International Journal of Research in Industrial Engineering

www.riejournal.com

Int. J. Res. Ind. Eng. Vol. 13, No. 1 (2024) 88–103.

Paper Type: Research Paper

# Leveraging Deep Feature Learning for Handwriting Biometric Authentication

**Parvaneh Afzali[1], Abdoreza Rezapour[2,*], Ahmad Rezaee Jordehi[3]**

[1] Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran; parvaneh.afzali@iau.ac.ir.

[2] Department of Computer Engineering, Astaneh Ashrafieh Branch, Islamic Azad University, Astaneh Ashrafieh, Iran; abdoreza.rezapour@gmail.com.

[3] Department of Electrical Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran; ahmad.rezaeejordehi@iau.ac.ir.

**Citation:**

| | |
|---|---|
| *Received: 11 October 2023*<br>*Revised: 12 December 2023*<br>*Accepted: 01 Januray 2024* | Afzali, P., Rezapour, A., & Rezaee Jordehi, A. (2024). Leveraging deep feature learning for handwriting biometric authentication. *International journal of research in industrial engineering*, *13*(1), 88-103. |

**Abstract**

The authentication of writers through handwritten text stands as a biometric technique with considerable practical importance in the field of document forensics and literary history. The verification process involves a meticulous examination of the questioned handwriting in comparison to the genuine handwriting of a known writer, aiming to determine whether a shared authorship exists. In real-world scenarios, writer verification based on the handwritten text presents more challenges compared to signatures. Signatures typically consist of fixed designs chosen by signers, whereas textual content can vary and encompass a diverse set of letters, numbers, and punctuation marks. Moreover, verifying a writer based on limited handwritten texts, such as a single word, is recognized as one of authentication's open and challenging aspects. In this paper, we propose a Customized Siamese Convolutional Neural Network (CSCNN) for offline writer verification based on handwritten words. Additionally, a combined loss function is employed to achieve more accurate discrimination between the handwriting styles of different writers. The designed model is trained with pairs of images, each comprising one authentic and one questioned handwritten word. The effectiveness of the proposed model is substantiated through experimental results obtained from two well-known datasets in both English and Arabic, IAM and IFN/ENIT. These results underscore the efficiency and performance of our model across diverse linguistic contexts.

**Keywords:** Writer verification, Siamese neural network, Feature learning, Combined loss function.

## 1 | Introduction

Handwritten texts have held a special significance in human relations, and the writer verification of a written text has always been considered a biometric authentication method [1]. Due to the fact that each person's handwriting has unique and measurable traits that distinguish the writer, the verification process can be done

Corresponding Author: abdoreza.rezapour@gmail.com

https://doi.org/10.22105/riej.2024.432510.1412

[2]. The writer verification involves the task of determining whether two handwritten texts belong to the same writer [3]. Based on the methods of handwritten data collection, two approaches are proposed in the writer verification systems: online and offline. A digital device collects the handwriting samples in the online approach. Therefore, various dynamic features such as the writing speed and the pen pressure level can be accessible. However, in the offline approach, no temporal and dynamic data about the handwriting is available, and only the scanned image of the written text can be used [3]. However, because it is impossible to collect dynamic information in the analysis of historical documents or some forensic documents, the offline approach is vastly applicable [4]. Handwriting-based authentication systems can also be divided into text-dependent and text-independent categories. In the text-dependent group, the text of the training data and the test data must be identical, but there is no limit on textual content in the text-independent group [5]. The writer verification is a significant topic in computer vision, and many verification methods have been proposed; nevertheless, there are still some open challenges. For instance, most proposed verification methods are ineffective for writer verification based on a single handwritten word rather than multiple handwriting lines.

In this paper, a new offline text-independent framework is proposed to verify the writer based on handwriting. In the proposed approach using the Siamese network, a pair of images is entered into two identical subnetworks, and the obtained feature vectors are used to detect the similarity between the original and the questioned handwriting images.

The significant contributions of this study can be summarized as follows:

I. An efficient writer verification framework is proposed to extract discriminative characteristics based on handwritten word images.

II. As the backbone of the proposed Siamese network, a customized Convolutional Neural Network (CNN) model is designed for the subnetworks.

III. A combined loss function that incorporates both contrastive loss and cross-entropy loss is utilized to enhance the accuracy in discriminating between the handwriting styles of different writers. This approach allows us to simultaneously leverage the strengths of these two loss functions for more effective discrimination.

IV. The proposed approach is tested on two well-known English and Arabic handwriting datasets, IAM and IFN/ENIT, to demonstrate its generalizability to different languages and handwriting styles.

The rest of this paper is organized as follows. Section 2 includes a brief overview of the related research. The description of the suggested method is provided in the Section 3. Section 4 is dedicated to describing the datasets used to evaluate the performance of the proposed verification model, as well as the results and discussion. Finally, Section 5 concludes the paper.

## 2|Literature Review

In this section, some state-of-the-art writer verification approaches are reviewed. Various methods have been proposed to verify the writer using machine learning and deep learning techniques, but most of these studies are focused on signatures, not handwriting.

Yilmaz et al. [6] applied several descriptors and a concatenation of classifiers to use the local signature features and the overall shape of the signature. In [7], a writer verification approach is proposed by examining a unique fragment of the handwritten word and using the Levenshtein edit distance. The proposed approach was tested on a part of the IAM dataset including 100 authors, and 87% accuracy was achieved.

In [8], a CNN was applied to extract the features of the signature samples. The proposed model was writer-dependent, meaning that a separate training set was assigned to each writer. Dey et al. [9] proposed a structure named SigNet, which was based on the Siamese network and received pairs of signature images to verify the signature.

