Pattern Recognition 78 (2018) 318–330

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

An improved online writer identification framework using codebook descriptors

Vivek Venugopal, Suresh Sundaram*

Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, Guwahati 781039, India

ARTICLE INFO

Article history:
Received 28 August 2017
Revised 3 November 2017
Accepted 24 January 2018
Available online 1 February 2018

Keywords:
Online writer identification
Codebook
Descriptor
Gap parameter,

ABSTRACT

This work proposes a text independent writer identification framework for online handwritten data. We derive a strategy that encodes the sequence of feature vectors extracted at sample points of the temporal trace with descriptors obtained from a codebook. The derived descriptors take into account, the scores of each of the attributes in a feature vector, that are computed with regards of the proximity to their corresponding values in the assigned codevector of the codebook. A codebook comprises a set of codevectors that are pre-learnt by a k-means algorithm applied on feature vectors of handwritten documents pooled from several writers. In addition, for constructing the codebook, we consider features that are derived by incorporating a so called 'gap parameter' that captures characteristics of sample points in the neighborhood of the point under consideration. We formulate our strategy in a way that, for a given codebook size k, we employ the descriptors of only $k-1$ codevectors to construct the final descriptor by concatenation. The usefulness of the descriptor is demonstrated by several experiments that are reported on publicly available databases.

1. Introduction

The problem of writer identification refers to the task of deciding on the authorship of a piece of handwritten document by comparing it against a set of samples saved in a database [1]. Based on the mode of data capture, such systems are categorized into either online or offline. The recent advances in technology has enabled the release of hand held devices, wherein the data entry is captured through an electronic pen / stylus. The tip of the stylus, as such, has the capability to capture the trajectory information such as (x, y) coordinates, time stamp and pen status from the handwriting. In the literature of writer identification, the analysis of such temporal data is referred to as 'online'. The input to such a system consists of a set of strokes, each of which containing the sample points of the trace captured between a pen-down and pen-up signal. Off-line writer identification systems, on the other hand, capture the data as an image and subsequently establish the authorship by applying image processing techniques [2–5].

Another classification for online writer identification systems are those of text dependent and text independent approaches [6]. In the former, the handwriting samples of a writer are processed based on a specific transcript usually with the aid of a recognizer. The problem of signature recognition / verification is one such popular instance of text dependent writer identification [7–9]. In general, the use of the knowledge of the content of the data increases the accuracy of such systems. However, they fail in scenarios that require the textual contents of the documents to be different. Hence, as an alleviation to this, text independent writer identification systems capture the style information of handwriting and can identify the writer irrespective of the textual content. In this research, we focus our proposal towards such a system.

With regards to prior techniques being proposed for online writer identification, a popular one is that of the Gaussian Mixture Model-Universal Background Model (GMM-UBM) [10–12] - inspired from the domain of speaker identification [13]. The enrolled handwritten data from a set of writers is first used to learn the parameters of the UBM by employing the Expectation Maximization (EM) algorithm. Thereafter, individual GMMs are obtained for every writer from the UBM by applying the MAP adaptation on their corresponding training samples. The authorship of a test document is assigned to that writer, whose corresponding GMM outputs the highest log-likelihood score.

A number of strategies related to the domain of information retrieval have been explored in the literature of online writer identification. A score derived from the term-frequency inverse-document-frequency (tf-idf) weighing scheme is employed to represent the handwritten document of unknown authorship in [14–17]. Likewise, the works of [18–20] consider the concept of La-

* Corresponding author.
E-mail address: sureshsundaram@iitg.ernet.in (S. Sundaram).

https://doi.org/10.1016/j.patcog.2018.01.023

tent Dirichlet Allocation from the area of topic models. Here, each document is modelled with the assumption of being a combination of finite (shared) writing styles, that in turn is a combination of a set of text independent feature probabilities. The idea of subtractive clustering is explored to determine the unique writing styles / prototypes of a writer in [21]. Thereafter, for writer identification, a modified tf-idf approach and a nearest neighbor based method are considered. In the work [22], the idea of multi-fractals are used to model the segmented graphemes / substrokes. Subsequent to it, the authors utilize a weighting based on tf-idf framework. The scoring of the tf and idf term is based on a frequentist approach, that in a way, characterises the number of segmented grapheme patterns assigned to a given codevector in a codebook. In addition, based on the grouping of the graphemes for Persian / Arabic script, separate codebooks are constructed for each of them. Another work is that of Dwivedi et al. [23], where the exploration of a sparse coding framework has been proposed for learning the different prototypes of a given writer, which are then utilized for establishing his / her identity. The features considered for obtaining the sparse coefficients are extracted on the segmented substrokes and are motivated from the idea of Histogram of Gradients [24] popular in the area of computer vision. Finally, the tf-idf framework is incorporated on the sparse representation coefficients to characterise the author of the document.

Coming to other explorations, a Bayesian framework is utilized for identifying the writer of the handwritten text by considering the shape primitives that are segmented from the online trace [6]. The authors of [25] represent the dynamic features such as speed, pressure and shape of the temporal trace as a sequence of codes for writer identification. The probability density distribution of four dynamic features from each pre-identified stroke type are used to describe the individuality characteristics of a writer in [26]. Likewise, in another work [27], the distribution of the shape primitives in handwritten text are employed to characterize the orientation of writing trajectory. This is utilized in conjunction with the distribution of curvature and other dynamic features in a hierarchical matching scheme. In [28], a fusion of Dynamic Time Warping (DTW) and Support Vector Machine (SVM) approach has been presented for identifying the authorship of Arabic texts. The extraction of features in this work was done with regards to different levels such as the point, the stroke and the space between strokes.

In a recent work, a Beta-Elliptic model is proposed to characterize the velocity and spatial profiles of the substrokes – followed by classification using an ensemble of Multi Layer Perceptrons (MLP) [29–31]. Last but not the least, the utility of a Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) for learning the feature representation of an online handwritten document are explored in [32,33] , wherein promising results have been quoted over the prior works that used hand-crafted features.

Based on the preceding enumeration of works, we infer that the explorations being proposed for online writer identification have been inspired from areas such as speaker identification [10–12,21,28] and information retrieval [14–20,22,23].

2. A roadmap to the proposed strategy

Recent techniques (since year 2010) in the area of object retrieval have focussed on descriptors obtained from a codebook [34–38]. In this paper, we consider the notion from such methods to motivate the study of online writer identification. An important ingredient in developing a writer identification system is that of a codebook – that comprises a set of codevectors representing the frequently occurring writing patterns among writers in an average sense. The descriptors in our proposal are aimed to capture the relative location of the feature vectors corresponding to the handwriting samples of a writer with respect to their nearest codevector in the feature space. The idea of the same comes from the intuition that the relative location of feature vectors of the same writer are more or less aligned in near-by proximity in the feature space. However, such a trend may not be prevalent when considering the feature vectors across different writers.

Keeping the aforementioned discussion in perspective, we propose a strategy that encodes the sequence of feature vectors along the online trace of a document with descriptors from a codebook. Based on a distance criterion, each feature vector of the document is assigned to a specific codevector in the codebook. As we shall see in Section 5, the proposed descriptors take into consideration, the scores of each of the attributes[1] in a feature vector with regards of the proximity to their corresponding value in the assigned codevector. We show in Section 6 that such explicit scoring can provide useful cues for better discriminating handwritten samples of the different writers enrolled to the system, when compared to the Vector of Locally Aggregated (VLAD) descriptor.

The utility of codebook descriptors was first investigated by the authors in [39]. This article presents improvements with regards to the strategy, that provide higher writer identifications together with an extensive performance evaluation demonstrating its efficacy. These are outlined as follows:

1. The derivation in Section 5 is formulated in a way that, for a given codebook size k, we employ the descriptors of only $k-1$ codevectors to construct the final description (by concatenation) for the online handwritten document. This is different from the formulation in [39], wherein the descriptors from all the k codevectors had to be considered for describing the handwritten document. Likewise, while computing the descriptors of each of the $k-1$ codevectors in the present proposal, we take into regard the scores of the attributes of the feature vectors from the entire document.
2. For constructing the codebook, and subsequently the descriptor, we derive features / attributes (in Section 4) by incorporating a gap parameter that aids in capturing the characteristics of sample points in the neighborhood of the point under consideration. A study is also conducted on the variation of the writer identification rate in Section 8.2 with different values of the gap parameter.
3. One of the detailed steps described in [39] was that of the regression based normalization, that was applied to the features prior to generation of the codebook. The main premise to adopting such a normalization was to ensure that the codevectors obtained are not influenced by outliers. As an avoidance to this, we consider in Section 3 of this paper, a preprocessing step for removing the isolated sample points. Moreover, in lieu of the regression based normalization, we transform the derived point based features obtain across a document, so that they have zero mean and unit variance (z- score normalization). This indeed helps in saving computation load in the normalization step.
4. An empirical evaluation of the effectiveness of our proposal with several recent variants of VLAD [34–37] is presented in Section 8.3.
5. We describe a reduced $D \times (k-1)$ dimension variant of the descriptor in Section 8.4 and demonstrate its performance in writer identification.
6. Experiments of our proposal are conducted on several publicly available online handwritten databases. On the whole, we ob-

[1] We refer to the feature values in a feature vector as ‘attributes’. The attributes computed at each sample point of the online trace are stacked to form a feature vector. Accordingly, elsewhere in the paper, we interchangeably use ‘features’ or ‘attributes’.

