Engineering Applications of Artificial Intelligence 95 (2020) 103912

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A deep learning framework for text-independent writer identification

Malihe Javidi, Mahdi Jampour *

Quchan University of Technology, Quchan, Iran

ARTICLE INFO

Keywords:
Enhanced ResNet
Deep residual networks
Text-independent
Handwriting writer recognition

ABSTRACT

Handwriting Writer Identification (HWI) refers to the process of handwriting text image analysis to identify the authorship of the documents. It has yielded promising results in various applications, including digital forensics, criminal purposes, exploring the writer of historical documents, etc. The complexity of the text image, especially in images with various handwriting makes the writer identification difficult. In this work, we propose an end-to-end system that relies on a straightforward yet well-designed deep network and very efficient feature extraction, emphasizing feature engineering. Our system is an extended version of ResNet by conjugating deep residual networks and a new traditional yet high-quality handwriting descriptor towards handwriting analysis. Our descriptor analyzes the handwriting thickness as a preliminary and essential feature for human handwriting characteristics. Our approach can also provide text-independent writer identification that we do not need to have the same handwriting content for learning our model. The proposed approach is evaluated and achieved consistent results on four public and well-known datasets of IAM, Firemaker, CVL, and CERUG-EN. We empirically demonstrate that our conjugated network outperforms the original ResNet, and it can work well for real-world applications in which patches with few letters exist.

1. Introduction

Handwriting is an individual human characteristic that represents the writer's psychological state during the writing and can prove a person's authenticity through its pattern analysis. It has many potential applications such as digital forensics, criminal purposes, exploring the writer of historical documents, etc. Handwriting writer identification refers to finding an author from his/her handwriting documents among numerous writers' documents. This identifying is possible due to the handwriting style, which has multiple specific features that show the author's personality. The features are including the shape of letters, spacing between letters, the slope of letters, cursive or separated writing, rhythmic repetition of the elements, the pressure to the paper, the size of letters, the thickness of letters, etc. (Khan et al., 2019). Since a human can intelligently discover a bunch of above features, modern AI systems focus on proposing approaches that can imitatively identify human from handwriting images that is an offline handwriting writer identification system. In contrast, an online writer identification system analyzes a script during the writing. Therefore, it needs expensive peripheral devices like a digital pen to record vital information such as writing speed, chaotic behavior, etc. Of course, more information helps to achieve a better identification rate.

Handwriting writer identification has numerous employment. While handwriting analysis is useful for guessing the author of the historic or non-historic documents based on the recognized scripts (De Stefano et al., 2018), it is also pretty beneficial for forensics purposes like writer identification or verification due to carry the personality of the authors. Handwriting writer identification can also be divided into two branches of text-depended and text-independent categories. Both approaches find the most similar handwriting (author label) from a pre-defined gallery compared to the input test handwriting sample. Nevertheless, the former refers to the methods that use the same text for discovering author properties, while the latter is not sensitive to the analyzed text content (Xiong et al., 2017). This means that the content of the test and train texts must not be needed to be the same for all authors. Therefore, only the second category can be used for applications like exploring writers of historical documents, although both are useful for forensic purposes. In contrast to the identification, writer verification refers to confirming the similarity of two scripts (items). In handwriting analysis, while we know who the author of a document is, we need to verify whether the claimed person is correct or not? Similar to other verification biometrics like signature verification as well as fingerprint verification. Of course, the identification is more complicated than verification because the authentication system needs to find the similarity of input sample from thousands of examples in the gallery, which means that the identification is a process of finding one from N while verification is confirming 1 by 1.

Although, several approaches achieved excellent performance on different writer identification datasets; they are still far to address this problem in more challenging datasets or real-world applications.

* Corresponding author.
E-mail address: jampour@qiet.ac.ir (M. Jampour).

https://doi.org/10.1016/j.engappai.2020.103912
Received 27 February 2020; Received in revised form 26 July 2020; Accepted 23 August 2020
Available online 28 August 2020
0952-1976/© 2020 Elsevier Ltd. All rights reserved.

For instance, challenges such as using different pens and colors are prevalent and still open, which are considered to address. Most of the current related studies focused on feature extraction and generally emphasized on the feature engineering using either new modern deep approaches (Fiel and Sablatnig, 2015; Xing and Qiao, 2016; He and Schomaker, 2019, 2020) or classical descriptors (Chahi et al., 2018, 2020)

To this end in this paper, we propose a deep residual network-based writer identification system, which is a text-independent approach and very efficient for discovering author characteristics from handwriting images. It is also known as offline handwriting writer identification. Therefore, in our first contribution, we propose efficient residual networks architecture to explore the deep features of handwriting in conjugation with classical feature descriptors. Our second contribution relies on proposing a novel effortless feature descriptor that can explain the handwriting thickness. We found that conjugation of deep and traditional feature descriptors (e.g., the gradient information or the handwriting thickness information) is useful for handwriting writer identification applications. Moreover, according to the problem, we introduce an efficient and helpful sampling method that uses part of a word or sentence in a fixed size of an image with a dimension of 60×180 pixels, including few letters (mostly lower than four letters). This shows our approach can efficiently identify the writer of a text from its image even with few letters that is very useful for real-world applications, especially for historical handwriting writer identification. Our model conceptual architecture can be seen in Fig. 1.

In the following sections, we first review the related works; then introduce our main deep residual network architecture for writer identification in Section 3; we then propose our handwriting thickness descriptor along with the deep feature exploration and their concatenation strategy in the same Section. In Section 4, we introduce four well-known datasets for handwriting writer identification and data sampling with extensive evaluation on all datasets. Finally, the discussion and conclusion are provided in Sections Sections 5 and 6.

2. Related work

Recent works on writer identification are almost focused on deep CNN-based approaches, which are successful techniques for feature extraction and description from the images. An early work regarding the writer identification using CNN proposed by Fiel and Sablatnig (2015). The authors generated a CNN-based feature vector for each writer in their proposed approach. They compared then the explored feature vectors by pre-calculated vectors stored in the dataset. The authors used the second last fully connected layer as feature vectors and the nearest neighbor method for identification purposes in their architecture. They showed a significant improvement in contrast to the classical approaches.