Jang et al. [10] presented an approach based on the modified Hausdorff distance and the geometric features to verify the writer from Korean texts. In this method, the RGB images were converted into grey images, and the images were separated from the background with a binarization threshold. Then, the noises were detected and removed. Next, the images were created using the narrowing algorithm. The corners, endings, and intersection points were detected, and the segments between these points were removed. Finally, the distance criterion was determined without normalization and with normalization. The error rate was 37% without normalization and 36% with normalization.

Adak et al. [11] proposed an offline writer verification approach for Bengali scripts using hybrid features. In this method, the probability distribution function was first used to extract the handcrafted features. Then, these features were combined with the features automatically obtained from the CNNs. Finally, the combined features were applied as the input of the Siamese network. The proposed approach was evaluated using a dataset of 100 authors with 300 pages of Bengali manuscripts. Another hybrid deep method is proposed by Shaikh et al. for the handwriting verification of Shaikh et al. [1]. In this research, the engineered features were extracted using SIFT and GSC methods and combined with extracted features using CNN. In order to evaluate the proposed model, only the most frequent word "and" extracted from the CEDAR dataset, which contains 15518 words from 1567 authors, was used. Although the feature extractions using SIFT and GSC complement the deep networks, according to the results, the manual feature extraction process is very time-consuming, and 70% of the learning time is spent on this task. Using a combination of SIFT features and the deep Siamese network, the training accuracy of 99% and the test accuracy of 63% were achieved, indicating overfitting in the system.

Calik et al. [3] provided a new structure based on CNN for signature recognition on large-scale datasets. In the proposed structure, the nearest neighbor algorithm was also used to classify the extracted feature vectors of the dense layers. The evaluations were performed on GPDS, MCYT, and CEDAR dataset samples.

Maergner et al. [12] proposed a combination of two structural and statistical classifiers for signature verification. A triplet Siamese-based architecture, including three subnetworks sharing the same weights, was used as a statistical model. In the training phase, a triplet of signatures (the anchor image, the positive image, and the negative image) was fed to the model. Also, the graph edit distance was employed in the structural classifier. Parcham et al. [4] provided a new model with high performance for signature verification. In this model, a hybrid architecture named CBCapsNet, including the customized CNN and capsule neural network models, was presented to improve the model's capability in feature extraction and increase accuracy. The researchers tried to use the benefits of the convolutional networks to identify and extract the features. They also attempted to lessen the weakness of the CNNs in distinguishing the spatial changes and variations in image properties by utilizing the capsule neural networks.

Gosh [13] proposed a deep model using recurrent neural networks for signature verification and recognition. In this research, several features of the signature images were extracted, and the obtained feature vectors were classified. The method was evaluated on six public databases. Furthermore, a comparison was made with the CNNs, and a better result was achieved than with these networks. In [14], researchers proposed a new method of using generative adversarial networks as a data augmentation approach in training sets to solve the problem of limited data in the signature verification problem. The researchers evaluated their proposed method using two popular datasets GPDS and MCYT and four pre-trained CNN models, and acceptable results were obtained.

Aubin et al. [15] presented a new method using small segments of the graphemes. Two texture descriptors were used on five ordinary graphemes of a collected dataset of 3000 sample images written by 50 people. Using the support vector machine as a classifier, this research reported an average verification accuracy of 97%. Khan et al. [16] provided an approach to verify the writer based on partly damaged Arabic documents. The authors attempted to improve the verification performance by omitting the ineffective characters and focusing on the character shapes. By evaluating a collected Arabic dataset using a CNN, an accuracy of 95% was obtained.

The review of related works reveals a scarcity of studies in the realm of writer verification centered on handwriting, with the majority concentrating on signature-based approaches. Signatures, typically fixed designs chosen by the writer, offer a limited scope. However, the realm of writer verification through handwriting introduces a greater complexity, given the variability in textual content, encompassing a diverse array of letters, numbers, and various symbols. This intricacy makes writing verification based on handwriting more challenging than signature-based methods. Unlike signature-based approaches, which focus on a fixed design, our study delves into the dynamic nature of handwriting, specifically concentrating on single handwritten words.

Furthermore, our approach stands out by removing the dependence on language-specific graphemes. Our method avoids this limitation, unlike many existing methodologies in previous research, which focus on extracting features tied to particular graphemes. This departure enhances the generalizability of our approach, making it more versatile and applicable across diverse languages.

In light of these observations, our current study assumes a pivotal role in bridging this research gap. By presenting an innovative approach to writer verification based on a single handwritten word independent of language-specific graphemes, we aim to contribute to a more applicable and robust methodology in the realm of handwriting-based writer verification.

# 3|The Proposed Customized Siamese Convolutional Neural Network Methodology

In this section, we present the architecture of the proposed methodology for writer verification. This architecture consists of a customized convolutional Siamese network (cSCNN), including two convolutional subnetworks with a shared architecture and equal weights.

CNNs are a special class of deep neural networks widely used in various machine vision tasks [17]. CNNs include several important layers, such as convolutional, pooling, and Fully Connected (FC) layers [18]. As the core block of the network, the convolutional layers can have different kernel sizes. The lower convolutional layers extract low-level features like color and edge, while high-level features like lines and objects are extracted in the upper layers. After the convolutional layer, the pooling layer performs the down-sampling of the outputs. Based on the extracted features, the prediction can be done in the FC layer [19].

The Siamese neural network includes two subnetworks with identical configurations as a specific neural network architecture. The parameters update is reflected in both branches and is connected through a loss function. This function calculates a similarity measure between the feature vectors obtained from two subnetworks [20]. This network has achieved acceptable results in various challenging issues of machine vision, such as face verification and signature verification [9].