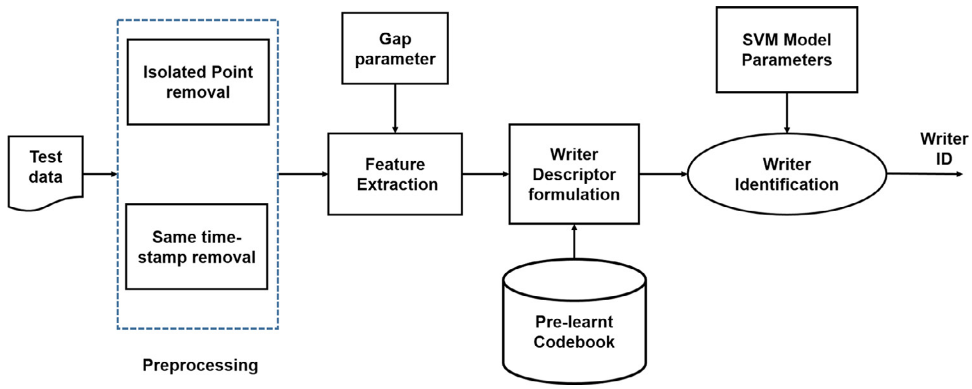

Fig. 1. Block diagram of the proposed text independent writer identification methodology. The test data refer to paragraphs or text lines written by the enrolled writers.

tain improved results in writer identification when compared to Venugopal and Sundaram [39].

7. We also evaluate the performance of our scheme on a multi language writer identification set-up.

Fig. 1 presents a block schematic of the proposed writer identification system. The temporal (*x, y*) sequence from a test document of unknown authorship is first passed through pre-processing module. Thereafter, by incorporating a gap parameter, a set of attributes / features are constructed at each sample point of the online trace in the feature extractor module. The pre-learnt codebook is obtained by applying the *k*-means algorithm on feature vectors pooled from several samples of handwritten documents of writers enrolled to the system. By employing the codevectors, we compute the proposed descriptor that encapsulates the writer specific characteristic of the handwritten test document. Finally, a SVM classifier is used to retrieve the identity of the writer.

2.1. Organization of the paper

The remainder of this paper is organized as follows: We discuss the details of the preprocessing and the feature extraction modules in Sections 3 and 4, respectively. Thereafter, the derivation of the proposed codebook descriptor for writer representation is elaborated at length in Section 5. A brief discussion of our proposal to VLAD is provided in Section 6, where we highlight the main differences. Subsequent to this, the datasets being used in this work with their enrolment and evaluation protocols are discussed in Section 7. The effectiveness of the proposed writer identification framework is demonstrated in Section 8 with several experiments. A comparison of our method with regards to previous works on online writer identification is provided in Section 9. In Section 10, we summarize our work.

3. Preprocessing

The preprocessing steps employed in this study are as follows.

Isolated sample point removal: We consider a neighbourhood of points within the same stroke around the sample point (say $\mathbf{p}$) under consideration. The point $\mathbf{p}$ is identified as an isolated point if its Euclidean distances to all of the remaining points is greater than a threshold *t*.

The value of *t* is empirically chosen from the document. We compute the mean *m* and standard deviation σ from the Euclidean distances between the successive sample points of all the strokes. Using these values, we remove points from the stroke whose distances to their neighboring sample points are greater than $m + (3 \times \sigma)$.

Same time stamp removal: Due to possible timing issues in data capture, there could be points with different *x* and *y* coordinates but with the same time stamp. As a result of this, features dependent on the time stamp difference like speed could turn out to be infinite. To avoid such values, we consider the *x* and *y* coordinates amongst the set of points having same time stamp and average their values.

4. Feature extraction

In this work, we consider a set of attributes / features, that are proposed at each sample point of the online trace of the handwritten input. As we shall see in this section, a subset of them are computed between a pair of points $(\mathbf{p}_i, \mathbf{p}_{i+r})$ in a stroke, where $\mathbf{p}_i = (x_i, y_i)$, $r > 0$. Said in another way, these features are derived by incorporating the information of sample points whose indices are spaced by a gap *r*. Accordingly, in this work, we hereinafter refer to this value as a 'gap parameter'. The use of $r = 1$ implies that we use consecutive sample points for deriving the attributes - the values of which at times may be influenced by the unintentional trembling of hand / jitter during the writing process [40]. Therefore we consider a higher value of *r* for feature extraction.[2]

In addition to the above, we also consider the parameter *r* to capture the information in the neighborhood around the sample point $\mathbf{p}_i$ by deriving vicinity based features. Accordingly, based on the preceding notion, we enumerate the proposed attributes / features below:

- *Speed*: computed at the sample point $\mathbf{p}_i$ as:

$$v_i = \frac{\Delta(\mathbf{p}_i, \mathbf{p}_{i+r})}{t_{i+r} - t_i} \tag{1}$$

  Here $\Delta(\mathbf{p}_i, \mathbf{p}_{i+r})$ is the Euclidean distance between the sample points $(x_i,\ y_i)$ and (x_{i+r}, y_{i+r}) that are spaced with a gap *r*. The denominator $t_{i+r} - t_i$ is the difference between their corresponding time stamps.
- *Writing direction*: The writing direction at $\mathbf{p}_i$ is defined by two features, that denote the cosine and sine of the angle θ_i made by the line segment joining $\mathbf{p}_i$ and $\mathbf{p}_{i+r}$ with the horizontal axis.

$$cos(\theta_i) = \frac{x_{i+r} - x_i}{\Delta(\mathbf{p}_i, \mathbf{p}_{i+r})} \qquad sin(\theta_i) \quad = \frac{y_{i+r} - y_i}{\Delta(\mathbf{p}_i, \mathbf{p}_{i+r})} \tag{2}$$

- *Curvature*: These two features / attributes relate to the cosine and the sine of the angle ϕ_i, computed using the vectors $\mathbf{p}_{i+r} - \mathbf{p}_i$ and $\mathbf{p}_i - \mathbf{p}_{i-r}$ respectively.
- *Vicinity aspect* : is the height to width ratio of the bounding box (*BB*) encompassing the $2r+1$ sample points $\{\mathbf{p}_{i-r}, ..., \mathbf{p}_i, ..., \mathbf{p}_{i+r}\}$ in the vicinity of $\mathbf{p}_i$.

[2] The choice for the higher value of *r* is justified in Section 8.2 by an empirical study that demonstrates its influence on the performance of the identification system.

- *Vicinity curliness*: is defined as the ratio of the length of the trajectory to the maximum amongst the width and height of *BB*.

Each of the above features is normalized to zero mean and unit variance. Accordingly, we represent the set of N_T normalized feature vectors for a test document T by $\{\mathbf{f}^1, \mathbf{f}^2, \ldots, \mathbf{f}^{N_T}\} = \{\mathbf{f}^j\}_{j=1}^{N_T}$. Here $\mathbf{f}^j$ is the D dimensional feature vector whose attributes / features can be written as : $\mathbf{f}^j = [f^j(1), \ldots, f^j(d), \ldots, f^j(D)]$. It may be noted that the value of D is seven in our work.

5. Proposed codebook descriptor

In deriving our descriptor, it is assumed that a codebook comprising k codevectors $\{\boldsymbol{\mu_i}\}_{i=1}^k$ is known to us in advance. The codevectors are obtained by applying the k-means algorithm on the D dimensional feature vector sequences corresponding to the documents of randomly picked subset of writers.

Given a test document T comprising N_T points, we assign each of the feature vectors to their nearest codevector. Let $\{\mathbf{f}_i^p\}_{p=1}^{n_i}$ denote the feature vectors contained in the Voronoi region of codevector $\boldsymbol{\mu_i}$. By definition, n_i represents the number of feature vectors and satisfies $n_i < N_T$ and $\sum_{i=1}^k n_i = N_T$.