Deep writer (Xing and Qiao, 2016) presented another CNN-based approach that proposed a deep multi-stream CNN model for the handwriting representation. Authors designed and optimized a multi-stream structure by exploiting ordered patches and merged element-wise of them for identification that improved their basic idea, also outperformed previous related approaches on two IAM and HWDB datasets. An exciting part of this work is that the authors used patch-based information of each handwriting. They introduced a patch scanning model which can handle text images with different size. We inspired a similar patch scanning strategy for our handwriting patch extraction.

He and Schomaker have recently proposed two deep architectures (He and Schomaker, 2019, 2020). The former represents that the handwriting contains two implicit and explicit information where the explicit information refers to the length, number of characters, and lexical content of single words. In contrast, the implicit information is the writer's behavioral information that can be used for author identity. They proposed using both implicit and explicit information so that explicit information can be used along with the implicit features to make extra information. Their proposed CNN architecture is based on the AlexNet, including two parallel pathways. Nevertheless, a challenging issue with their work relies on resizing handwriting patches into $120 \times 140 \times 1$, where we believe such affine transformation can lose the writer's intrinsic information. Therefore, we do not resize the handwriting images and keep handwriting slant, thickness, and similar features. Instead, we crop parts of handwriting images so that our patches contain all handwriting samples' width. Our design helps to keep the writer's intrinsic characteristics during sampling. The latter work also proposed a deep model, namely FragNet (He and Schomaker, 2020) that is a two pathways deep architecture including feature pyramid and fragment pathway. By the first pathway, it accepts the whole word image as the input. In contrast, the second pathway accepts only the segmented fragments from the input image and feature maps. Inspired from this work, we provide a two pathway architecture that helps us for a more efficient representation of the text images.

Further works considered joint-based approaches. For instance, Tang and Wu (2016) proposed an offline text-independent writer identification using CNN and Bayesian. They first generated thousands of handwriting images using their augmentation technique since CNN needs a large number of training data. Extracting discriminative features through deep CNN is the next stage they used in their work. Their CNN-based approach contains ten layers, including four layers of convolution, four layers of pooling, and two fully connections layers. Their experiments, especially on the CVL dataset, show significant results.

Ni et al. (2017) focused on offline handwriting writer identification from noisy images. They showed that while CNN based approaches are efficient methods for feature representation but their joint approach using CNNs and traditional computer vision descriptors can efficiently improve applications like writer identification even from noisy images. The authors employed an existing clean dataset for training purposes and then added noises artificially to generate noisy data for evaluation. Their synthetic approach, including several parts like segmentation, denoiser, particular CNNs, and handcrafted features, outperformed the related works by about 10%.

In contrast to the modern approaches, Chahi et al. proposed classic approaches emphasizing extracting desirable features (Chahi et al., 2018, 2020). Authors in Chahi et al. (2018) proposed a Block Wise Local Binary Count (BW-LBC) operator inspired by traditional LBP that represents multiple histograms. The histograms created based on the occurrence distribution of the pixels in small blocks. They employed the well-known nearest-neighbor classification using the Hamming distance and showed that their approach is comparable or better than the modern approaches. In Chahi et al. (2020), the authors proposed another classical feature extraction method for writer identification. Their descriptor represents a salient feature for local writing structure and applies to small connected regions of the sample. These feature maps have been used as inputs for the nearest neighbor classifier to classify the query writer. These two works show that traditional features are still useful for writer identification and superior in some tasks.

The related work found that the CNN-based approaches are an excellent methodology for feature explanation, especially in handwriting writer identification. It has been widely shown that their feature extraction is very efficient. Nevertheless, He and Schomaker (2019) shows that not only the deep features are useful but also other meaningful features such as those features that they called explicit features (e.g., length, number of characters) can still help to improve the accuracy of the identification as well as (He and Schomaker, 2020). Moreover, Chahi et al. (2018, 2020) showed the superiority of the traditional feature descriptors. Motivated by the observations above, in this paper, we propose a deep convolutional neural network joined to traditional features. However, we do not resize the handwriting images because we believe it will lose the writers' personalities due to the stretching the image. Instead, we suggest to crop patches of the handwriting and propose a descriptor to explain each handwriting image

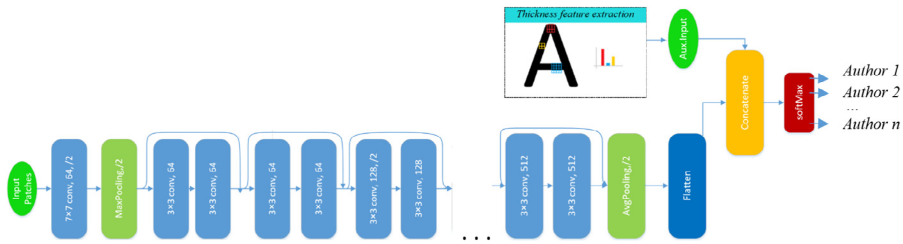

Fig. 1. Our model conceptual architecture containing both deep and traditional features.

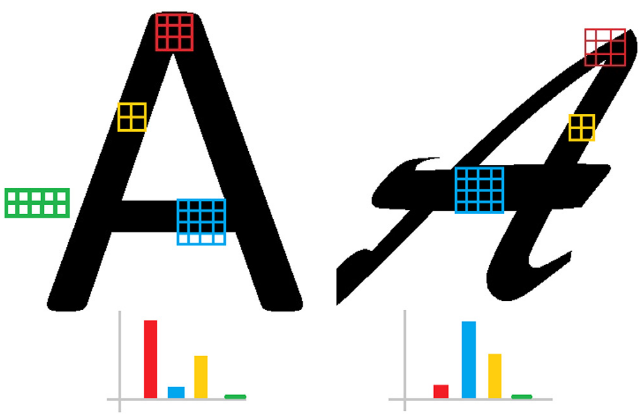

Fig. 2. Handwriting thickness descriptor (HTD), counts fully black patches with different size.

regarding the script thickness. Therefore, our model is a conjugating approach to fill the gap of using classical and modern features and achieves appropriate results.

3. Proposed methodology

This section proposes our methodology containing a conjugating architecture, where we exploit classical local descriptions along with the modern deep features extracted from the text images. The goal is to profit the usefulness of both deep and classic features. In the following, we describe more details.