The detailed information about the architecture of the proposed cSCNN is illustrated in *Fig. 1*. The subnetworks of our cSCNN utilize a proposed deep architecture consisting of five convolutional blocks (Conv-Block). Two convolutional layers are embedded in each Conv-Block. A Batch Normalization (BN) layer is applied after each convolutional layer to extract the most salient features, discard less significant details, improve the generalization, and aid in faster and more effective training. Applying dropout after BN in the first three Conv-Blocks is attempted to enhance the generalizability and decrease the risk of overfitting. Each conv-Block is ended with a max pooling layer with the size of 2×2 and stride 2 to capture the most important handwriting information while discarding the redundant details. The pair of handwriting images (size: 80×180) is fed as the input to the customized subnetworks. In all convolutional layers, the convolution operation is performed by sliding the filters of size 3×3 over the input handwriting images. The stride and the padding are also set to 1. To enhance the non-linear transformations in the feature learning process [21], we use the interval type 2 fuzzy unit proposed in [22] as the activation function. The high-level handwriting features can be captured by increasing the number of filters from 32 to 512 while going deeper into the Conv-Blocks and combining the low-level features. Following the convolutional blocks, Global Average

Aggregation (GAP), aiming to reduce the number of parameters and the complexity of the model, and the flattened layer, converting the multi-dimensional dimension into a one-dimensional vector, finalize the architecture of the customized Siamese subnetworks. The architecture configuration details of each subnetwork are also presented in *Table 1*. Moreover, *Fig. 2* illustrates the pseudocode of the proposed writer verification framework.

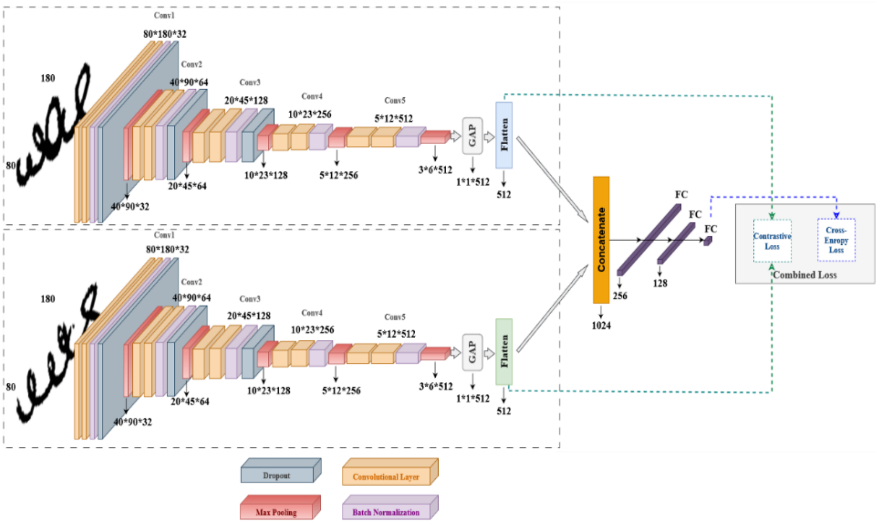



**Fig. 1. The custom architecture used in cSCNN.**

A concatenation layer is then applied to the two output vectors of the deep subnetworks. Afterward, the concatenated vector is passed through the three FC layers. The output of the last FC layer determines whether the verification result is Genuine or Forged, meaning that both images of the pair are written by the same writer (Similar) or different writers (Dissimilar), respectively.

**Table 1. The architecture configuration details of the identical subnetworks.**

| Layers | Number of Kernels | Kernel Size | Output Size |
|---|---|---|---|
| 2 × Convolution2D+BN | 32 | 3×3 | 80×180×32 |
| Max pooling2D | | | 40×90×32 |
| 2 × Convolution2D+BN | 64 | 3×3 | 40×90×64 |
| Max pooling2D | | | 20×45×64 |
| 2 × Convolution2D+BN | 128 | 3×3 | 20×45×128 |
| Max pooling2D | | | 10×22×128 |
| 2 × Convolution2D+BN | 256 | 3×3 | 10×22×256 |
| Max pooling2D | | | 5×11×256 |
| 2 × Convolution2D+BN | 512 | 3×3 | 5×11×512 |
| Max pooling2D | | | 3×5×512 |
| Global Average Pooling | | | 1×1×512 |
| Flatten | | | 512 |

```
Input:
  Handwriting Dataset: Handwriting images;
  Hyperparameters: num_epochs: Number of Epochs; batch_size: Batch size; learning_rate: learning rate
Output:
  Evaluated performance of the model

Preprocessing
  Convert handwriting_samples to grayscale images
  Crop images into fixed-size patches
  Normalize pixel values
  Create pairs of images and corresponding labels (1 for positive/similar pairs, 0 for negative/dissimlar pairs)
  Implement Data Augmentation(Brightness Adjustment, Zooming, Random Rotation, Random Noise)

Training the Proposed Verification  Model
 Initialize the customized Siamese Convolutional Neural Network with two identical Subnetworks

Training Loop:
while epoch ≤ num_epochs do
   Divide the Data into batches of batch_size
   for each batch of handwriting paired_images do
      embedding1 ← Extract deep features from Subnetwork1
      embedding2 ← Extract deep features from Subnetwork2
      concatenated_features ← Concatenate the extracted features
      output ← Pass concatenated_features through two fully connected layers
      prediction ← Pass output through dense layer for writer classification
      Contrastive_Loss ← Compute the loss (embedding1, embedding2 )
      Binary_Cross_Entropy_loss ← Compute the loss (prediction, true label)
      Combined_Loss ← a.Contrastive_Loss + b.Binary_Cross_Entropy_loss
      update the model parameters by computing gradients through backpropagation and utilizing Adam optimizer with  the learning_rate
   end for
end while

Evaluating the Proposed Verification Model
Accuracy; FAR; FRR ← Evaluate the model performance based on the test data
```

