{Sx} \leftarrow \textsc{ListClose.closure} - (x)$).
- If CL_{Sx} is equivalent to S_x, then feature x is removed.

2. The second sub-module (Algorithm 2) consists of computing the Galois connection of $ListS_x$ according to the specified value of δ (i.e., CL_{Sx})
3. Another module may be added in order to update the context if CLS_x is equal to S_x.

To summarize, the algorithm falls into the following detailed steps:

- Compute the F-measure for each feature.
  1. Vary different values of δ (for example, 0.95), and generate the features that satisfy the Lukasiewicz implication according to the fixed value of δ;
  2. Choose the best results of the features and determine the percentage of reduction;
  3. Combine the feature with the highest score (e.g., the F-measure) with all the other features;
  4. Compute the F-measure for the combined two selected features; and
  5. Retain only the combined feature with the highest score.
- Repeat the steps above in a similar way for combinations of the next levels (3, 4 and so on) until no improvement is obtained.
- Select the combination of features with the highest F-measure score.

4.5 Results and their analysis

To evaluate our approach, we use the QUWI dataset presented earlier in this paper. We use standard evaluation metrics, including the precision, recall and F-measure (which is derived from the precision and recall [41]). In our experiments, we conduct the evaluation of the features after applying the feature reduction process using the Lukasiewicz implication for different values of δ (varying from 0..1). Recall that the precision of a class i is defined as

Table 4 Detailed accuracy for left and right handwriting: *fuzzy 3-combinations*

		Left and right handedness			
# of Att	δ	% Reduction	Precision	Recall	F-measure
f1-f3-f2	0.98	32.3 %	67.81 %	67.75 %	67.77 %
f1-f3-f4	0.94	16.7 %	70.27 %	70.26 %	70.26 %
f1-f3-f23	0.995	*33.3* %	71.99 %	71.87 %	*71.93* %
f1-f3-f6	0.997	9.9 %	68.91 %	68.58 %	68.59 %
f1-f3-f26	0.997	19.6 %	71.99 %	71.87 %	71.93 %
f1-f3-f16	0.994	24.4 %	71.99 %	71.87 %	71.93 %

$$Precision = \frac{\#documents\ Correctly\ Classified\ into\ Class\ i}{\#of\ documents\ classified\ into\ Class\ i}$$

and the recall of class i is defined as:

$$Recall = \frac{\#documents\ Correctly\ Classified\ into\ Class\ i}{\#of\ documents\ that\ are\ truly\ in\ class\ i}$$

and then the F-measure, which reflects the relative importance of the recall versus precision, is defined as

$$\text{F-measure} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$Accuracy = \frac{\alpha}{\beta}$ such that $\alpha = \alpha_1 + \alpha_2$ and $\beta = \alpha_1 + \alpha_2 + \beta_1 + \beta_2$ where

- $\alpha_1 =$ # of true documents correctly classified into $Class_i$
- $\alpha_2 =$ # of true documents incorrectly classified into $Class_i$
- $\beta_1 =$ # of false documents correctly classified into $Class_i$
- $\beta_2 =$ # of false documents incorrectly classified into $Class_i$

The precision, recall and F-measure metrics are used to evaluate our approach using English and Arabic texts. In this work, we have chosen the K-Nearest-Neighbors (K-NN) algorithm, which is widely used for classification, machine learning, and pattern recognition by data miners [42].

In the (K-NN) classifier, we have used a cross validation which is defined as follows:

- Divide training examples into two sets, a training set (95 %) and a validation set (5 %);
- Predict the class labels for the validation set by using the examples in the training set; and
- Choose the number of neighbors $K = 5$ that maximizes the classification accuracy.

Table 5 Detailed accuracy for left and right handwriting: *fuzzy 5-combinations*

		Left and right handedness			
# of Att	δ	% Reduction	Precision	Recall	F-measure
f1-f3-f23-f16-f2	0.995	8.9 %	66.94 %	66.94 %	66.94 %
f1-f3-f23-f16-f4	0.992	32.4 %	66.12 %	66.12 %	66.12 %
f1-f3-f23-f16-f6	0.990	11.7 %	70.27 %	70.23 %	70.25 %
f1-f3-f23-f16-f26	0.995	*31.3* %	71.99 %	71.87 %	*71.93* %