3.1. Handwriting Thickness Descriptor (HTD)

As in Jampour and Naserasadi (2019) proposed, there are intrinsic features in signatures that are useful for handwriting analysis and particularly the identification. One of the most essential and obvious characteristics in handwriting is the script's thickness, which is one of the early features for handwriting discrimination by a human. It has been previously shown (Crespo et al., 2018) that the handwriting thickness is even an interpretable characteristic for human behavior analysis. In this section, we propose a spatial feature descriptor that can analyze the handwriting thickness. This characteristic strongly depends on humans' personality and, subsequently, the pressure of the pen on the paper. According to this characteristic, mostly, the thickness of two different handwriting is not exactly the same. Therefore, we propose to compute each writer's handwriting thickness and use it for better discrimination. To this end, we count the number of rectangular masks on the script with different size $w \times h$ (e.g., 2×2 or 3×5, etc.) using convolutional processing which represented in the following equations:

$$w, h \in \mathbb{N}, \tag{1}$$

$$2 \leq w \leq n, \quad w \leq h \leq n, \tag{2}$$

$$t_k = \mathrm{Card}\{(i,j) | I^k_{i,j} = 1\} = \sum_{i=1}^{M} \sum_{j=1}^{N} I^k_{i,j}, \tag{3}$$

$$F^k_{(w,h)} = \frac{\sum_{i=1}^{M-w+1} \sum_{j=1}^{N-h+1} I^k_{i,j} \odot W(w,h)}{t_k} \tag{4}$$

where I is an inversed binary handwriting image and W is a rectangular mask in size of $w \times h$.

Let $M_{w\times h}$ be the space of sub-matrices in size $w \times h$. We define operator $\odot$ as the following:

$$\odot : M_{w\times h} \times M_{w\times h} \rightarrow \{0, 1\} \tag{5}$$

where

$$A \odot B := \begin{cases} 1 & \text{if } \sum_{i=1}^{w} \sum_{j=1}^{h} A_{ij} \cdot B_{ij} = wh \\ 0 & \text{o.w.} \end{cases} \tag{6}$$

The operator $\odot$ returns 1 if the dot product between all elements of I and W are opposite to zero; otherwise, it returns 0. In other words, Eq. 6 says that the operator $\odot$ returns 1 if an arbitrary mask of W is exactly on handwriting pixels. Also, t is the sum of the handwriting pixels to normalize our counters to be independent of different textual content. The sum of such occurrence describes a handwriting thickness. In our work, we define $n = 10$ which means $2 \leq w \leq 10$, therefore, we have 45 (i.e. $(n(n-1))/2$) rectangular masks (W) that make a feature vector with 45 dimensions to describe the handwriting thickness. Fig. 2 shows an example of two different writing styles of the letter 'A' and their HTD descriptors schematic along with four types of rectangular masks.

3.2. The baseline network

In this work, we employed a deep residual network to enhance our idea towards writer identification. A residual neural network (ResNet) is a very successful DNNs architecture introduced by He et al. (2016b). The major problem of neural networks with deep layers is a degradation problem that occurs when the network depth increased, which leads to higher training error (He and Sun, 2015). Moreover, training deeper neural networks with many parameters requires high learning costs. To overcome this problem, He et al. (2016a) developed a modular architecture that stacks residual blocks with identity mapping that skipped one or more layers. It is shown that the residual learning framework which utilized skip connections can increase network layers to thousands while the performance is not reduced. In the residual block, the following computation is performed:

$$x_{l+1} = \sigma(x_l + \mathcal{F}(x_l \cdot W_l)) \tag{7}$$

where x_l and $x_{(l+1)}$ are the input and output of the lth residual block. W_l is a set of weights, a function σ is an activation function (ReLU), and $\mathcal{F}$ is a residual mapping function. For a basic residual block which contains two convolutional layers, the residual function can be defined as: $\mathcal{F}(x \cdot W) = W_2\sigma(W1 \cdot x)$. The Eq. (7) can be realized by 'shortcut connections' where perform identity mapping and skip one or more layers. The output of the Lth residual block is the sum of all lower residual blocks as:

$$x_L = \sigma(x_l + \sum_{i=1}^{L-1} \mathcal{F}(x_i \cdot W_i)) \tag{8}$$

for the loss function ϵ according to the chain rule of backpropagation we have:

$$\frac{\partial \epsilon}{\partial x_l} = \frac{\partial \epsilon}{\partial x_L} \times \frac{\partial x_L}{\partial x_l} \tag{9}$$

To obtain $\frac{\partial x_L}{\partial x_l}$, we consider $u = x_l + \sum_{i=1}^{L-1} \mathcal{F}(x_i \cdot W_i)$, then:

$$\frac{\partial x_L}{\partial x_l} = \frac{\partial \sigma}{\partial x_l} = \frac{\partial \sigma}{\partial u} \times \frac{\partial u}{\partial x_l} = \frac{\partial \sigma}{\partial u} \times (1 + \frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} \mathcal{F}(x_i \cdot W_i)) \tag{10}$$

we substitute Eq. (10) in Eq. (9) so we have:

$$\frac{\partial \epsilon}{\partial x_l} = \frac{\partial \epsilon}{\partial x_L} \times \frac{\partial x_L}{\partial x_l} = \frac{\partial \epsilon}{\partial x_L} \times (\frac{\partial \sigma}{\partial u} \times (1 + \frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} \mathcal{F}(x_i \cdot W_i))) \tag{11}$$

Eq. (11) indicates that the gradient $\frac{\partial \epsilon}{\partial x_l}$ propagate information with and without weight layers, so when the weights are arbitrarily small, the gradient of a layer does not vanish.

3.3. The conjugating approach

We propose a combined deep architecture for our handwriting identification problem, shown in Fig. 3. As mentioned in the previous Section, the residual networks solve vanishing gradients in deep networks by utilizing skip connections over some layers. Therefore, we employ a residual network with 18 layers, where four residual blocks with almost similar structures exist. The first residual block design is depicted in Fig. 4. As shown, the batch normalization (BN) and the activation function (i.e., ReLU), are both adopted before convolution layers. In each residual block, the same number of filters are used except the number of filters for the last convolutional layer, which is doubled to preserve the time complexity per layer. Therefore, the number of filters in residual blocks is 64, 128, 256, and 512, respectively. The computational complexity of our proposed model is linear on depth, which means a 100-layer network is ten times more complicated than a 10-layer similar network. Also, the BN, which is placed between the convolution and ReLU layers, is an effective way to accelerate CNN network training and dramatically improve performance.