We define two scores $S_i^{p+}(d)$ and $S_i^{p-}(d)$ for the d^{th} attribute of the p^{th} feature vector assigned to the codevector $\boldsymbol{\mu_i}$.

$$S_i^{p+}(d) = \begin{cases} \dfrac{1}{1 + |f_i^p(d) - \mu_i(d)|} & f_i^p(d) \geq \mu_i(d) \\ 0 & otherwise \end{cases}$$
$$S_i^{p-}(d) = \begin{cases} \dfrac{-1}{1 + |f_i^p(d) - \mu_i(d)|} & f_i^p(d) < \mu_i(d) \\ 0 & otherwise \end{cases} \tag{3}$$
$$1 \leq p \leq n_i, \;\; 1 \leq d \leq D$$

By notation, $\mu_i(d)$ represents the value of the d^{th} attribute in the codevector $\boldsymbol{\mu_i}$. The negative sign in the formulation of $S_i^{p-}(d)$ is an implication of the value of $f_i^p(d)$ being less than $\mu_i(d)$. Further to this, it may be observed from Eq. (3), that we are essentially capturing in a way the proximity of each of the attributes in a feature vector to their corresponding values in the assigned codevector. In particular, the term $|f_i^p(d) - \mu_i(d)|$ denotes the absolute value of the residual / distortion value between $f_i^p(d)$ and $\mu_i(d)$.[3] Note that, the more proximal the feature attribute $f_i^p(d)$ is to $\mu_i(d)$, the higher is the absolute value of the score $S_i^{p-}(d)$ or $S_i^{p+}(d)$.

With regards to obtaining the descriptor of the codevector $\boldsymbol{\mu_i}$, the scores obtained for the d^{th} feature attribute of the feature vectors $\{\mathbf{f}_i^p\}_{p=1}^{n_i}$ are accumulated and normalized with regards to the values obtained from the entire document as follows:

$$\tilde{S}_i^{+}(d) = \frac{\sum_{p=1}^{n_i} S_i^{p+}(d)}{\sum_{j=1}^{k}\sum_{p=1}^{n_j} S_j^{p+}(d)}$$
$$\tilde{S}_i^{-}(d) = \frac{\sum_{p=1}^{n_i} S_i^{p-}(d)}{\sum_{j=1}^{k}\sum_{p=1}^{n_j} S_j^{p-}(d)} \tag{4}$$
$$1 \leq d \leq D$$

The numerator terms $\sum_{p=1}^{n_i} S_i^{p+}(d)$ and $\sum_{p=1}^{n_i} S_i^{p-}(d)$ with $1 \leq d \leq D$ represent respectively, the aggregation of scores from the positive and negative distortion values corresponding to a codevector $\boldsymbol{\mu_i}$.

In our work, the normalized scores in Eq. (4) for each of the D feature attributes are concatenated as the descriptor $\mathbf{S}_i$ for codevector $\boldsymbol{\mu_i}$.

$$\mathbf{S}_i = \left[\tilde{S}_i^{+}(1) \;\; \tilde{S}_i^{-}(1) \;\; \cdots \;\; \tilde{S}_i^{+}(d) \;\; \tilde{S}_i^{-}(d) \;\; \cdots \;\; \tilde{S}_i^{+}(D) \;\; \tilde{S}_i^{-}(D)\right]^T \tag{5}$$

It is interesting to note from Eq. (4) that the scores $\tilde{S}_i^{+}(d)$ and $\tilde{S}_i^{-}(d)$ for the d^{th} feature attribute obtained across the k codevectors sum to one. Owing to this constraint, the document can be represented by concatenating descriptors corresponding to any of the $k-1$ codevectors. In our work, we consider the descriptors of the first $k-1$ codevectors. Put in another way, the final descriptor $\mathbf{S}$ is the concatenation of $\{\mathbf{S}_i\}_{i=1}^{k-1}$

$$\mathbf{S} = \left[\mathbf{S}_1 \quad \mathbf{S}_2 \quad \cdots \quad \mathbf{S}_{k-1}\right]^T \tag{6}$$

We see that the dimension of $\mathbf{S}$ is $2 \times D \times (k-1)$. For clarity, the pseudocode of our proposed codebook descriptor derived from the first $k-1$ codevectors is shown in Algorithm 1. On a passing note, in the scenario when none of the points in the document T are assigned to a codevector $\boldsymbol{\mu_i}$, the $2 \times D$ entries of the corresponding descriptor $\mathbf{S}_i$ are set to zero.

Given the codebook descriptor $\mathbf{S}$, the identity of the writer is established using a Support Vector Machine (SVM) classifier. Considering that we are using a one versus all multi-class SVM, the test document is assigned to the writer corresponding to the largest value of the discriminant function. As far as the training of the SVM is concerned, the Radial Basis Function (RBF) is used as the kernel with its parameters C and γ being selected using grid

Algorithm 1 Proposed Codebook descriptor generation.

- **Input:**
  - Document containing N_T points.
  - Codebook with a set of k codevectors $\{\boldsymbol{\mu_i}\}_{i=1}^k$.
- **Output:**
  - Codebook descriptor $\mathbf{S}$.
- **Algorithm:**

For a given gap parameter value r, compute the D dimensional feature vector sequence $\{\mathbf{f}^j\}_{j=1}^{N_T}$.
Assign each feature vector $\mathbf{f}^j$ to the nearest codevector in the codebook.

```
% Descriptor Computation:
```

for every codevector in $\{\boldsymbol{\mu_i}\}_{i=1}^{k-1}$ **do**
  Compute n_i – the number of feature vectors assigned to codevector $\boldsymbol{\mu_i}$.
  for every feature vector assigned to $\boldsymbol{\mu_i}$ **do**
    Compute $S_i^{p+}(d)$ and $S_i^{p-}(d)$ scores across the D attributes.
  end for
end for

for every codevector in $\{\boldsymbol{\mu_i}\}_{i=1}^{k-1}$ **do**
  Compute $\tilde{S}_i^{+}(d)$ and $\tilde{S}_i^{-}(d)$ scores across the D attributes.
end for

Stack $\tilde{S}_i^{+}(d)$ and $\tilde{S}_i^{-}(d)$ scores across the D attributes to get $\mathbf{S}_i$.
Concatenate descriptors $\{\mathbf{S}_i\}_{i=1}^{k-1}$ to obtain the final descriptor $\mathbf{S}$.

[3] Without loss of generality, the distortion / residual value can be positive or negative, depending on the sign of the difference $f_i^p(d) - \mu_i(d)$.