Table 6 Detailed accuracy for left and right handwriting: *fuzzy 6-combinations*

		Left and right handedness			
# of Att	δ	% Reduction	Precision	Recall	F-measure
f1-f3-f23-f16-f26-f2	0.990	29.7 %	66.15 %	66.09 %	66.12 %
f1-f3-f23-f16-f26-f4	0.997	18.8 %	66.97 %	66.95 %	66.96 %
f1-f3-f23-f16-f26-f6	0.992	*20.6* %	69.47 %	69.40 %	*69.43* %

4.5.1 Experiments with no feature reduction

The classification is carried out separately for the Arabic and English languages in a first step and jointly in a second step. The results are reported for the case of similar texts written by all the writers and different texts for each writer. In the following, we present the results of the classification at the end of each iteration:

First iteration: we compute the F-measure of the features separately. We present (1) the reduction percentage and (2) the improvement of the F-measure through application of the Lukasiewicz implication.

Second iteration: it is clear that feature f1 has the highest F-measure (i.e., 70.73 %), with a 20 % data reduction percentage. Therefore, we have combined f1 with each feature, a combination of two features (i.e., f6, f4, f3 and so on) and the results obtained are shown in Table 3. The combination strategy proves that combining a feature with a low recognition rate can provide a good result while reducing the amount of data. When we combine f1 with f3, the recognition rate drops by one point, but a data reduction of 31.3 % is obtained. This is almost 13 % improvement than the first result with no combination.

The highest F-measure score was 69.43 % for the two combined features f1 and f3 with a data reduction of 31.3 %, Table 3. Therefore, we have combined the latter with the other remaining features (e.g., f2 and f4). This prpocess is continued until the end where we only select the combination having the highest F-measure with the respective high-reduction percentage. The highest selected F-measure is marked in italics, as shown in the next Tables 4, 5, 6, 7, and 8.

The remaining classification rates obtained in each evaluation are given in Tables 4 through 8 for three to eight combinations, respectively. With the three combinations as depicted in Table 9, it can be seen that further improvements are achieved in the F-measure rates, with the best performance obtained when we combine f1, f3, and f23. Furthermore, the reduction rate improved by 3 % compared to the best result using two combinations. In the case of four combinations, (e.g., Table 4), the result of the F-measure remains almost the same as when using three combinations but with further reductions using f1-f3-f23-f4 and f1-f3-f23-f16.

Table 7 Detailed accuracy for left and right handwriting: *fuzzy 7-combinations*

		Left and right handedness			
# of Att	δ	% Reduction	Precision	Recall	F-measure
f1-f3-f23-f16-f26-f4-f2	0.995	*4.6* %	67.01 %	66.91 %	*66.96* %
f1-f3-f23-f16-f26-f4-f6	0.990	30.7 %			66.94 %

Table 8 Detailed accuracy for left and right handwriting: *fuzzy 8-combinations*

		Left and right handedness			
# of Att	δ	% Reduction	Precision	Recall	F-measure
f1-f3-f23-f16-f26-f4-f6-f2	0.995	*5.2* %	68.77 %	68.55 %	*68.66* %

The combined features f1-f3-f23-f16-f26 yield the highest score (F-measure is approximately equivalent to 71.93 %). Therefore, this feature will be combined with the other features (i.e., f26-f4, f26-f23, f26-f16, f26-f3, f26-f6, and f26-1). We continue the combination process until reaching the point where no possible combination of features that obtain a higher F-measure score is possible. In the previous tables, this is shown by the recognition rates and the reduction rates which start to decrease. Finally, the F-measure slightly improves with the 8-combination with only a 5 % reduction rate.

4.5.2 Summary of experiments using the Lukasiewicz implication

The scores, as shown in Table 5, are computed. The top scores for the combined features approach were Precision = 71.99 %, Recall = 71.87 %, F-measure = 71.93 %, and Accuracy = 71.86 +/- 7.74 % (71.90 %). The data reduction percentage was 31.3 %. This represents a considerable data reduction ability relative to the whole volume of data with an interesting improvement in the precision and recall metrics. Interestingly, computing confidence intervals for these results yields a confidence interval of (+/-5 %). Thus, we conclude that our approach

Table 9 Detailed accuracy for left and right handwriting: *fuzzy 4-combinations*

		Left and right handedness			
# of Att	δ	% Reduction	Precision	Recall	F-measure
f1-f3-f23-f2	0.995	12.1 %	66.96 %	66.93 %	66.94 %
f1-f3-f23-f4	0.992	46.8 %	67.02 %	66.97 %	66.99 %
f1-f3-f23-f6	0.995	33.3 %	71.99 %	71.87 %	71.93 %
f1-f3-f23-f26	0.997	16.2 %	71.99 %	71.87 %	71.93 %
f1-f3-f23-f16	0.992	*35.0* %	71.99 %	71.87 %	*71.93* %