In the original ResNet model, the primary input is a sequence of image patches. However, to make a rich representation, our model will also have an auxiliary input, receiving extra data from HTD. We feed our auxiliary input data into the model by conjugating it with the last residual block's flatten features output. The ResNet-18 explores a feature vector of 512 dimensions for each handwriting patch and our HTD, which provides a feature vector of 45 dimensions where we concatenate them to provide richer feature vector. For more details, the configuration of our model and its parameters is given in Table 1. According to this table, the total parameters of our architecture is about 11.54M. However, the complexity that our model adds in comparison with the standard ResNet is due to the last fully connected layer in which our auxiliary input data conjugated with the output of the features from the last residual block. We believe, along with the information extracted using developed deep models, the proposed HTD can perform and concentrate on local information. In the following, we provide multiple experiments and more information about our approach results.

Table 1
The configuration of our model architecture.

Layer	No. of activations	No. of parameters
Input layer	(60, 180, 1)	0
Convolutional layer	(30, 90, 64)	$7 \times 7 \times 64 + 64$
Max pooling layer	(15, 45, 64)	0
4 × Convolutional layer	(15, 45, 64)	$4 \times (3 \times 3 \times 64 \times 64 + 64)$
4 × batch normalization	(15, 45, 64)	4 × 256
Convolutional layer	(8, 23, 128)	$3 \times 3 \times 128 \times 64 + 128$
Convolutional layer	(8, 23, 128)	$1 \times 1 \times 128 \times 64 + 128$
3 × Convolutional layer	(8, 23, 128)	$3 \times (3 \times 3 \times 128 \times 128 + 128)$
Convolutional layer	(4, 12, 256)	$3 \times 3 \times 256 \times 128 + 256$
Convolutional layer	(4, 12, 256)	$1 \times 1 \times 256 \times 128 + 256$
3 × Convolutional layer	(4, 12, 256)	$3 \times (3 \times 3 \times 256 \times 256 + 256)$
Convolutional layer	(2, 6, 512)	$3 \times 3 \times 512 \times 256 + 512$
Convolutional layer	(2, 6, 512)	$1 \times 1 \times 512 \times 256 + 512$
3 × Convolutional layer	(2, 6, 512)	$3 \times (3 \times 3 \times 512 \times 512)$
Average pooling layer	(1, 1, 512)	0
Flatten layer	512	0
Concatenate with auxiliary input	512 + 45	0
Fully connected layer	No of classes	No of classes × 557 + No of classes

4. Experiments

In this section, we first introduce our employed dataset and protocols, then describe our experimental results along with comparisons.

4.1. Data

To evaluate the proposed approach, four public datasets of IAM (Marti and Bunke, 2002), Firemaker (Schomaker and Vuurpijl, 2000), CVL (Kleber et al., 2013), and CERUG-EN (He et al., 2015a) are used. In the following, we introduce the employed datasets and evaluation metrics and our data scanning approach.

4.1.1. IAM dataset

The IAM is one of the most popular and well-known English handwriting datasets contains 1539 documents from 657 writers. IAM has a different number of pages (from 1 to 59) per writer. However, the interesting point of this dataset is its data in different availability of pages, lines, and word images that provide various protocols for word recognition, handwriting identification, etc. Therefore, similar to Xing and Qiao (2016), we used text lines to extract the test and training patches. Thus we divided all text lines into two subsets and then extracted 1500 training patches from the first subset and 300 test patches from the second subset per writer for our model training and evaluation.

4.1.2. Firemaker dataset

The Firemaker dataset has 1000 handwriting pages containing the handwriting of 250 writers, which means 4 pages for each Dutch writers. The first page contains a short normal paragraph, the second page is written with uppercase, page 3 is written with forged handwriting, and the fourth page contains a story written by writers in their own words. Most of the writer identification approaches used page 1 and page 4 for analysis and validation. For instance, He and Schomaker (2020), Chahi et al. (2020), and Khan et al. (2019) used the first page as training data and the fourth page for evaluation. We used the first page of the handwriting similarly to extract 1500 patches with a fixed size of 60 × 180 for training our model and the fourth page to extract 300 same size patches for testing.

4.1.3. CVL dataset

The CVL handwriting dataset is another newer handwriting dataset for writer identification, word recognition, and similar challenges. It contains 310 unique writers with seven different samples where six of them are written in English and one in German for most of the writers. In general, the CVL has 1604 color handwriting images. Although the bounding box for each word is available, we used four English pages and extracted fixed size patches of 60 × 180 similar to Chahi et al. (2020). We extracted the patches from the first three pages as training data and used the fourth page's extracted patches for evaluation. Therefore, the number of training patches is 224 to 850, which is very small for learning our model. Therefore, we augmented all training data and provided 1200 to 7300 patches per writer, and we then used 1200 patches per each writer for the training. Also, as the number of patches in the test subset was similarly various (between 42 to 2100), we selected 40 random test patches per writer for evaluation.

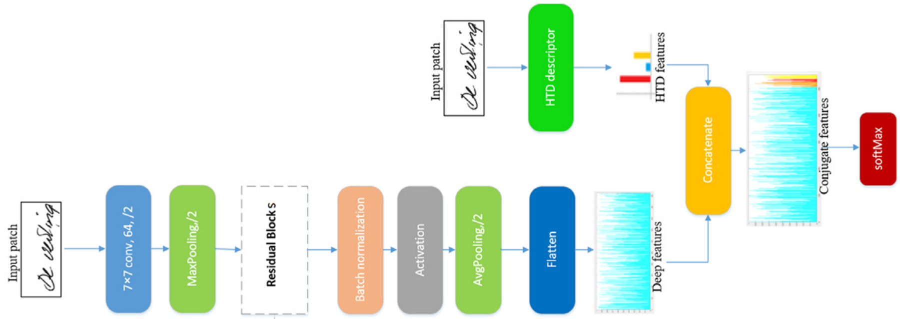

Fig. 3. The main architecture of the proposed model.