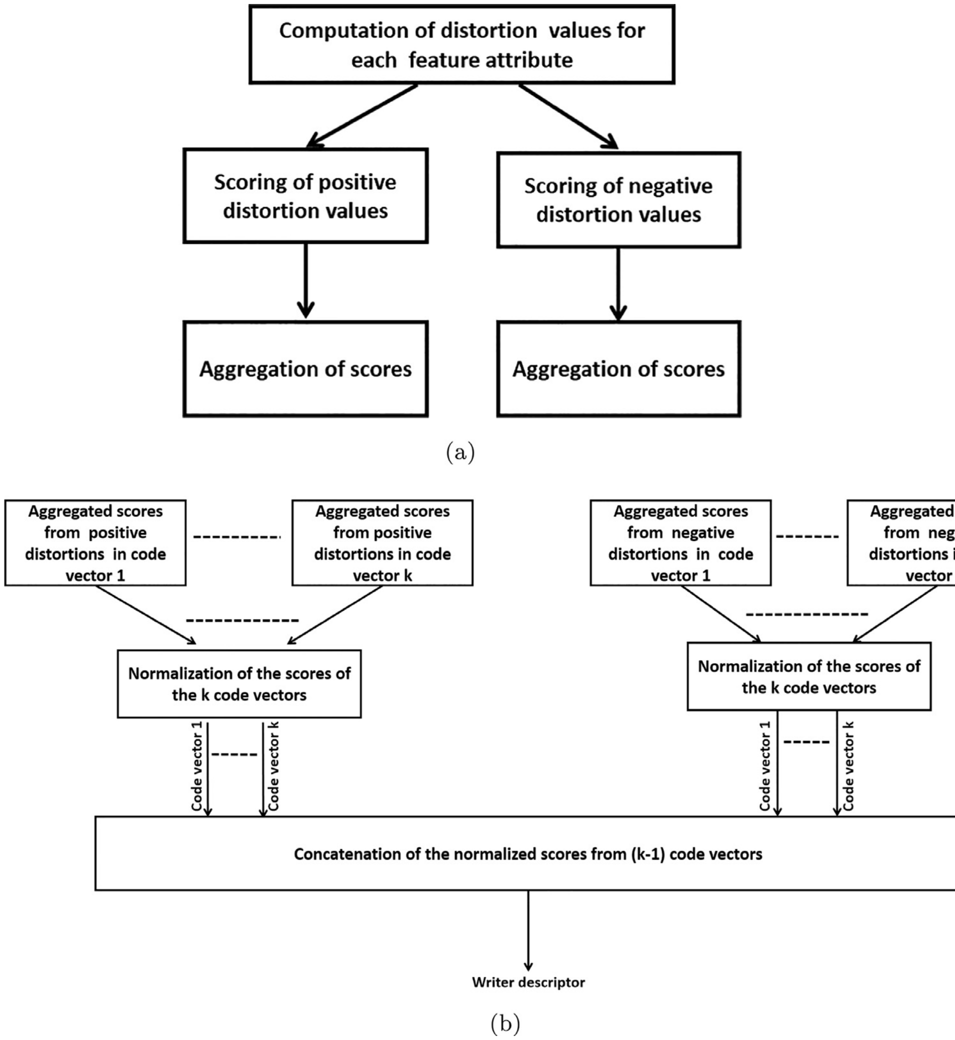

Fig. 2. Block diagram overview of the steps of our proposed code-book descriptor approach. For better clarity, we present the scoring strategy based on distortion values of the D feature attributes in a given code-vector in sub-figure (a). The distortion values for each attribute are scored separately depending on whether they are positive or negative. The aggregated scores of the attributes from the k codevectors are normalized and subsequently used to represent the writer descriptor, as highlighted in sub-figure (b).

search. The choice of this kernel is attributed to its encouraging results with our strategy, amongst those available in LIBSVM Toolkit.

We conclude this Section by a block schematic in Fig. 2 summarizing the steps of our proposed code-book descriptor approach. The distortion values of the D feature attributes in a given code-vector are scored according to their sign and aggregated, as depicted in sub-figure (a). The resulting aggregated scores of the D attributes from each of the codevectors are then normalized and used to represent the writer descriptor, as highlighted in sub-figure (b).

6. Discussion to the VLAD

As stated earlier, our proposal has been inspired by the success of codebook descriptors for the application of object retrieval in image processing [34–38]. However, at the same time, we highlight that our devised strategy is quite different in the formulation step.

To emphasize this distinction further, we discuss the Vector of Locally Aggregated Descriptor (VLAD) [34] as follows – with regards to each codevector in the codebook, the VLAD accumulates the residuals / distortions obtained from all the feature vectors assigned to it. In this case, the expression of $\mathbf{S}_i$ for the codevector $\boldsymbol{\mu_i}$ is given by:

$$\mathbf{S}_i = \sum_{p=1}^{n_i} (\mathbf{f}_i^p - \boldsymbol{\mu_i}) \qquad 1 \le i \le k \tag{7}$$

The vectors $\{\mathbf{S}_i\}_{i=1}^{k}$ are concatenated to give rise to the final descriptor $\mathbf{S}$ of dimension $D \times k$, which is subsequently L_2 normalized.

Between our proposal and the VLAD, there are two differing aspects worth noting.

- In the VLAD formulation of Eq. (7), both high and low distortions are given same preference. This is owing to the fact that each of the components $f_i^p(d) - \mu_i(d),\ 1 \le d \le D$ in a residual vector $\mathbf{f}_i^p - \boldsymbol{\mu_i}$ contribute equally to the summation. This strategy, at times, can present a limitation for the writer identification problem [39]. However, our proposal alleviates on the same by considering different scores / votes to each distortion component $f_i^p(d) - \mu_i(d)$ in the residual vector $\mathbf{f}_i^p - \boldsymbol{\mu_i}$. As seen from Eq. (3), we give high scores to low distortion components and vice versa. Thereafter, in place of summing up the distortion components as done with the VLAD, we accumulate their encoded scores instead.
- Secondly, the VLAD merely accumulates the distortion components – each of which may be either positive or negative. Con-

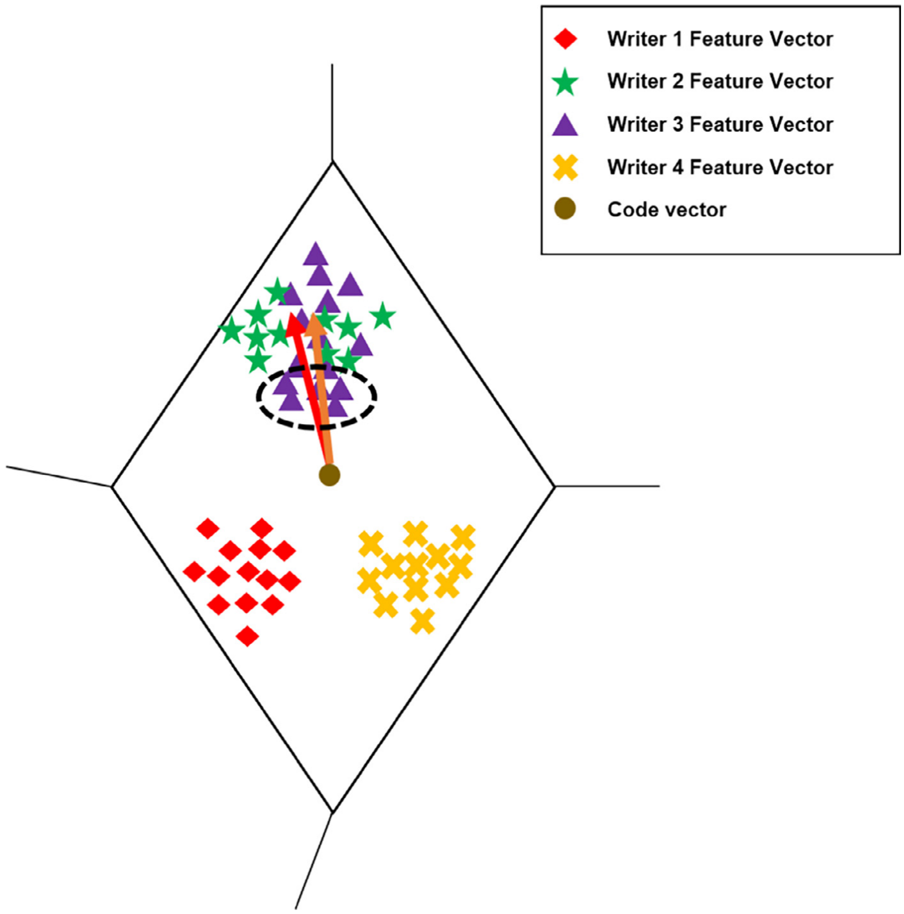

Fig. 3. Visual diagram presenting the drawback of the VLAD , while at the same time depicting the idea of our proposed strategy. For more details, please refer to the textual content in Section 6. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

trary to this, in our proposed descriptor, we separately score each of the distortion components by taking into regard their algebraic sign and thereafter perform the accumulations (as inferred from Eqs. (3)–(5)).

We demonstrate a visual illustration of a Voronoi cell in Fig. 3, with two dimensional feature vectors from four writers (denoted by different symbols). With the VLAD framework, we see that the normalized sum of residuals of feature vectors of writers 2 and 3 (marked using red and orange arrows) are nearby by, possibly leading to a reduced separation. However, on visual inspection, we do note that the feature vectors of writer 3 (highlighted by the area in the black ellipse) are more proximal to the representative two dimensional codevector of the cell (denoted by a brown circle) as compared to those of writer 2. We capture the essence of proximity in our proposal by providing a higher score the attributes of to such vectors - thereby bringing out better discrimination of these two writers.

7. Experimental framework

In this section, we present an outline of the four databases used, together with their enrollment and evaluation protocols adopted in our experiments.

7.1. Database description

The IAM Online Handwriting Database (IAM-OnDB) comprise handwritten samples acquired on a white-board, that are provided by 217 writers [41]. Each writer contributed eight paragraphs of 50 words each, with the (x, y) spatial coordinates and time stamp of the pen trace being recorded.

The IBM UB 1 Handwriting dataset [42] is contributed by 43 writers, with handwritten documents being characterized by summary and query text pages. The summary text comprises one or two pages of writing on a particular topic. Likewise, the query text contains an encapsulation of the summary text using approximately 25 words. The information of the trajectory captured by the pen include the spatial x-coordinate and y-coordinate.