Table 10 Different combinations

Left and right handedness	
Combination	Features selected
1-combination	f6
2-combination	f1-f3
3-combination	f1-f3-f23
4-combination	f1-f3-f23-f16
5-combination	f1-f3-f23-f16-f26
6-combination	f1-f3-f23-f16-f26-f6
7-combination	f1-f3-f23-f16-f26-f4-f2
8-combination	f1-f3-f23-f16-f26-f4-f6-f2

improves the quality of the obtained features and contributes enormously to the reduction of the number of features. Figure 5 results of different combined features.

4.5.3 Key findings

In the following, we provide a summary of the obtained results. We show the best results for each combination. We then graph these results in an appropriate figure. Finally, we comment on the results.

Discussions: Based on the conducted experiments, we provide the following remarks. The most striking feature is that according to the obtained scores, the highest number of labels is explored using the combined features f1-f3-f23-f16-f26 (the F-measure is approximately equal to 71.93 %).

- The accuracy is reasonable (69.78 % on average) in all approaches except for the combination f1-f3 (69.47 %). This is due to the quality of the features (the accuracy of f1 was 70.58 %, and the accuracy of f3 was 62.85 %). It reaches its maximum for the following combination (approximately 71.99 %): f1-f3-f23-f16-f26 (Table 10).
- The recall attains its highest values (71.87 %) for the combination f1-f3-f23-f16 and the lowest value (69.60 %) for the combination f1-f3-f23-f16-f26-f4-f2. It is clear that the features f26-f4-f2 did not improve the score. On average, the score was 69.60 %.
- The F-measure reaches its highest values (71.93 %) for the combination f1-f3-f23-f16 and the lowest value (69.69 %) for the combination f1-f3-f23-f16-f26-f4-f2. It is clear that the same features f26-f4-f2 did not improve the score. On average, the score was 69.69 %.
- The reduction percentage reaches its maximum (83.43 %) for the feature F6 alone and its minimum of 4.57 % for the feature f1-f3-f23-f16-f26-f4-f2, while on average, the reduction percentage was 30.36 %); and
- Finally, if one takes into consideration the highest F-measure with an improvement in the reduction percentage, it is clear that using f1-f3-f23-f16 with the F-measure (71.93 %) results in a percent reduction of 31.3 %, which emphasized a considerable improvement in dimensionality reduction of the features. Tables 11, 12 and 13.

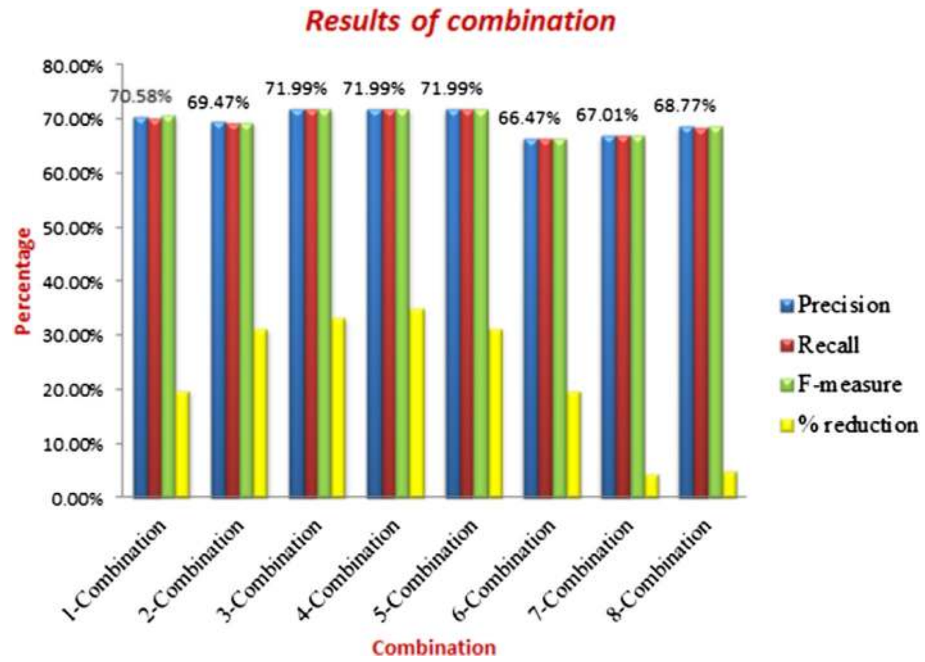


Fig. 5 Results of different combined features

Table 11 Detailed accuracy for left- and right-handwriting: *fuzzy 1-combination*