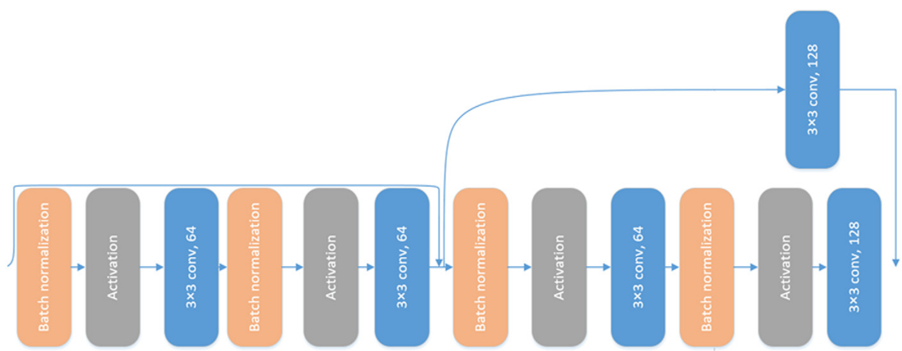

Fig. 4. The residual block in the proposed model.

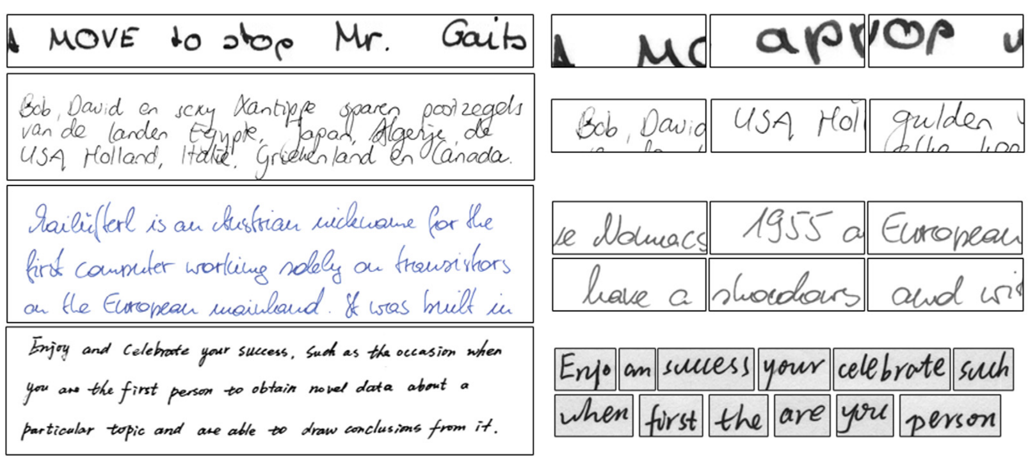

Fig. 5. Examples of extracted patches with our patch scanning strategy. From top to bottom: IAM, Firemaker, CVL and CERUG-EN datasets. (Left) Main documents, (Right) Samples of extracted patches.

4.1.4. CERUG-EN dataset

The CERUG is a challenging multiple script dataset containing the handwriting of 105 Chinese writers. Like the Firemaker, it has four pages per writer where the third page is famous as CERUG-EN because it includes two paragraphs in English. All documents scanned grayscale with 300 dpi resolution. In our work, we followed the protocol of Sheng He (He and Schomaker, 2020) and used his distributed data. He provided the CERUG-EG dataset into two fixed subsets of test and train word images. We extracted patches with a fixed size of 60 × 180 from all word images. Since the number of word images in the dataset is not the same for all writers, we have a various number of patches for each writer. The number of extracted training patches was between 54 and 810 samples for writers that we augmented to become 600 to 1500 patches per writer. Also, the number of patches in the test subset was between 31 to 787, that we selected 30 random test patches for evaluation.

4.2. Patch scanning strategy

There are various protocols according to the different possibilities of using data in handwriting writer identification datasets. As described in Section 2, some works used the whole of handwriting pages, and several works used handwriting lines, and a few used word images and alphabets patches. As we explained before, we use a patch-based strategy of scanning handwriting images similar to Xing and Qiao (2016) for several reasons. First, it is a model-friendly strategy to exploit fix size of handwriting images. Second, the dataset of handwriting lines is in different sizes and, of course, affects our CNN-based model. Third, we believe stretching handwriting images in width or length changes the writer's personality behind the handwriting. Therefore, we proposed to crop patches in fix size of 60 × 180 from handwriting images, which are almost equal or less than a word. Nevertheless, we defined some constraints that the cropped patches which have 10 800 pixels

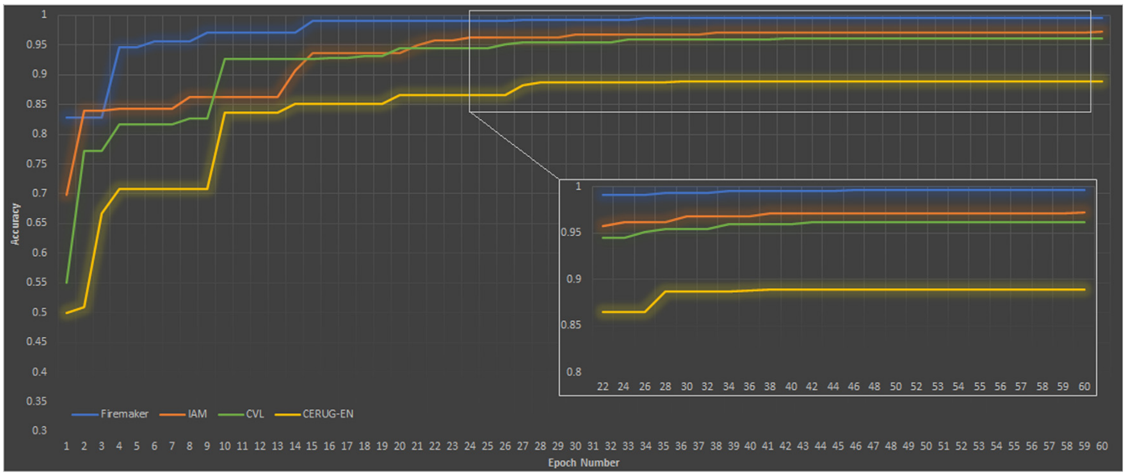

Fig. 6. Our approach results progress during 60 epoch on four datasets of IAM, Firemaker, CVL, and CERUG-EN.

Table 2
Performance of the proposed approach on IAM dataset with different batch size setting.

Batch size	8	16	32	64	128
Accuracy	91.07%	96.75%	97.19%	97.50%	97.45%

Table 3
Kernel size comparison. The $C\alpha S\beta P\sigma$ notation specifies that the convolutional layer filter with kernels of size $\alpha \times \alpha$ with a stride of β pixels and a padding of σ pixels.