The other two datasets, referred to as CASIA DS1 and CASIA DS2 in this work, consists of Chinese and English texts provided by the Institute of Automation, Chinese Academy of Sciences [43]. With regards to details of CASIA DS1, the handwritten data is taken over two sessions in which 187 writers wrote three Chinese pages in the first session. In the second session, a subset of the writers wrote an extra page of content. Keeping in line with the protocol of Yang et al. [32,33], we consider, for our experiments. online handwritten data from the first session. The CASIA DS2 comprises online handwritten samples from 134 writers with each having contributed to three pages of English text. The recorded data in both the databases of CASIA contain x-coordinate, y-coordinate, time stamp, button status, pressure, inclination and azimuth collected with a Wacom Intuos2 tablet.

7.2. Enrolment and evaluation protocol

For each writer of the IAM-OnDB, four documents are chosen at random for enrolment, with the remaining being reserved for evaluation. Likewise, for the IBM UB 1 handwriting data-set, we are using the protocol as suggested in [19] with 80% of the summary document paragraphs selected at random from each writer for enrolment and the rest for testing (including the query pages). For the CASIA Chinese and English datasets DS1 and DS2, two random pages of text per user are selected for enrolment – so that the performance of our algorithm can be evaluated on the remaining one page [32,33]. For each database, the codebook is generated by applying the k-means algorithm on the enrolled paragraphs / pages from 25% of the number of users randomly chosen.

The writer identification results of this paper are reported at paragraph and text line level. In the former, a codebook descriptor is derived separately for each paragraph / page data. Likewise, for the text line level, the paragraphs are divided into the individual text lines and a descriptor is obtained for each of them.

The enrolment of randomly selected documents, codebook generation, derivation of the descriptor and the identification strategy, referred to as a trial is performed ten times for the IAM-OnDB and IBM-UBM1 database. Likewise, for the CASIA DS1 and DS2, we consider three trials. The average writer identification rate (obtained over the trials) is reported.

8. Performance evaluation

In this section, we demonstrate results of experiments that present the utility of our proposal in identifying the authorship of documents.

8.1. Influence of codebook size

In this subsection, we show the trends of the VLAD and our proposed framework with regards to writer identification rates for varying sizes of codebook k. The average writer identification rates for the two techniques is depicted in Fig. 4 (a)–(h) for codebook sizes ranging from 5 to 100 in steps of five. Note that the blue and red curves denote the performance of VLAD and our proposal, respectively. The sub-figures in the first column (a), (c), (e) and (g) represent the paragraph level identification rates for the IAM-OnDB, IBM-UB1, CASIA DS1 and CASIA DS2, respectively. Likewise, the text line level performance for these databases are plotted on the remaining sub-figures of (b), (d), (f) and (h). For simplicity, we abbreviate the average writer identification rate as IR.

Owing to the fact that the above plots are aimed at demonstrating, in general the trend with different values of k, we employ the gap parameter $r = 1$ to derive the features for obtaining the codebook descriptors. However, at the same time, it is to be mentioned that the gap parameter value applied on the feature set does have an influence on improving the writer identification rates, as will be demonstrated in the following subsection.

The inferences from the different plots of Fig. 4 across the databases are as follows:

1. For the varying codebook sizes being considered, our descriptor provides a better performance than VLAD. This is owing to its ability to assign scores based on the residual / distortion values, thereby increasing the discriminativeness of the feature vectors of the writers.
2. Taking into consideration, the trend of the performance of VLAD and the proposed descriptor, we see that the average identification rate initially increases with codebook size and becomes comparable at moderate sizes.
3. Owing to coarser quantization, small-sized codebooks do not adequately capture the nuances that discriminate the features of different writers and hence the low average identification rate.

For sake of completion, we provide in Fig. 5, a snapshot of two documents from the IAM-OnDB that were wrongly identified using the VLAD framework but rectified by our proposal in an experimental trial.

Table 1
Average writer identification rates (in %) obtained with our codebook descriptor proposal for different values of the gap parameter. The best average writer identification rates obtained for each of the four databases is highlighted in *bold*.

(a) Paragraph level

Gap	IAM-OnDB		IBM UB1		CASIA DS1		CASIA DS2	
parameter (r)	IR	k	IR	k	IR	k	IR	k
1	97.64	60	92.72	60	88.61	65	92.04	60
2	98.28	60	94.13	65	89.82	65	93.28	60
3	**98.82**	60	**96.10**	60	**91.42**	65	95.03	65
4	98.61	60	95.62	60	90.28	70	**95.68**	60
5	98.03	60	94.44	60	88.80	70	94.40	60
6	97.80	60	93.67	65	86.32	70	92.77	60

(b) Textline level

Gap	IAM-OnDB		IBM UB1		CASIA DS1		CASIA DS2	
parameter (r)	IR	k	IR	k	IR	k	IR	k
1	87.78	70	77.45	65	78.16	70	80.91	70
2	88.52	65	79.45	70	79.54	65	82.55	70
3	**89.92**	70	**81.59**	70	**82.22**	70	84.45	75
4	89.06	65	81.32	70	80.17	70	**85.30**	70
5	87.90	65	80.79	65	76.54	65	83.37	70
6	87.14	70	78.56	70	73.37	70	81.08	70

8.2. Influence of the gap parameter

Recall from Section 4 that the features used for generating the codebook and the corresponding descriptor for an input handwritten document is dependent on the gap parameter value r. As a next experiment, we empirically show its influence on the variation of writer identification rates at both paragraph and text-line respectively by evaluating on the proposal discussed in Section 5. Table 1 (a) and (b) depicts the result of the same for values of r ranging from one to six in steps of one. The entries in the Table report the best average identification rates together with the size of the codebook employed to achieve it.

Clearly, across all the databases, a higher value of r is found to be more apt in providing a higher identification rate when compared to $r = 1$. The highest average identification rates with our proposal (marked in *bold*) is obtained with $r = 3$ for the IAM-OnDB, IBM UB1 and CASIA DS1 and $r = 4$ for the CASIA DS2. However with increasing values of r, the performance of the writer identification begins to drop – as can be noted by the entries for $r = 6$.

Furthermore, for the implementation of the descriptors in Sections 8.3 and 8.4 across the databases, we did analyze the influence of the gap parameter values on the writer identification performance. Without loss of generality, it was empirically observed that the value of the gap parameter to be chosen for the best average identification rate are the same as those specified in the aforementioned paragraph for the databases.

8.3. Comparison to variants of VLAD

The experiments outlined so far considered the original VLAD framework described in [35] for writer identification. Nevertheless, there have been variants of this and hence, for sake of completion, we compare our proposed scheme with them. In Table 2 (a) and (b), we outline the results of this empirical study at paragraph and text-line level.

For sake of convenience, we have abbreviated each of the variants as follows:

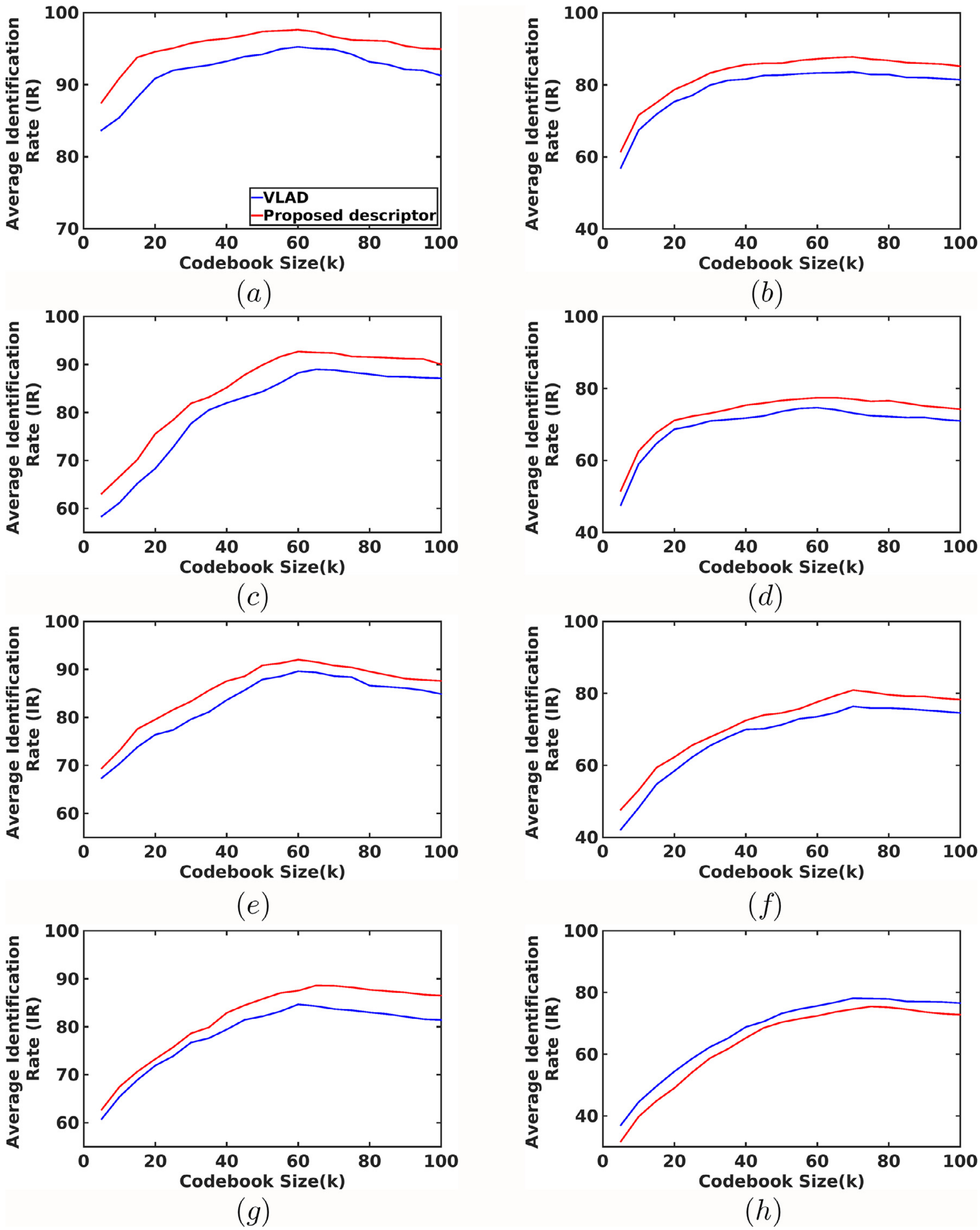

Fig. 4. Trend of the average writer identification rate obtained for VLAD (shown in blue) and the proposed codebook descriptor (in red) for varying size of codebook k. The sub-figures (a), (c), (e) and (g) represent the performance at the paragraph level for the IAM-OnDB, IBM-UB1, CASIA DS1 and CASIA DS2 respectively, while those of (b), (d), (f) and (h) denote at the text-line level. From these eight plots, it may be inferred that the proposal outperforms over VLAD at both paragraph and text-line levels. Note that for this experiment, we use the gap parameter $r = 1$. The influence of the gap parameter value in improving the writer identification performance of the codebook descriptor proposal is presented in the Section 8.2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- `VLAD+` L_2`-Norm` [34]: This is the version of the VLAD discussed in Section 6. It comprises accumulating the D residual vectors across the k codevectors of the codebook – thus leading to a $D \times k$ descriptor $\mathbf{S}$, which is then L_2 normalized.
- `VLAD+ Power-law` [35]: Each of the individual aggregated distortion $S_i(d)$ in a residual vector $\mathbf{S}_i$ are transformed with the function $sign(S_i(d)) \times |S_i(d)|^{\alpha}$, where $0 \le \alpha \le 1$. Thereafter, they are accumulated across the k codevectors, followed by a L_2 norm. Our experiments were run for different combinations of $(\alpha,\ k)$ and thereafter the best average identification results reported in Table 2.
- `VLAD+ innorm` [36]: This version is similar to `VLAD+` L_2`-Norm` scheme, except that L_2 norm is performed for each of the descriptors $\mathbf{S}_i$.

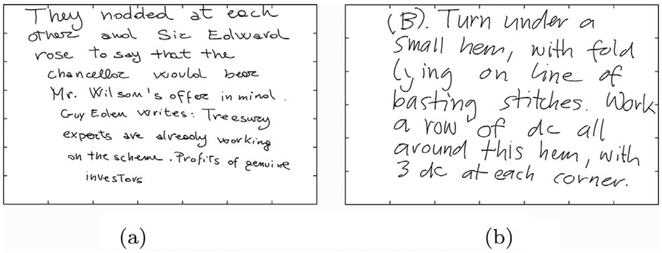

Fig. 5. Illustration of two documents from the IAM-OnDB that were wrongly identified using the VLAD framework but rectified by our proposal.

Table 2
Summary of the best average writer identification rate (in %) with different variants of VLAD and our proposal for the four databases. The size of the codebook k employed in each case is indicated.

(a) Paragraph level

Descriptor	IAM-OnDB		IBM UB1		CASIA DS1		CASIA DS2	
type	IR	k	IR	k	IR	k	IR	k
VLAD+L_2-Norm [34]	97.31	55	92.76	65	88.29	55	93.21	65
VLAD+Power law [35]	98.03	50	94.97	60	89.41	55	94.57	60
VLAD +innorm [36]	97.73	50	93.85	60	88.95	60	93.76	55
VLAD+SSR [36]	97.96	55	94.36	55	88.53	55	94.15	60
VLAD+RN [37]	97.72	50	93.71	55	88.82	60	93.65	65
Proposed descriptor	**98.82**	60	**96.10**	60	**91.42**	65	**95.68**	60

(b) Text line level

Descriptor	IAM-OnDB		IBM UB1		CASIA DS1		CASIA DS2	
type	IR	k	IR	k	IR	k	IR	k
VLAD+L_2-Norm [34]	87.46	60	78.16	60	79.06	65	82.72	60
VLAD+Power law [35]	88.34	55	79.84	65	81.10	60	84.15	65
VLAD +innorm [36]	88.09	55	79.06	65	80.29	65	83.69	65
VLAD+SSR [36]	88.23	60	79.56	70	80.82	70	83.92	70
VLAD+RN [37]	88.04	60	78.90	65	80.12	65	83.44	65
Proposed descriptor	**89.92**	70	**81.59**	70	**82.22**	70	**85.30**	70

- `VLAD+ SSR` [36]: Each of the aggregated distortion $S_i(d)$ in $\mathbf{S}_i$ are transformed with the function $sign(S_i(d)) \times \sqrt{|S_i(d)|}$. Thereafter, they are accumulated across all k codevectors and normalized with L_2.
- `VLAD+ RN` [37]: This version is similar to `VLAD+` L_2`-Norm`, except that L_2 normalization is performed separately for each residual vector $\mathbf{f}_i^p - \boldsymbol{\mu}_i$, prior to accumulation.

With regards to the performances, our proposed descriptor does better across all the four databases. The reason for the improvement is owing to the fact that it deviates from the VLAD variants by considering the proximity criterion for scoring rather than relying on the residual distortion.

8.4. Empirical study with a reduced version of our descriptor

Despite the proposed codebook descriptor outperforming the VLAD (discussed in Section 6) at both paragraph and text line levels, it comes at the cost of increased feature vector dimension. Note that for a codebook size k, the dimension of VLAD and our descriptor is $D \times k$ and $2 \times D \times (k-1)$, respectively. As a circumvention to this issue, we attempt at judging the writer identification performance by utilizing the codebook descriptor with a lower dimension as follows:

We consider only the contributions from the $\tilde{S}_i^+(d)$ scores of each of the D attributes in the $k-1$ codevectors $\{\boldsymbol{\mu_i}\}_{i=1}^{k-1}$.[4] The performance of the resulting $D \times (k-1)$ codebook descriptor was then evaluated for varying sizes of codebook. However, we present only the best average identification rates obtained along with the corresponding codebook size in Table 3 (a) and (b) across all the databases for the paragraph and text-line level, respectively.

The reduced dimension variant of our proposal works better than the VLAD. Likewise, when comparing to the full dimension $2 \times D \times (k-1)$ descriptor, we see that it trails behind in performance by up-to around 1%. Nevertheless, this trend may be acceptable, owing to the fact that the reduced dimension descriptor does not contain the complete feature description described in Section 5 for the handwritten paragraph / text under consideration.

Even though comparable identification rates are reported with large-size codebooks, the $D \times (k-1)$ descriptor gets outperformed by the one with $2 \times D \times (k-1)$ dimension at smaller sizes. To validate further, we present the performance trend for varying number of codebook sizes (less than 20) at the paragraph level (Table 4) for the IAM-OnDB. We see that for the codebook comprising five codevectors, the proposed $D \times (k-1)$ dimension descriptor performs worse than the VLAD. This degradation in performance at small codebook sizes is largely attributed to the omission of the $\tilde{S}_i^-(d)$ scores during the construction of the descriptor. A small codebook size implies relatively larger Voronoi cells and the process of this omission leads to a prodigious loss of information thereby lowering the identification rate. However, this problem of the reduced $D \times (k-1)$ descriptor is alleviated at higher codebook sizes (values of k greater than 20) due to the relatively finer Voronoi regions that help in a better discrimination between writers.