**Fig. 2. The pseudocode of the proposed writer verification framework.**

## 3.1 | Training and Evaluating of the Proposed Model for Writer Verification

The training process of the proposed model for writer verification requires the creation of image pairs after preprocessing. Thus, for a pair of images, if two handwritten images belong to the same writer, it is a positive pair; otherwise, it is considered a negative pair. The proposed deep Siamese network is trained on these positive and negative pairs. During the training phase, the proposed model extracts embeddings from each image in the input pair. Then, these embeddings are used to calculate the loss. By iteratively adjusting the parameters based on the calculated loss, the model learns to produce embeddings that effectively discriminate between genuine and questioned handwriting. A pair of two handwriting images is presented as an input element to test the model. One image belongs to a specific writer, and the other is the questioned handwriting that needs to be verified. An overview of the training and testing of the proposed model for writer verification is shown in *Fig. 3*.

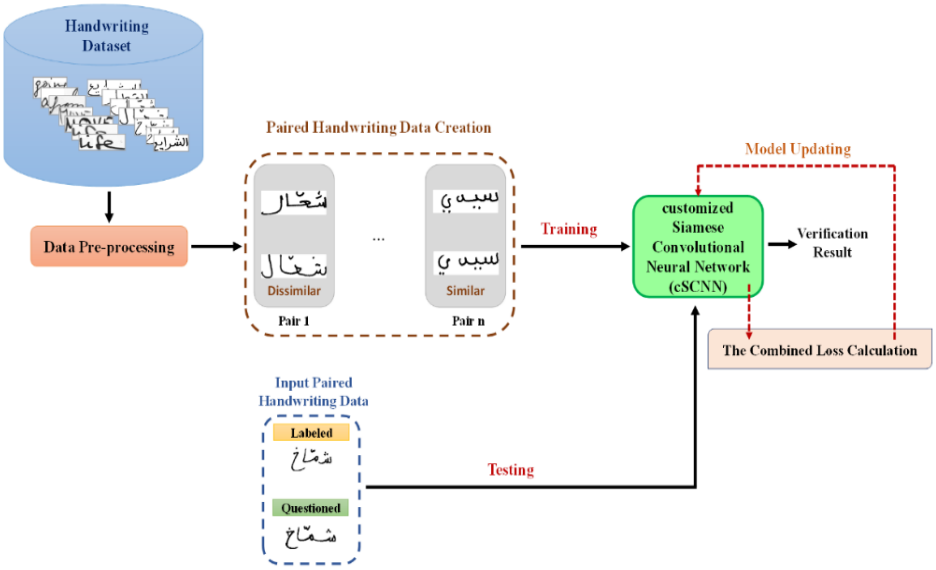



**Fig. 3. The flow diagram of cSCNN for writer verification.**

### 3.1.1 | The proposed combined loss

Loss function is essential for training neural networks or machine learning models [23]. This function provides evidence of estimation errors in the training phase and directly affects the learning performance [24]. Each loss function can include different aspects of learning objectives.

In this study, a combined loss function is provided to enhance the training process and model performance. During the training of the proposed model, two vectors obtained from the Flatten layers in two Siamese subnetworks are used to apply the contrastive loss function [25], and the cross-entropy loss function [26] is also applied to the output of the last FC layer. As the components of the proposed combined function, the cross-entropy loss function focuses on Binary classification accuracy, and the Contrastive loss function emphasizes the accurate separation and discrimination between similar and dissimilar handwriting samples.

**Contrastive loss**

The contrastive loss function is widely employed for comparing two samples. The cSCNN leverages this loss function to highlight significant features within the feature space, bringing similar samples (written by the same writer) closer together while pushing dissimilar samples (written by different writers) farther apart. This loss function is applied to the feature vectors obtained from two siamese subnetworks.

The Contrastive loss is obtained as follows:

$$L_C = \frac{1}{2N} \sum_{i=1}^{N} (y_i D_i^2 + (1 - y_i) \max(0, \text{margin} - D_i)^2), \quad (1)$$

where $D_i$ denotes the Euclidean distance between two feature representations of the pair of handwritten images. A pair of input images is positive/similar when the same writer writes yi=1 and the two handwriting images; otherwise, it is considered a negative/dissimilar pair for $y_i$=0. The margin determines the desired separation threshold between similar and dissimilar samples.

**Cross-entropy loss**

The cross-entropy loss function is one of the most beneficial and widely used loss functions in neural networks. The measured prediction is a number between zero and one. The main goal is to achieve a model with a log loss around zero [24].

The cross-entropy loss is formulated as follows:

$$L_{CE} = -\frac{1}{N}\sum_{n=1}^{N}(\text{similarity\_label} * \log(\text{predicted\_similarity}) + (1 - \text{similarity\_label}) * \log(1 - \text{predicted\_similarity}))], \quad (2)$$

where the similarity_label is either 0 or 1, indicating whether the pair is dissimilar (0) or similar (1). The predicted_similarity is the network output that represents the predicted similarity between the pair of handwriting samples, and it is a value between 0 and 1. The 1/N factor is applied to average the loss of overall data pairs. This averaging helps ensure that each pair of handwriting data equally influences the overall loss calculation.

In the suggested approach, cross-entropy loss is applied to the outputs of the last FC layer and allows the model to be optimized for accurate classification.

**The combined loss formula**

The combined loss is formulated as the linear combination of the employed loss functions *Eq. (3)* and applied to train the proposed network structure by optimizing all the loss functions with back-propagation at the same time.

$$\text{Combined Loss} = \alpha L_C + \beta L_{CE}, \quad (3)$$

where α and β are the weights to balance the contributions of each loss function based on the desired trade-off.

# 4 | Experiments and Results

This section details the datasets and experiments conducted to evaluate the proposed method.