Left and right handedness					
# of Att	δ	% Reduction	Precision	Recall	F-measure
f26	0.997	74.6 %	66.12 %	66.12 %	66.12 %
f6	0.992	83.43 %	70.07 %	69.35 %	69.71 %
f4	0.92	33.3 %	68.67 %	68.57 %	68.61 %
f3	0.851	33.3 %	62.85 %	62.83 %	62.84 %
f16	0.998	5.6 %	59.55 %	59.47 %	59.50 %
f1	0.885	*20.0* %	70.58 %	70.19 %	*70.73* %
f23	0.999	13.4 %	68.00 %	67.72 %	67.86 %
f2	0.995	32.6 %	45.40 %	45.43 %	45.41 %

Table 13 Summary of the highest obtained results using different combinations

Left and right handedness					
Combination	δ	% Reduction	Precision	Recall	F-measure
1-combination	0.885 %	20 %	70.58 %	70.19 %	70.73 %
2-combination	0.85 %	31.3 %	69.47 %	69.40 %	69.43 %
3-combination	0.995 %	33.3 %	71.99 %	71.87 %	71.93 %
4-combination	0.992 %	35 %	71.99 %	71.87 %	71.93 %
5-combination	0.995 %	31.3 %	71.99 %	71.87 %	71.93 %
6-combination	0.992 %	20 %	66.47 %	66.40 %	69.43 %
7-combination	0.995 %	4.6 %	67.01 %	66.91 %	66.96 %
8-combination	0.995 %	5.2 %	68.77 %	68.55 %	68.66 %

4.6 Computational complexity reduction

We consider the features f1, f2, f3, f4, f6, f16, f23, and f26 for the computation of the complexity analysis of our approach. Thus, the computational complexity of the determination of the best combined features, for the previous n features, is determined as follows, as the objects correspond to features and attributes to writers. So, the time complexity is definitively $\mathcal{O}(n \star m^2)$ where n is the number of writers and m the number of features. As a matter of fact, we are calculating the closure of each one of the m features, where the closure requires m x n comparisons, where n is the number of writers.

5 Conclusion and future work

We have proposed a new generic approach for combined feature extraction based on a successive combination of the best feature with the highest score with every other feature. We plan to apply this approach to many applications including gender, age, and nationality prediction. The goal of this research consists of investigating the text-independent identification of a script writer. We have employed a set of features (e.g., f1, f2, f3, f4, f6, f16, f23, and f26), which have shown promising results on a database of handwritten documents in two different languages: Arabic and English. The evaluations were carried out on the only existing database of its type, containing short writing samples from 121 different writers. The results of the determination of the best combined feature identification are very encouraging (the F-measure is approximately equal to 71.08 %). They reflect the effectiveness of the run-length features in a text-independent script environment and validate the hypothesis put forward in this research, i.e., that the writing style remains approximately the same across different scripts. It is also worth mentioning that, unlike most of the studies that use complete pages of text, our results are based on a limited amount of handwritten text, which is more realistic. Another interesting aspect of this study was the evaluation and comparison of a number of state-of-the-art methods on this dataset. The features used in these methods naturally show a decrease in performance when exposed to different script scenarios. In all cases, the run-length features outperform these features. Finally, for the comparison of the proposed method with other methods, the average correct handedness detection results are over 83.43 %, which exceeds the results reported in [30] for off-line gender identification (70 %) on the same dataset. The results also compare well with the 73 %; 55.39 % reported for gender classification in [36, 43] on different datasets. It would be interesting to evaluate these features on a larger dataset with a large number of writers and many scripts per writer. This, however, involves the challenging task of finding individuals who are familiar with multiple scripts. To extend this study, we intend to utilize a database including writing samples in Arabic, French, and other languages provided by the same writer. In addition, classifiers other than those discussed in this paper can be evaluated to find out how they perform in a many-script environment. The proposed approach can also be extended to include a rejection threshold to reject writers that are not a part of the database. Finally, it would be interesting to apply a feature selection strategy to reduce the dimension of the proposed feature set and to study which subset of features is the most discriminative in characterizing the writers.