To summarize the preceding discussion, either of the proposed descriptors of dimension $D \times (k-1)$ or $2 \times D \times (k-1)$ would suffice in providing a comparable performance at large size codebooks. However for small codebook sizes, the higher dimensional descriptor does better in discrimination as can be inferred from the entries in Table 4. We wish to state that though the discussion was presented for the IAM-OnDB, we nonetheless see a similar trend across the other three databases as well - with either of the descriptors as a choice at higher sizes of codebook.

8.5. Evaluation on a multi-language writer identification set-up

In this subsection, we evaluate the performance of our descriptor proposal (of Section 5) for a multi-language writer identification set-up. For this, we have used the third dataset from CASIA [43] that comprises online handwritten data corresponding to 133 writers in two languages – namely English and Chinese. For simplicity, we abbreviate this dataset as CASIA DS3. Each of the writers contributed to three pages of each language. For this experiment, we randomly consider four pages consisting of two English and two Chinese handwritten contents for enrolment. The remaining pages are reserved for performance evaluation. Likewise, for generating the codebook, we consider the enrolled documents corresponding to 25% of the writers that are randomly chosen. The resultant codebook is then employed for generating the writer descriptors for documents written in both languages.

Based on the above mentioned set up, a comparison of our proposed descriptor with that of the VLAD was performed at both document / paragraph and text line level. The best average writer identification rates (over a set of three trials) are depicted in Table 5 with the size of the codebook and the gap parameter value being specified. It can be noted from the entries of the Table that our proposal achieves a higher average identification rate with an improvement of 5.18% and 11.36% over the VLAD descriptor at document and text line level, respectively.

[4] Note that, we could have as well considered only the scores $\tilde{S}_i^-(d)$ for making the $D \times (k-1)$ dimensional codebook descriptor.

Table 3
Summary of the average writer identification rates (in %) for the best performing codebook size corresponding to the different descriptors. Note, that for our work, $D = 7$, as mentioned in Section 4.

(a) Paragraph level

Descriptor	Dimension	IAM-OnDB		IBM UB1		CASIA DS1		CASIA DS2	
		IR	k	IR	k	IR	k	IR	k
VLAD	$D \times k$	97.31	55	92.76	65	88.29	55	93.21	65
Proposed	$2 \times D \times (k-1)$	98.82	60	96.10	60	91.42	65	95.68	60
Proposed	$D \times (k-1)$	98.47	65	95.12	60	90.32	65	94.98	65

(b) Text line level

Descriptor	Dimension	IAM-OnDB		IBM UB1		CASIA DS1		CASIA DS2	
		IR	k	IR	k	IR	k	IR	k
VLAD	$D \times k$	87.46	60	78.16	60	79.06	65	82.72	60
Proposed	$2 \times D \times (k-1)$	89.92	70	81.59	70	82.22	70	85.30	70
Proposed	$D \times (k-1)$	89.10	70	80.57	65	81.79	60	84.69	70

Table 4
Average paragraph level writer identification rates (in %) on the IAM-OnDB obtained for varying codebook sizes with the different descriptors.

Codebook size (k)	VLAD ($D \times k$ dimension)	Proposed descriptor ($2 \times D \times (k-1)$ dimension)	Proposed descriptor ($D \times (k-1)$ dimension)
5	88.96	90.62	84.85
10	92.03	94.12	92.46
15	93.85	95.97	94.31
20	94.93	96.66	95.53

Table 5
Performance comparison of the proposed writer identification system to the VLAD descriptor at document and text line level for a multi-language scenario. The third dataset from CASIA (abbreviated as CASIA DS3) is used for this experiment.

Descriptor	Document level		Text line level	
Name	IR	(k, r)	IR	(k, r)
VLAD	82.45	(65,3)	66.82	(70,3)
Proposed	86.72	(80,3)	74.41	(85,3)

9. Comparison to previous studies

In this section, we provide an indication on how our proposed writer identification method fares with regards to previous works of the databases being experimented. While discussing on these, we admit that a direct one to one comparison is not always possible. This is owing to the use of different features, classifier training methodologies and enrolment strategies.

9.1. IAM-OnDB

To the best of our knowledge, the writer identification systems proposed in [12,19,21,23,39] utilize the IAM-OnDB database for experimentation. In the GMM based framework of Schlapbach et al. [12], the authors report an average writer identification rate of 98.56 and 88.96% for paragraph and text line levels. Contrast to this, the proposed methodology with the average writer identification rate of 98.82% at paragraph level indicate that we are on par (albeit with a slightly higher accuracy). It is also worth noting here that an average performance of 89.92% is observed at the text line – higher than 88.96% in the GMM based system. Said in another way, the obtained results are quite competitive.

Likewise, in the work based on the concept of Latent Dirichlet Allocation in [19], a writer identification rate of 93.39% at paragraph level has been cited – which is lower to that of our methodology. Thirdly, with regards to the subtractive clustering based approach presented in [21], an average writer identification rate of 96.3% was reported at paragraph level. The sparse framework representation employed in the work [23] achieved a paragraph level identification rate of 98.94% and a text line level accuracy of 83.30%.

Lastly, when compared to our work published in [39], the present proposal achieves at paragraph an increased accuracy of 1.01% when compared to 97.81%. However, we obtain a high improvement of 11.55% at the text line level – from 80.61 to 89.92%. The reason for the higher writer identification rates is owing to the formulation of the codebook descriptor by considering the scores from the feature vectors of the entire document (refer Eq. (4)). Contrary to this, in the approach of Venugopal and Sundaram [39], the contribution of the scores for the generation of the descriptor of a given codevector was limited to only a subset of features assigned to it. Another factor to the improved performance is the incorporation of the gap parameter in the extraction of the feature vectors used for codebook generation.

9.2. IBM UB 1 database

Two explorations have been made till date on this database [19,39]. In the former, a writer identification rate of 89.47% at paragraph level is obtained with the system being based on Latent Dirichlet Allocation. The second pertains to the preliminary exploration of our codebook descriptor with regression based feature normalization scheme - wherein the performance reported is that of 94.37% at paragraph level. In comparison to these approaches, our present proposal, owing to the same reasoning given for the IAM-OnDB gives a performance improvement of 1.83% – with an accuracy of 96.10% at the paragraph level. At the text-line level, we see that compared to our previous work in [39], the best average writer identification rate increases from 76.86 to 81.59%.

9.3. CASIA DS1 and DS2

The studies of online writer identification utilizing the CASIA databases for performance evaluation are those of [32,33]. These works are based primarily on the exploration of the CNN and RNN based deep learning approach on these databases, leading to higher identification rates for DS1 and DS2, respectively. The improvements are attributed to the capability of such a network to learn feature representations from data as opposed to hand crafted features used in our proposal.

Apart from the above, the authors of [32] also mention Top 1 writer identification results for the GMM-UBM framework of Schlapbach et al. [12] and the textural and allographic features based system of Bulacu and Schomaker [44]. In particular, the former provided a document level accuracy of 80 and 82% on DS1 and DS2, while the latter – an offline image based approach attained a writer identification rate of 84 and 85%. Clearly, we see that our method outperforms the results of these methods on the datasets DS1 and DS2.

Hence to conclude, amongst all the methods proposed in the literature using hand crafted features in Sections 9.1, 9.2 and 9.3, it can be observed that our proposal achieves a commensurate performance if not better on all the databases being experimented in this paper. In this regard, we conclude with the remark that our method is quite promising for identifying the authorship of online handwritten text documents.

10. Conclusion

In this paper, we improved on the preliminary attempt being made on codebook descriptors in the work [39]. Given a codebook of size k, we consider the descriptors of only $k-1$ codevectors to construct the final descriptor by concatenation. In addition, for constructing the codebook and subsequently the descriptor, we extract point-based features by incorporating a so called gap parameter. The experiments performed on a number of databases illustrate the promising nature of our approach to previous explorations of online writer identification.

In the future, we plan to utilize a sparse framework to learn the dictionary / codebook and then to thereby modify our proposal to construct the descriptor.