## 4.1 | Data

The well-known datasets of IAM [27] and IFN/ENIT [28] are employed to train and evaluate the proposed model.

### 4.1.1 | The IAM dataset

The IAM dataset, renowned for its collection of English handwriting samples, encompasses 1539 documents originating from 657 unique writers. This dataset is meticulously organized, featuring handwriting images across various document elements, including pages, lines, and words. Within this rich dataset, a total of 1539 pages, 5685 sentences, 13353 text lines, and 115320 labeled words are available [27]. For our study, we utilized the handwriting images of 130 writers from this dataset to generate both similar and dissimilar handwriting samples. Refer to *Fig. 4* for visual representations of sample images from this dataset.

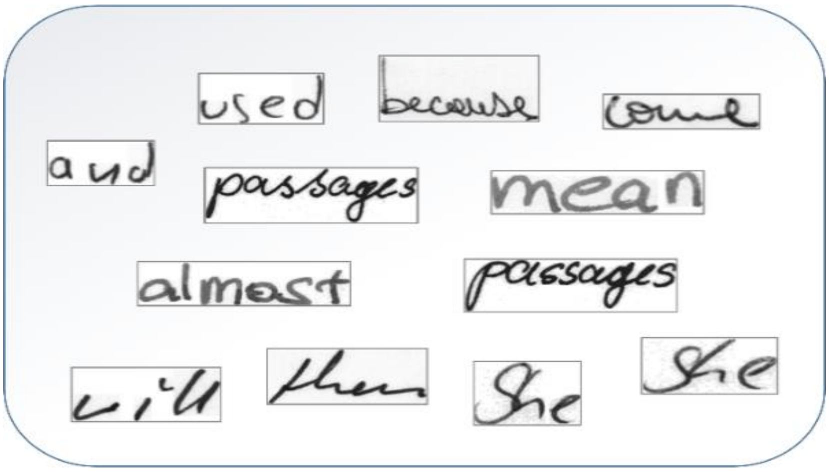


**Fig. 4. Some instances of handwritten words from the IAM dataset.**

### 4.1.2 | The IFN/ENIT dataset

The IFN/ENIT dataset, a comprehensive collection of Arabic handwriting, encompasses 26,459 handwritten names representing Tunisian cities or villages, contributed by 411 participants [28]. To construct a diverse set of handwriting samples, we specifically leveraged the handwriting images of 120 writers from this extensive dataset. For a visual representation, please refer to *Fig. 5*, which showcases a selection of examples featuring handwritten words extracted from the IFN/ENIT dataset. This dataset provides a rich source of Arabic handwriting and facilitates the creation of varied samples for our study.

**Fig. 5. Some exemplars of handwritten words from the IFN/ENIT dataset.**

### 4.1.3 | Preprocessing and preparing the pair of input images

Preprocessing is crucial for effective feature extraction and precise analysis in the field of writer verification using handwriting images. Thus, as the initial step, we converted the collected samples into grayscale images. The handwritten word images within the datasets vary in size, requiring standardization for consistent model input. To achieve this, we meticulously cropped the images into uniform 80×180 patches to ensure that the model receives standardized input across all samples. Moreover, we normalized pixel values to foster an environment conducive to effective model learning and performance optimization.

In the realm of data augmentation, we employed a comprehensive set of transformations to enhance dataset variability. Rotation, with a variability of 0.8, was introduced to simulate diverse perspectives. Zooming, within a range of 0.1, and brightness adjustments spanning from 0.4 to 0.7 were applied to augment the dataset further, providing the model with a more diverse and robust set of samples for training.

Addressing the creation of pairs for training, positive (similar) pairs were meticulously curated by selecting two samples from the handwriting of a single writer. Similarly, negative (dissimilar) pairs were formed by pairing two samples from different writers. Importantly, to counteract potential biases and ensure a balanced dataset, an equal number of positive and negative samples were generated for each writer.

Then, turning to the data-splitting strategy, 70% of the word samples were allocated for training, providing the model with a substantial foundation for learning. A reserved 10% was set aside for validation and fine-tuning during the training process, aiding in parameter adjustments and enhancing model generalization. The final 20% was dedicated to evaluating the model's performance on unseen data during the testing phase. The distribution of training and testing data for the two datasets is detailed in *Table 2*.

**Table 2. The data splitting on IAM and IFN/ENIT datasets.**

| Dataset | # Pair Images | # Training Data (70%) | | # Validation Data (10%) | | # Testing Data (20%) | |
|---|---|---|---|---|---|---|---|
| | | Similar | Dissimilar | Similar | Dissimilar | Similar | Dissimilar |
| IAM | 2600 | 910 | 910 | 130 | 130 | 260 | 260 |
| IFN/ENIT | 2040 | 714 | 714 | 102 | 102 | 204 | 204 |

## 4.2|Hyperparameters Setting

Fine-tuning the hyperparameters, as detailed in *Table 3*, included setting the learning rate to 0.0001 and the batch size to 64. A dynamic adjustment strategy was applied to address potential learning stagnation, reducing the learning rate by 0.2 whenever stagnation persisted for five consecutive epochs.

The model was trained for 70 epochs, optimizing parameters using the Adam optimizer and the Binary cross-entropy loss function. The experiments were conducted on an Nvidia GEFORCE GTX 1070 with 8 GB RAM, ensuring computational efficiency and robust model training.