Table 12 Detailed accuracy for left and right handwriting using the Lukasiewicz implication

Left and right handedness					
# of Att	δ	% Reduction	Precision	Recall	F-measure
f1-f3					
f23-f16	0.995	*31.3* %	71.99 %	71.87 %	*71.93* %
f26					

The application of Latent Semantic Analysis (LSA) techniques seems promising regarding the reduction of the volume of data. These aspects will constitute the focus of our future research on writer recognition. With regard to data reduction, in our future work, we would like to investigate reducing the high-dimensional data of features gathered from user-cognitive loads, which results from the density of data to be visualized and mined, and reducing the dimensionality of the dataset while associations (or dependencies) between objects as applied to writer identification. This dimensionality reduction will be based on fuzzy conceptual reduction through the application of the Lukasiewicz implication.

Endnote

[1] https://www.kaggle.com/c/icdar2013-gender-prediction-from-handwriting.

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

This publication was made possible by a grant from the Qatar National Research Fund NPRP 09-864-1-128. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the QNRF.

Received: 25 September 2014 Accepted: 27 November 2015
Published online: 05 January 2016

References

1. A Hassaine, S Al Maadeed, J Aljaam, A Jaoua. ICDAR 2013 Competition on Gender Prediction from Handwriting, Twelfth International Conference on Document Analysis and Recognition ICDAR2013 (IEEE, Washington, DC, 2013). https://www.computer.org/csdl/-abs.html
2. S Impedovo, L Ottaviano, S Occhinegro, Optical character recognition. Int.J. Pattern Recognit. Artif. Intell. **5**(1-2), 1–24 (1991)
3. VK Govindan, AP Shivaprasad, Character recognition□ a review. Pattern Recognit. **23**(7), 671–683 (1990)
4. R Plamondon, SN Srihari, On-line and off-line handwritten character recognition: a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 63–84 (2000)
5. N Arica, F Yarman-Vural, An overview of character recognition focused on off-line handwriting. IEEE Trans. Syst. Man Cybernet. Part C: Appl. Rev. **31**(2), 216–233 (2001)
6. U Bhattacharya, BB Chaudhuri, Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. IEEE Trans. Pattern Anal. Mach. Intell. **31**(3), 444–457 (2009)
7. K Bandi, SN Srihari, in *Proceedings of the International Graphonomics Society Conference (IGS)*. Writer demographic identification using bagging and boosting (Publisher International Graphonomics Society (IGS), 2005), pp. 133–137. http://www.graphonomics.org/publications.php
8. S Srihari, SH Cha, H Arora, S Lee, in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*. Individuality of handwriting: a validation study (IEEE, 2001), pp. 106–109
9. M Liwicki, A Schlapbach, P Loretan, H Bunke, in *Proceedings of the 13th Conference of the International Graphonomics Society*. Automatic detection of gender and handedness from on-line handwriting (Publisher International Graphonomics Society (IGS), 2007), pp. 179–183. http://www.graphonomics.org/publications.php
10. M Liwicki, A Schlapbach, H Bunke, Automatic gender detection using on-line and off-line information. Pattern. Anal. Appl. **14**, 87–92 (2011)
11. KDnuggets, Data integration, analytical ETL, data analysis, and reporting, rapid miner journal (2012). Software available at, http://sourceforge.net/projects/rapidminer/
12. B Ganter, R Wille, *Formal Concept Analysis*. (Springer-Verlag, Berlin Heidelberg, 1999), p. 283
13. L Wang, in *Proceedings Part I of the Second International Conference, (FSKD 2005), Changsha, China*. Fuzzy Systems and knowledge discovery (Springler-Verlag Berlin, Heidelberg, 2005), pp. 515–519. ISBN 10 3–540–28312–9
14. B Ganter. *Two basic algorithms in concept analysis*. Preprint 831, Technische, (Hochschule Darmstadt, Germany, 1984)
15. G Birkhoff. Lattice Theory First edition, Providence: American. Mathemathics Society (Springler-Verlag Berlin, Heidelberg, 2005), pp. 515–519. ISBN 10 3–540–28312–9
16. LA Zadeh, Fuzzy sets, information and control. **8**, 338–353 (1965)
17. S Elloumi, J Jaam, A Hasnah, A Jaoua, I Nafkha, A multi-level conceptual data reduction approach based on the Lukasiewicz implication. Inf. Sci. **163**, 253–262 (2004)