References

[1] A.K. Jain, A. Ross, S. Prabhakar, An introduction to biometric recognition, IEEE Trans. Circuits Syst. Video Technol. 14 (1) (2004) 4–20.
[2] I. Siddiqi, N. Vincent, Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features, Pattern Recognit. 43 (11) (2010) 3853–3865.
[3] D. Bertolini, L.S. Oliveira, E. Justino, R. Sabourin, Texture-based descriptors for writer identification and verification, Expert Syst. Appl. 40 (6) (2013) 2069–2080.
[4] A. Nicolaou, A.D. Bagdanov, M. Liwicki, D. Karatzas, Sparse radial sampling LBP for writer identification, in: Proceedings of the Thirteenth International Conference on Document Analysis and Recognition (ICDAR 2015), 2015, pp. 716–720.
[5] V. Christlein, D. Bernecker, F. Hönig, A. Maier, E. Angelopoulou, Writer identification using GMM supervectors and exemplar-SVMs, Pattern Recognit. 63 (2017) 258–267.
[6] A. Namboodiri, S. Gupta, Text independent writer identification from online handwriting, in: Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition, 2006.
[7] Kholmatov, B.A. Yanikoglu, Identity identification using improved online signature identification method, Pattern Recognit. Lett. 26 (15) (2005) 2400–2408.
[8] J.F. Vargas, M.A. Ferrer, C. Travieso, J.B. Alonso, Off-line signature verification based on grey level information using texture features, Pattern Recognit. 44 (2) (2011) 375–385.
[9] M. Parodi, J.C. Gómez, Legendre polynomials based feature extraction for online signature verification. consistency analysis of feature combinations, Pattern Recognit. 47 (1) (2014) 128–140.
[10] M. Liwicki, A. Schlapbach, H. Bunke, S. Bengio, J. Mariéthoz, J. Richiardi, Writer identification for smart meeting room systems, in: Document Analysis Systems VII, Springer, 2006, pp. 186–195.
[11] M. Liwicki, A. Schlapbach, H. Bunke, Fusing asynchronous feature streams for on-line writer identification, in: Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 2007, pp. 103–107.
[12] A. Schlapbach, M. Liwicki, H. Bunke, A writer identification system for on-line whiteboard data, Pattern Recognit. 41 (7) (2008) 2381–2397.
[13] D.A. Reynolds, Speaker identification and verification using gaussian mixture speaker models, Speech Commun. 17 (1) (1995) 91–108.
[14] S.K. Chan, C. Viard-Gaudin, Y.H. Tay, Online writer identification using character prototypes distributions, Proceedings of the Sixth International Conference on Information, Communications & Signal Processing, 2007, pp. 1–5.
[15] G.X. Tan, C. Viard-Gaudin, A.C. Kot, Automatic writer identification framework for online handwritten documents using character prototypes, Pattern Recognit. 42 (12) (2009) 3313–3323.
[16] G. Tan, C. Viard-Gaudin, A. Kot, Online writer identification using fuzzy c-means clustering of character prototypes, in: Proceedings of the International Conference on Frontiers in Handwriting Recognition, 2008, pp. 475–480.
[17] G.X. Tan, C. Viard-Gaudin, A.C. Kot, A stochastic nearest neighbor character prototype approach for online writer identification, in: Proceedings of the Nineteenth International Conference on Pattern Recognition, 2008, pp. 1–4.
[18] A. Shivram, C. Ramaiah, U. Porwal, V. Govindaraju, Modeling writing styles for online writer identification: a hierarchical Bayesian approach, in: Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR), 2012, pp. 387–392.
[19] A. Shivram, C. Ramaiah, V. Govindaraju, A hierarchical Bayesian approach to online writer identification, IET Biom. 2 (4) (2013) 191–198.
[20] C. Ramaiah, A. Shivram, V. Govindaraju, Data sufficiency for online writer identification: a comparative study of writer-style space vs. feature space models, in: Proceedings of the International Conference on Pattern Recognition (ICPR), 2014, pp. 3121–3125.
[21] G. Singh, S. Sundaram, A subtractive clustering scheme for text-independent online writer identification, in: Proceedings of the Thirteenth International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 311–315.
[22] A. Chaabouni, H. Boubaker, M. Kherallah, A.M. Alimi, H. El Abed, Multi-fractal modeling for on-line text-independent writer identification, in: Proceedings of the International Conference on Document Analysis and Recognition, 2011, pp. 623–627.
[23] I. Dwivedi, S. Gupta, V. Venugopal, S. Sundaram, Online writer identification using sparse coding and histogram based descriptors, in: Proceedings of the Fifteenth International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 572–577.
[24] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, IEEE, 2005, pp. 886–893.
[25] B. Li, T. Tan, Online text-independent writer identification based on temporal sequence and shape codes., in: Proceedings of the Tenth International Conference on Document Analysis and Recognition, IEEE, 2009, pp. 931–935.
[26] Z. Sun, B. Li, T. Tan, Online text-independent writer identification based on Stroke's probability distribution function, in: Proceedings of the International Conference on Advances in Biometrics, 4642, 2007, pp. 201–210.
[27] B. Li, Z. Sun, T. Tan, Hierarchical shape primitive features for online text-independent writer identification, in: Proceedings of the Tenth International Conference on Document Analysis and Recognition, IEEE, 2009, pp. 986–990.
[28] M. Gargouri, S. Kanoun, J.-M. Ogier, Text-independent writer identification on online Arabic handwriting, in: Proceedings of the Twelfth International Conference on Document Analysis and Recognition (ICDAR), 2013, pp. 428–432.
[29] T. Dhieb, W. Ouarda, H. Boubaker, M.B. Halima, A.M. Alimi, Online Arabic writer identification based on beta-elliptic model, in: Proceedings of the International Conference on Intelligent Systems Design and Application (ISDA), 2015, pp. 74–79.
[30] T. Dhieb, W. Ouarda, H. Boubaker, A.M. Alimi, Beta-elliptic model for online writer identification from online arabic handwriting, J. Inf. Assur. Secur. 11 (5) (2016) 263–272.
[31] T. Dhieb, W. Ouarda, H. Boubaker, A.M. Alimi, Deep neural network for online writer identification using beta-elliptic model, in: Proceedings of the International Joint Conference on Neural Networks, 2016, pp. 1863–1870.
[32] W. Yang, L. Jin, M. Liu, DeepWriterID: an end-to-end online text-Independent writer identification system, IEEE Intell. Syst. 31 (2) (2016) 45–53.
[33] X.Y. Zhang, G.S. Se, C.L. Liu, Y. Bengio, End-to-end online writer identification with recurrent neural network, IEEE Trans. Hum. Mach. Syst. 47 (2) (2017) 285–292.
[34] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3304–3311.
[35] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1704–1716.
[36] R. Arandjelović, A. Zisserman, All about VLAD, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.
[37] J. Delhumeau, P.-H. Gosselin, H. Jégou, P. Pérez, Revisiting the VLAD image representation, in: Proceedings of the Twenty-First ACM International Conference on Multimedia, 2013, pp. 653–656.
[38] Z. Liu, S. Wang, Q. Tian, Fine-residual VLAD for image retrieval, Neurocomputing 173, Part 3 (2016) 1183–1191.
[39] V. Venugopal, S. Sundaram, An online writer identification system using regression-based feature normalization and codebook descriptors, Expert Syst. Appl. 72 (2017) 196–206.

[40] K. Barkoula, G. Economou, S. Fotopoulos, Online signature verification based on signatures turning angle representation using longest common subsequence matching, Int. J. Doc. Anal. Recognit. (IJDAR) 16 (3) (2013) 261–272.
[41] M. Liwicki, H. Bunke, IAM-OnDB-an on-line English sentence database acquired from handwritten text on a whiteboard, in: Proceedings of the Eighth International Conference on Document Analysis and Recognition, 2005, pp. 956–961.
[42] A. Shivram, C. Ramaiah, S. Setlur, V. Govindaraju, IBM_UB_1: a dual mode unconstrained English handwriting dataset, in: Proceedings of the Twelfth International Conference on Document Analysis and Recognition (ICDAR), 2013, pp. 13–17.
[43] CASIA Handwriting Database – Biometrics Ideal Test, The Institute of Automation, Chinese Academy of Sciences (CASIA), 2007. http://biometrics.idealtest.org/.
[44] M. Bulacu, L. Schomaker, Text-independent writer identification and verification using textural and allographic features., IEEE Trans. Pattern Anal. Mach. Intell. 29 (4) (2007) 701–717.

Vivek Venugopal received the Bachelors degree in Electronics and Communication Engineering from Amrita School of Engineering, Coimbatore, India in 2011 and M.Tech degree in Electronics and Communication Engineering from National Institute of Technology Jalandhar, India, in 2014. He is currently working towards his Ph.D. degree in the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, India. His research interests include online writer identification, online signature verification, image processing, and pattern recognition.

Suresh Sundaram received his Doctoral degree from the Indian Institute of Science, Bangalore, India in March 2012. Between 2012 September and July 2013, he was a research consultant at Hewlett Packard Research Labs, Bangalore, India. He is currently an Assistant Professor in the department of Electronics and Electrical Engineering at Indian Institute of Technology Guwahati, India. His research interests encompass the areas of handwriting recognition, biometrics, document analysis and pattern recognition. He served as a Program Committee member for the ICDAR 2017. He is also an active reviewer for the journal of Pattern Recognition Letters.