Kernel size	Conv1:C7S2P3 Conv2:C3S1P2	Conv1:C15S2P7 Conv2:C7S1P3
Accuracy	97.50%	93.03%

must include at least 10% black pixels. It means that all patches carry some information regarding the handwriting, and we do not generate patches without or with less information. Moreover, we preferred to automatically generate patches with scripts in the middle of the image because the handwriting information will be at maximum availability. As an example, some patches are shown in Fig. 5.

4.3. Parameters setting

In all experiments, we used constant settings for our network and validations. For instance, the batch size and the maximum number of epochs were defined to 64 and 100, respectively; however, in all experiments, the final results achieved before reaching this maximum value. Furthermore, Table 2 is provided for evaluating our network based on the batch size on the IAM dataset. The comparison of the results shows that we achieve the best accuracy using a batch size of 64. Therefore we set the batch size parameter to 64 in all of our experiments. However, our memory limitation did not allow us to try a batch larger than 128. Also, in the original ResNet, the convolutional layers mostly have filters with a size of 3×3. To better evaluate this parameter, we doubled the filters' size and trained the network on the IAM dataset. As shown in Table 3, the accuracy of standard ResNet, and this new adjustment has hardly changed. It is also evident that this structure increases the number of parameters and, consequently, increases the learning cost in terms of time and preparing numerous training samples. Therefore, we prefer to adjust this parameter as standard ResNet to preserve the network generalization. Moreover, the weights were initialized using the method proposed in He et al. (2015b), although the weights updating algorithm for training of our deep network was based on Adam optimizer (Kingma and Adam, 2014). We adopted the learning rate of 0.001 and reduced it by a factor of 0.3 once learning stagnates for five epochs.

Table 4
Writer identification comparison with the conjugate approach in contrast to the simple method.

Approach	Simple method		Conjugate method	
Dataset	No. of params	Acc.	No. of params	Acc.
IAM	11,510,289	95.25%	11,539,854 (+0.26%)	97.50%
Firemaker	11,301,498	99.51%	11,312,748 (+0.09%)	99.61%
CVL	11,332,278	95.82%	11,346,228 (+0.12%)	96.16%
CERUG-EN	11,227,113	88.35%	11,231,838 (+0.04%)	89.14%

Table 5
Text independent writer identification comparison with the related works on IAM dataset.

Proposed approach	Year	No.	Accuracy
Xing and Qiao (2016)	2016	657	96.92%
Khan et al. (2017)	2017	**650**	**97.20%**
He and Schomaker (2017)	2017	650	89.90%
Pandey and Seeja (2018)	2018	650	88.57%
Chahi et al. (2018)	2018	657	90.11%
Nguyen et al. (2019)	2019	650	90.12%
Chahi et al. (2019)	2019	657	91.17%
He and Schomaker (2019)	2019	657	92.00%
Chahi et al. (2020)	2020	657	94.06%
He and Schomaker (2020)	2020	657	96.30%
Our simple method	–	657	95.25%
Our conjugate approach	–	**657**	**97.50%**

4.4. Experimental results

We evaluated our approach on four well-known datasets, as introduced in 4.1. We used Keras functional API that works with both Theano-TensorFlow to train our deep network. Moreover, we used an Nvidia GEFORCE GTX 1070 Ti Graphics card with 2432 Cuda core and 8 GB RAM regarding the hardware.

In the first experiment, we evaluated the usefulness of our conjugating idea of deep and traditional features. We analyzed all four introduced datasets regarding the writer identification from extracted patches. It means that, in our preliminary protocol, we separated input text images into two subsets of train and test collection, then extracted handwriting patches and learned our network with the training set, later evaluated the test patches. In the second protocol, we similarly extracted the patches but described them using proposed HTD and conjugate them with our deep network features at the flatten layer. Finally, the features are evaluated for writer identification. We name the first protocol as the simple method and the second protocol as a conjugate method for quick reference. To provide more details for comparing our network complexity regarding the number of network

Table 6
Text independent writer identification comparison with the related works on Firemaker dataset.

Proposed approach	Year	No.	Accuracy
Brink et al. (2012)	2012	250	86.00%
Ghiasi and Safabakhsh (2013)	2013	250	91.80%
Wu et al. (2014)	2014	250	92.40%
He et al. (2015a)	2015	250	89.80%
Khan et al. (2017)	2017	250	89.47%
Nguyen et al. (2019)	2019	250	92.38%
Khan et al. (2019)	2019	250	97.98%
Chahi et al. (2020)	2020	250	97.60%
He and Schomaker (2020)	2020	250	97.60%
Our simple method	–	250	99.51%
Our conjugate approach	–	**250**	**99.61%**

Table 7
Text independent writer identification comparison with the related works on CVL dataset.

Proposed approach	Year	No.	Accuracy
Fiel and Sablatnig (2015)	2015	310	98.90%
Tang and Wu (2016)	2016	310	99.70%
Hannad et al. (2016)	2016	310	96.20%
Khan et al. (2017)	2017	310	99.60%
Mohammed et al. (2017)	2017	310	**99.80%**
Chahi et al. (2018)	2018	**309**	98.38%
He and Schomaker (2019)	2019	310	79.10%
Chahi et al. (2019)	2019	310	99.35%
He and Schomaker (2020)	2020	310	99.10%
Chahi et al. (2020)	2020	310	99.67%
Our simple method	–	310	95.82%
Our conjugate approach	–	310	**96.16%**

parameters Table 4 presents the effectiveness of the parameters in simple and conjugated methods on all four datasets. As can be seen, however, a negligible number of parameters (less than one percent) have been added to the baseline network, our conjugate method has consistent results compared to the simple method.

In the following, we also provided related comparisons for all four datasets. We achieved 97.50% accuracy on the IAM dataset, which is better than the state-of-the-art. Furthermore, Table 5 provides a comparison of the related works with ours. In this table, however, Khan et al. (2017) reported accuracy (97.20%) close to ours but for 650 writers in contrast to our experiment with 657 writers. Moreover, our investigation on the Firemaker dataset shows significant improvement than the state-of-the-art, where we achieved 99.61% accuracy, which is 1.63% better than Khan et al. (2019). More details can be seen in Table 6. We also evaluated and compared our approach result on the CVL dataset shown in Table 7, where we achieved 96.16% accuracy on this dataset. Finally, we evaluated our approach on CERUG-EN, which is a more challenging dataset. The result and comparison are shown in Table 8, where our approach reached 88.95% accuracy that outperformed the state-of-the-art.