18. J Riguet, Lattice Theory First edition, Relations binaires, fermetures et correspondances de Galois. Bull.Soc. Math. France. **78**, 114–155 (1948)
19. R Belohlavek, Fuzzy Galois connections. Math. Logic Quart. **45**, 497–504 (1999)
20. R Belohlavek, Lattices of fixed points of Galois connections. Math.Logic Quart. **47**, 111–116 (2001)
21. A Frascella, Lattice Theory First edition, Fuzzy Galois connections under weak conditions, fuzzy sets and systems. **172**, 33–50 (2011)
22. SN Srihari, H Arora, SH Cha, S Lee, Individuality of handwriting. J.Forensic Sci. **47**(40), 1–17 (2002)
23. HE Said, TN Tan, KD Baker, Personal identification based on handwriting. Pattern Recognit. **33**(1), 149–160 (2000)
24. A Bensefia, T Paquet, L Heutte, A writer identification and verification system. Pattern Recognit. Lett. **26**(13), 2080–2092 (2005)
25. M Bulacu, L Schomaker, L Vuurpijl, in *Seventh International Conference on Document Analysis and Recognition*. Writer identification using edge-based directional features (IEEE, 2003), pp. 937–941
26. A Schlapbach, H Bunke, in *9th Int.Workshop on Frontiers in Handwriting Recognition*. Using HMM based recognizers for writer identification and verification (IEEE, 2004), pp. 167–172
27. M Bulacu, L Schomaker, Text-independent writer identification and verification using textural and allographic features. IEEE Trans. Pattern Anal. Mach. Intell. **29**(4), 701–717 (2007)
28. I Siddiqi, N Vincent, Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. Pattern Recognit. **43**(11), 3853–3865 (2010)
29. U Pal, T Wakabayashi, F Kimura, in *Ninth International conference on Document Analysis and Recognition ICDAR 07*. Handwritten numeral recognition of six popular scripts, vol. 2 (IEEE Computer Society, Washington, DC, USA, 2007), pp. 749–753. ISBN:0-7695-2822-8
30. S Al Maadeed, F Ferjani, S Elloumi, Hassaine Ai, Jaoua A, in *2013 IEEE GCC Conference and exhibition, November 17□20, Doha, Qatar*. Automatic handedness detection from off-line handwriting (IEEE, 2013), pp. 119–124. ISBN: 978-1-4799-0722-9
31. UV Marti, R Messerli, H Bunke, in *proceedings of Sixth International Conference on Document Analysis and Recognition*. Writer identification using text line based features (IEEE Computer Society, Washington, DC, USA, 2001), pp. 101–105
32. EN Zois, V Anastassopoulos, Morphological waveform coding for writer identification. Pattern Recognit. **33**, 385–398 (2000)
33. S Al-Ma□deed, W Ayouby, A Hassaine, J Aljaam. QUWI: An Arabic and English handwriting dataset for off-line writer identification, *International Conference on Frontiers in Handwriting Recognition* (IEEE, 2012), pp. 746–751. ISBN: 978-1-4673-2262-1
34. N Otsu, A threshold selection method from gray-level histograms. Automatica. **11**, 285–296 (1975)
35. A Hassaine, S Al-Maadeed, J AlJaam, A Jaoua, A Bouridane. The ICDAR2011 Arabic writer identification Contest, *Proc. Eleventh International Conference on Document Analysis and Recognition,Beijing, China* (IEEE, 2011)
36. S Al Maadeed, A Hassaine, Automatic prediction of age, gender, and nationality in offline handwriting. EURASIP J Image Video Process. **2014**, 10 (2014)
37. RO Duda, PE Hart, DG Stork, *Pattern Classification*, second edition. (John Wiley & Sons Inc., New York, 2000)

38. BV Dasarathy, *Nearest Neighbor: Pattern Classification Techniques*. (IEEE Computer Society Press, New York, 1990)
39. Y Yang, in *Proc.17th Annual Intl. ACM SIGIR Conf. Research and Development in Information Retrieval, Dublin (Ireland)*. Expert network: effective and efficient learning from human decisions in text categorization and retrieval (ACM, 1994), pp. 13–22
40. L Breiman, Random forests. Mach. Learn. **45**(11), 5–32 (2001)
41. G Salton, J Michael, *An Introduction to Modern Information Retrieval*. (McGraw-Hill, New York, 1983)
42. KDnuggets, Data integration, analytical ETL, data analysis, and reporting, rapid miner (2012). Software available at http://sourceforge.net/projects/rapidminer/
43. M Liwicki, A Schlapbach, H Bunke, Automatic gender detection using on-line and off-line information. Anal. Appl. **14**, 87–92 (2011)