**Table 3. List of training hyperparameters.**

| Hyperparameters | Value |
|---|---|
| Initial learning rate | 0.0001 |
| Batch size | 64 |
| Dropout | 0.3 |
| Optimizer | Adam |
| Loss function | Binary cross-entropy |
| Epoch | 70 |

## 4.3 | Evaluation Metrics

In our study, the chosen criteria to evaluate the verification performance are as follows:

I. Accuracy (Acc): the standard metric of accuracy is used to evaluate how well the proposed approach can verify the writer of the handwritten text and identify the genuine/similar and forged/dissimilar handwriting.

$$\mathrm{Acc} = (\mathrm{TP} + \mathrm{TN}) / (\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}), \tag{4}$$

where True Positives (TP) is the number of samples correctly predicted as genuine/similar handwriting. True Negatives (TN) is the number of samples correctly predicted as forged/dissimilar handwriting. False Positives (FP) is the number of samples incorrectly predicted as genuine/similar handwriting. False Negatives (FN) is the number of samples incorrectly predicted as negative forged/dissimilar handwriting.

II. False Acceptance Rate (FAR): This metric is commonly used in verification systems and refers to the percentage of forged/dissimilar handwriting pairs classified as genuine/similar. A lower FAR demonstrates that the biometric system is more reliable and mistakenly accepts fewer forged inputs.

$$\mathrm{FAR} = \mathrm{FP} / (\mathrm{TN} + \mathrm{FP}). \tag{5}$$

III. False Rejection Rate (FRR): this is another common biometric performance metric used in verification systems and refers to the percentage of genuine/similar handwriting pairs classified as forged/dissimilar. A lower FRR indicates that the biometric system mistakenly rejects fewer genuine handwritings.

$$\mathrm{FRR} = \mathrm{FN} / (\mathrm{TP} + \mathrm{FN}). \tag{6}$$

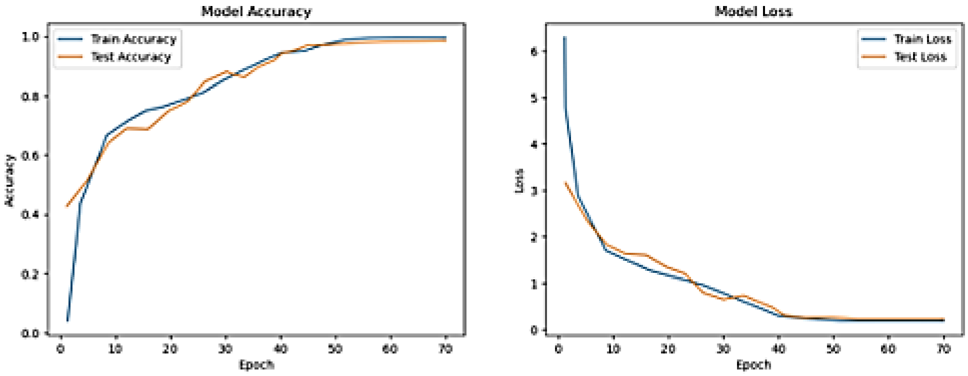
## 4.4|Evaluation Results of the Proposed Writer Verification Architecture

The effectiveness of the cSCNN architecture is assessed using IAM and IFN/ENIT datasets. The evaluation outcomes, measured in terms of accuracy, FAR, and FRR, are presented in *Table 4*.

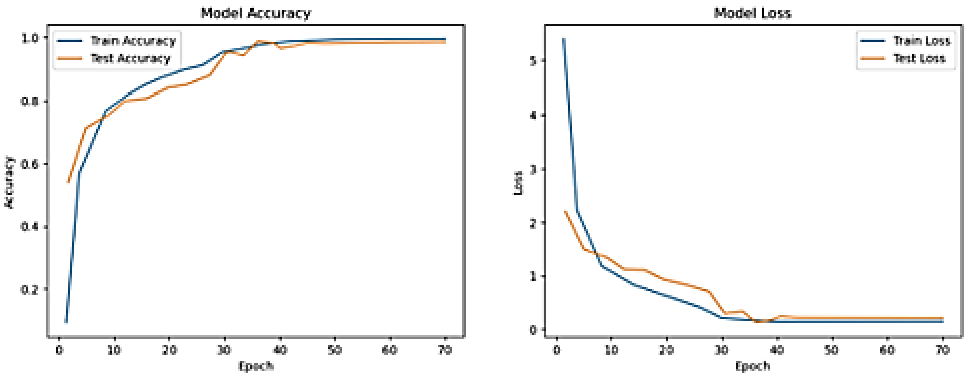
**Table 4. The performance results of the cSCNN model on IAM and IFN/ENIT datasets.**

| Dataset | Language | Train accuracy (%) | Validation accuracy (%) | Test accuracy (%) | FAR (%) | FRR (%) |
|---|---|---|---|---|---|---|
| IAM | English | 99.61 | 98.65 | 98.46 | 1.92 | 1.15 |
| IFN/ENIT | Arabic | 99.50 | 98.77 | 98.52 | 1.47 | 1.47 |

Moreover, *Fig. 6* illustrates the performance of the proposed model during both the training and test phases.

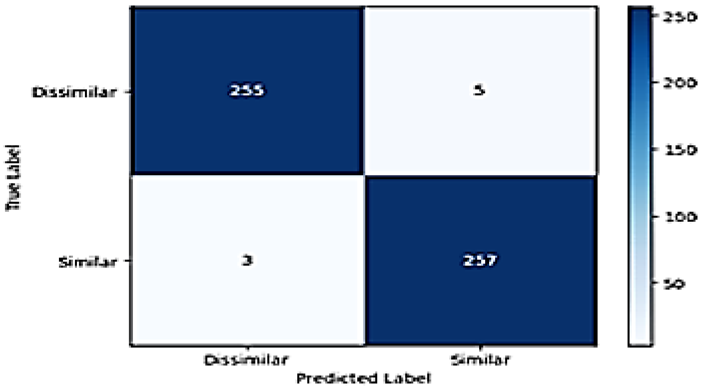



a.