Furthermore, we have shown our method progress during the 60 epochs of running the proposed network on all four datasets in Fig. 6 that reveals its fast growth during the early epochs. As can be seen, the network trained quickly in fewer epochs; the graph has also confirmed this fact in the last 15 epochs, where the accuracy curves for all four datasets are rarely changed.

4.5. Training data effectiveness

We evaluated the influence of the number of training data with our proposed approach reported in Table 9. In this experiment, we selected some parts of the IAM training data (10%, 20%, …, 100%) and evaluate our approach, while the whole of the training data is 1500 patches per person. Therefore, 10% of the training data means, we only used 150 patches per person for training our model. Also, in

Table 8
Text independent writer identification comparison with the related works on CERUG-EN dataset.

Proposed approach	Year	No.	Accuracy
ResNet18 (He et al., 2016a)	2015	105	70.60%
Junclets (He et al., 2015a)	2016	105	87.10%
$LBPrunsG_{hv}$ (He and Schomaker, 2017)	2016	105	88.50%
WordImgNet (He and Schomaker, 2020)	2020	105	77.30%
FragNet (He and Schomaker, 2020)	2020	105	77.50%
Our simple method	–	105	88.35%
Our conjugate approach	–	105	**88.95%**

this experiment, the number of test data is always constant and equal to 300 patches. Fig. 7 shows our model accuracy details. As can be seen, increasing the number of training data from 150 (10%) patches has a remarkable performance in the identification accuracy during all epochs. Noticeably, the accuracy curves for the least amount of data (10%) are so far from other curves that used more training data. Thus, the best result obtained when the most training number of data (100%) is used.

5. Discussion

As described, the proposed network lets us properly join the deep feature obtained via residual network and traditional features in a conjugation framework and takes advantage of both features. Although our network is not as deep as the original ResNet with 152, 101, or even 50 layers, nevertheless, using the residual version instead of plain network yields less variation in response layers, which has been recommended by He et al. (2016a). The results also reveal the response strength of the residual network, where the reported accuracy in the previous section is comparable with the methods that originally used CNN-based approaches or even better in most cases.

Also, analyzing the proposed approach regarding the standard experiments shows that while the deep residual networks are very successful for image analysis and feature description, but there is still information that deep networks miss them. Such information extracted by our HTD increases the identification rate, as reported in Table 4. We can find it by comparing our achievements with the related studies. For instance, the FragNet (He and Schomaker, 2020), which is recently published and is very close to our work, exploits a deep approach along with the fragments information. For this purpose, the FragNet fuses the information from the convolutional layer of the fragment pathway with the information from the corresponding convolutional layer of the feature pyramid. They also achieved good results, but we reached still better accuracy on IAM and Firemaker. If we assume our patches are equal or less than a word, our accuracy on CERUG-EN also outperformed their reported result. Our differences in accuracy are due to our HTD classic descriptor versus their fragment description. However, the complexity of their network, especially in terms of the number of trainable parameters, has increased almost doubled. To prove this claim, we refer to the number of floating-point operations (FLOPs) of their method which was reported 3.9G FLOPs compared with standard ResNet with 1.8G FLOPs.

Nevertheless, in contrast to our achievements, there was a strange issue regarding the CVL dataset. While our approach shows its usefulness for writer identification on two datasets of IAM and Firemaker, its result on CVL, shown in Table 7, was not expectable. To this end, we intensely focused on the reason(s) and found that both IAM and Firemaker datasets have different text contents (Firemaker has one same text and one variant). However, CVL uses the same text for all writers. In the CVL experiment, we found that our networks not only learned the writer's handwriting styles but also learned the handwriting contents while in IAM and Firemaker, our system learned only the handwriting style due to the various text contents. Therefore, this conflict in CVL eventuates and leads to decreasing accuracy.

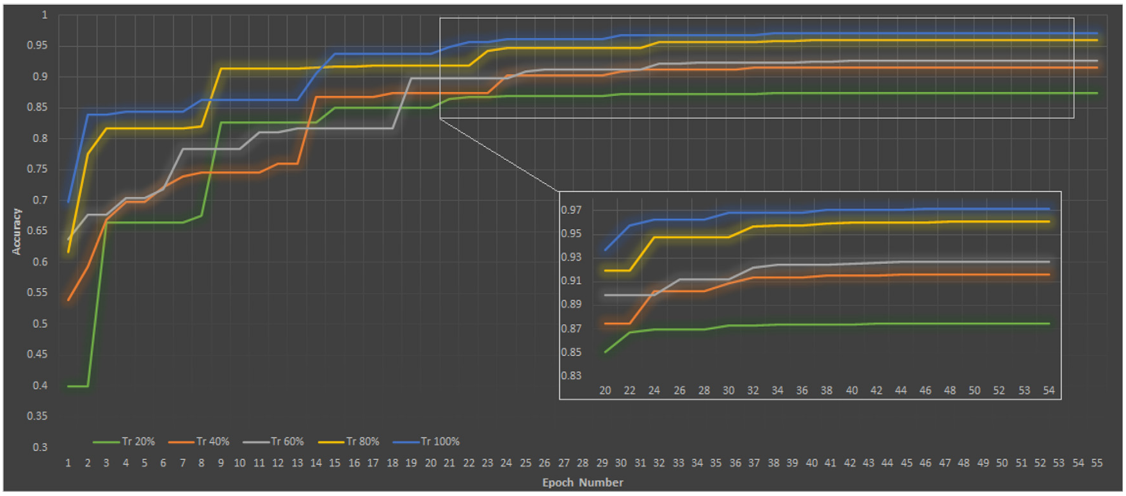

Fig. 7. Influence of the number of training data on the accuracy of the proposed method on the IAM dataset.

Table 9
Comparison accuracy results with different training number (%) on IAM dataset.

Training samples	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Accuracy (%)	81.92	87.46	89.97	91.62	92.44	92.65	94.63	96.04	96.39	97.50

6. Conclusion

This paper proposed an offline text-independent handwriting writer identification approach based on the conjugation of the deep and traditional features. Proposed deep architecture is an extended version of ResNet, which is reached using the auxiliary information of handwriting thickness descriptor (HTD). The HTD computed the thickness of handwriting as an essential and preliminary feature for human handwriting analysis. We showed that the conjugation of such information with deep description is useful and improves writer identification accuracy. We also provided extensive evaluations on four public handwriting datasets, including the IAM, Firemaker, CVL, and CERUG-EN. Our comparisons with the-state-of-the art show that our approach achieved a par with or better than the related works on IAM, Firemaker, and CERUG-EN. More reasonable features and multimodal descriptions can be directions for future works.

CRediT authorship contribution statement

Malihe Javidi: Formal analysis, Methodology, Validation, Writing - review & editing. **Mahdi Jampour:** Conceptualization, Investigation, Resources, Writing - original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Authors would like to thank Dr. Mohammad A. Moarrefi for his help.

References

Brink, A., Smit, J., Bulacu, M., Schomaker, L., 2012. Writer identification using directional ink-trace width measurements. Pattern Recognit. 45 (1), 162–171.
Chahi, A., khadiri, I.E., merabet, Y.E., Ruichek, Y., Touahni, R., 2018. Block wise local binary count for on-line text-independent writer identification. Expert Syst. Appl. 93, 1–14.
Chahi, A., merabet, Y.E., Ruichek, Y., Touahni, R., 2019. An effective and conceptually simple feature representation for off-line text-independent writer identification. Expert Syst. Appl. 123, 357–376.
Chahi, A., merabet, Y.E., Ruichek, Y., Touahni, R., 2020. Cross multi-scale locally encoded gradient patterns for off-line text-independent writer identification. Eng. Appl. Artif. Intell. 89 (1), 103459.
Crespo, Y., Soriano, M.F., Iglesias-Parro, S., Aznarte, J.I., Ibez-Molina, A.J., 2018. Spatial analysis of handwritten texts as a marker of cognitive control. J. Motor Behav. 50 (6), 643–652.
De Stefano, C., Maniaci, M., Fontanella, F., Scotto di Freca, A., 2018. Reliable writer identification in medieval manuscripts through page layout features: The “Avila” Bible case. Eng. Appl. Artif. Intell. 72, 99–110.
Fiel, S., Sablatnig, R., 2015. Writer identification and retrieval using a convolutional neural network. In: Computer Analysis of Images and Patterns. Springer International Publishing, pp. 26–37.
Ghiasi, G., Safabakhsh, R., 2013. Offline text-independent writer identification using codebook and efficient code extraction methods. Image Vis. Comput. 31 (5), 379–391.
Hannad, Y., Siddiqi, I., Kettani, M.E.Y.E., 2016. Writer identification using texture descriptors of handwritten fragments. Expert Syst. Appl. 47, 14–22.
He, S., Schomaker, L., 2017. Writer identification using curvature-free features. Pattern Recognit. 63, 451–464.
He, S., Schomaker, L., 2019. Deep adaptive learning for writer identification based on single handwritten word images. Pattern Recognit. 88, 64–74.
He, S., Schomaker, L., 2020. FragNet: Writer identification using deep fragment networks. IEEE Trans. Inf. Forensics Secur. 15, 3013–3022.
He, K., Sun, J., 2015. Convolutional neural networks at constrained time cost. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5353–5360.
He, S., Wiering, M., Schomaker, L., 2015a. Junction detection in handwritten documents and its application to writer identification. Pattern Recognit. 48 (12), 4036–4048.
He, K., Zhang, X., Ren, S., Sun, J., 2015b. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: IEEE International Conference on Computer Vision (ICCV), Santiago, pp. 1026-1034.
He, K., Zhang, X., Ren, S., Sun, J., 2016a. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778.
He, K., Zhang, X., Ren, S., Sun, J., 2016b. Identity mappings in deepresidual networks. In: Computer Vision - ECCV 2016. pp. 630–645.
Jampour, M., Naserasadi, A., 2019. Chaos game theory and its application for offline signature identification. IET Biom. 8 (5), 316–324.
Khan, F.A., Khelifi, F., Tahir, M.A., Bouridane, A., 2019. Dissimilarity Gaussian mixture models for efficient offline handwritten text-independent identification using SIFT and RootSIFT descriptors. IEEE Trans. Inf. Forensics Secur. 14 (2), 289–303.
Khan, F.A., Tahir, M.A., Khelifi, F., Bouridane, A., Almotaeryi, R., 2017. Robust off-line text independent writer identification using bagged discrete cosine transform features. Expert Syst. Appl. 71, 404–415.
Kingma, Diederik P., Adam, Jimmy Ba, 2014. A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR).

Kleber, Florian, Fiel, Stefan, Diem, Markus, Sablatnig, Robert, 2013. CVL-database: An off-line database for writer retrieval, writer identification and word spotting. In: Proc. of the 12th Int. Conference on Document Analysis and Recognition (ICDAR), pp. 560-564.

Marti, U., Bunke, H., 2002. The IAM-database: An english sentence database for off-line handwriting recognition. Int. J. Doc. Anal. Recognit. (IJDAR) 5 (1), 39–46.

Mohammed, H., Mäergner, V., Konidaris, T., Stiehl, H.S., 2017. Normalised local Naïve Bayes nearest-neighbour classifier for offline writer identification. In: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, pp. 1013-1018.

Nguyen, H.T., Nguyen, C.T., Ino, T., Indurkhya, B., Nakagawa, M., 2019. Text-independent writer identification using convolutional neural network. Pattern Recognit. Lett. 121, 104–112.

Ni, K., Callier, P., Hatch, B., 2017. Writer identification in noisy handwritten documents. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1177–1186.

Pandey, P., Seeja, K.R., 2018. Forensic writer identification with projection profile representation of graphemes. In: Proceedings of First International Conference on Smart System, Innovations and Computing, pp. 129-136.

Schomaker, L., Vuurpijl, L., 2000. Forensic Writer Identification: A Benchmark Data Set and a Comparison of Two Systems. (research report), the Netherlands Forensic Institute.

Tang, Y., Wu, X., 2016. Text-independent writer identification via cnn features and joint bayesian. In: 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 566–571.

Wu, X., Tang, Y., Bu, W., 2014. Offline text-independent writer identification based on scale invariant feature transform. IEEE Trans. Inf. Forensics Secur. 9 (3), 526–536.

Xing, L., Qiao, Y., 2016. Deepwriter: A multi-stream deep cnn for text-independent writer identification. In: 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 584–589.

Xiong, Y.-J., Lu, Y., Wang, P.S.-P., 2017. On-line text-independent writer recognition: A survey. Int. J. Pattern Recognit. Artif. Intell. 31 (5), 1756008.