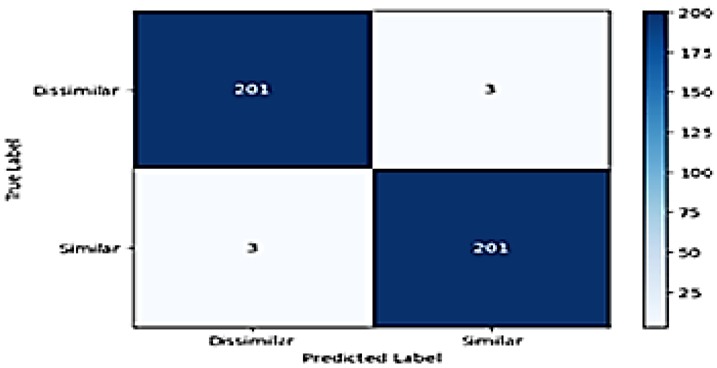



b.

Fig. 6. Accuracy and loss plots of the proposed cSCNN model on; a. IAM and b. IFN/ENIT.




a.



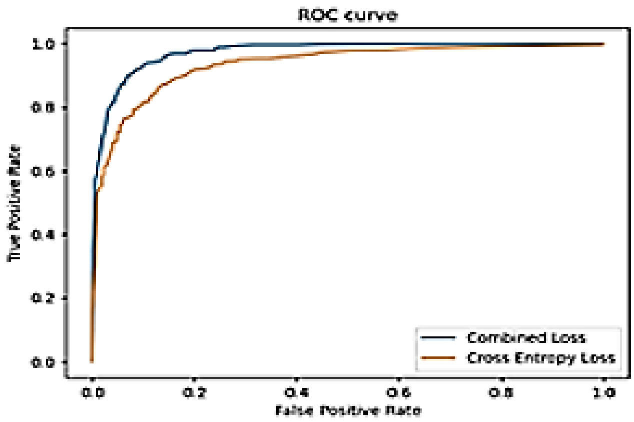

b.

Fig. 7. The confusion matrices of cSCNN on; a. IAM and b. IFN/ENIT.

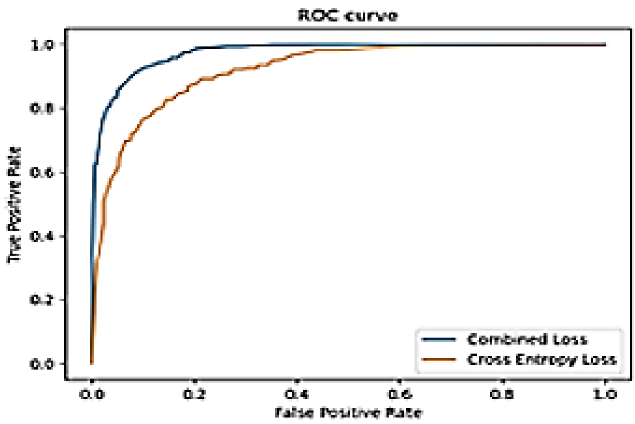
In addition, the confusion matrices of the cSCNN model on two datasets are displayed in *Fig. 7*.

## 4.5 | The Impact of Different Loss Functions

The next experiment evaluated the influence of using the proposed combined loss function compared to a single cross-entropy loss function. ROC curves were used to illustrate the comparison results. As shown in *Fig. 8*, applying the combined loss function to the proposed model, a higher area under the curve was obtained on both IAM and IFN/ENIT.




a.




b.

**Fig. 8. ROC curve of the verification model with proposed combined loss function and cross-entropy loss function on; a. IAM and b. IFN/ENIT.**

## 4.6 | The Impact of Different Numbers of Training Pairs

In another experiment, the effect of the number of training pairs per writer on the performance of the designed model is assessed. We assessed the verification performance by utilizing varying similar and dissimilar pairs (3 to 7) for each writer in the IAM dataset. As seen in *Table 5*, the verification accuracy is impacted by the increase in the number of training pairs. Also, a substantial improvement is observed when 5 training pairs are utilized for each author.

**Table 5. The effect of different numbers of the training pairs on the verification performance.**

| The number of training pairs (similar, dissimilar) | Train accuracy (%) | Test accuracy (%) |
|---|---|---|
| (3,3) | 92.50 | 89.53 |
| (4,4) | 95.82 | 93.74 |
| (5,5) | 99.32 | 98.15 |
| (6,6) | 99.47 | 98.41 |
| (7,7) | 99.60 | 98.44 |

## 4.7 | The Comparison with the Related Writer Verification Methods and Discussion

*Table 6* compares the accuracy of the proposed method with other writer verification approaches. In comparison to the study by Bensefia et al. [7], which sought to verify writers by analyzing the constituent graphemes of a handwriting word sample, our research demonstrates a notable improvement in accuracy. Specifically, our proposed method achieved a significant 11.46% enhancement over [7] when evaluated on the IAM dataset. This improvement underscores the efficacy and superiority of our cSCNN model in comparison to the approach presented by Bensefia et al.

Aubin et al.'s method [15], while achieving 97% accuracy on a specific dataset, is confined by its reliance on only five typical graphemes. This limitation raises concerns about the model's generalizability. In contrast, our proposed cSCNN is designed to offer greater versatility by not being restricted to a small set of graphemes. Our comprehensive evaluation of distinct datasets emphasizes the importance of considering model performance across diverse datasets to ensure reliability and robustness.

Similarly, the approach outlined by Khan et al. [16], which focuses on specific shapes of Arabic alphabets, presents limitations in its applicability to other languages and writing systems. Its reliance on predefined shapes hinders adaptability to different writing styles and scripts. In contrast, our method prioritizes feature extraction and generalization, offering a more flexible and adaptable solution for writer verification. By not solely depending on predefined shapes, our approach becomes applicable to a broader range of languages and handwriting styles.

Our novel writer verification method focuses specifically on handwritten words, addressing the observed limitations in previous approaches. Through extensive experimentation and evaluation of datasets representing both English and Arabic languages, our cSCNN showcased its ability to extract distinctive features effectively, thereby enabling reliable writer verification. The promising results obtained on datasets with different languages highlight the potential for our approach to be widely applicable and adaptable in real-world scenarios.

**Table 6. The performance comparison with the related writer verification methods.**

| Study | Method | Dataset | Accuracy (%) |
|---|---|---|---|
| Bensefia et al. [7] | Levenshtein edit distance | IAM (English) | 87.00 |
| Aubin et al. [15] | Two texture descriptors and support vector machine | Collected English dataset | 97.00 |
| Khan et al. [16] | CNN | Collected Arabic dataset | 95.00 |
| Present study | cSCNN | IAM (English) | 98.46 |
| | | IFN/ENIT (Arabic) | 98.52 |

# 5 | Conclusion

Various challenges are involved in analyzing handwritten text for writer verification, such as extracting more informative and differentiating features, achieving higher performance in real-world applications, and enabling writer authentication based on a small amount of handwritten text. In response to these challenges, this paper introduced an effective method for improved writer verification from single handwritten words.

Our proposed deep architecture is based on a Siamese network with a custom CNN model as the backbone of the sub-networks. This design aims to enhance handwriting verification performance by incorporating specially designed convolution blocks in each Siamese branch. The model is trained using a composite loss function, including contrastive and cross-entropy losses. This integration of multiple loss functions facilitates efficient extraction of handwriting characteristics, enhancing the model's ability to capture fundamental patterns and, consequently, improving overall performance compared to using a single loss function.

Notably, our model underwent evaluation on two diverse datasets representing different languages, English and Arabic, with the added distinction that one dataset is written right-to-left and the other left-to-right. The experimental results underscored the versatility and efficacy of our proposed model in handling various handwriting styles and languages, including the directional nature of the script.

In addition, our innovative approach diverges from the reliance on language-specific graphemes, setting it apart from numerous existing methodologies in prior research that concentrate on extracting features associated with specific graphemes. This deviation eliminates a potential limitation and significantly bolsters the generalizability of our method, rendering it more adaptable and applicable across a wide spectrum of languages.

Looking ahead, the potential of our proposed framework for effectively detecting forged signatures and identifying writers can be explored in future works. Additionally, we plan to enhance the model's capabilities by training it on a dataset of cursive English handwritten samples, presenting unique challenges due to their intricate connections.

In summary, our study contributes a novel and robust approach to writer verification from single handwritten words, demonstrating versatility across languages and superior performance compared to existing methods. The consideration of different directional datasets and the departure from language-specific graphemes add additional layers of complexity and relevance to the evaluation, highlighting the adaptability of our proposed model to diverse handwriting styles, scripts, and languages. The potential applications for detecting forged signatures and handling cursive English handwriting present exciting avenues for future exploration and expansion of our proposed framework.

# Data Availability

The used datasets can be found at (https://fki.tic.heia-fr.ch/databases/download-the-iam-handwriting-database) and (http://www.ifnenit.com/download.htm).

# Conflicts of Interest

The authors declare no conflict of interest.

# References

[1] Shaikh, M. A., Chauhan, M., Chu, J., & Srihari, S. (2018). Hybrid feature learning for handwriting verification. *2018 16th international conference on frontiers in handwriting recognition (ICFHR)* (pp. 187–192). IEEE. DOI: 10.1109/ICFHR-2018.2018.00041

[2] 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 root SIFT descriptors. *IEEE transactions on information forensics and security*, *14*(2), 289–303. DOI:10.1109/TIFS.2018.2850011

[3] Çalik, N., Kurban, O. C., Yilmaz, A. R., Yildirim, T., & Durak Ata, L. (2019). Large-scale offline signature recognition VIA deep neural networks and feature embedding. *Neurocomputing*, *359*, 1–14. https://www.sciencedirect.com/science/article/pii/S0925231219303674

[4] Parcham, E., Ilbeygi, M., & Amini, M. (2021). CBCapsNet: a novel writer-independent offline signature verification model using a CNN-based architecture and capsule neural networks. *Expert systems with applications*, *185*, 115649. https://www.sciencedirect.com/science/article/pii/S095741742101040X

[5] Xiong, Y. J., Lu, Y., & Wang, P. S. P. (2017). Off-line text-independent writer recognition: a survey. *International journal of pattern recognition and artificial intelligence*, *31*(5), 1756008. https://doi.org/10.1142/S0218001417560080

[6] Yılmaz, M. B., & Yanıkoğlu, B. (2016). Score level fusion of classifiers in off-line signature verification. *Information fusion*, *32*, 109–119. https://www.sciencedirect.com/science/article/pii/S1566253516300033

[7] Bensefia, A., & Paquet, T. (2016). Writer verification based on a single handwriting word samples. *Eurasip journal on image and video processing*, *2016*(1), 34. https://doi.org/10.1186/s13640-016-0139-0

[8] Hafemann, L. G., Sabourin, R., & Oliveira, L. S. (2017). Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern recognition*, *70*, 163–176. https://www.sciencedirect.com/science/article/pii/S0031320317302017

[9] Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Lladós, J., & Pal, U. (2017). *Signet: convolutional siamese network for writer independent offline signature verification*. https://arxiv.org/abs/1707.02131

[10] Jang, W., Kim, S., Kim, Y., & Lee, E. C. (2018). Automated verification method of korean word handwriting using geometric feature. *Advances in computer science and ubiquitous computing* (pp. 1340–1345). Singapore: springer singapore.

[11] Adak, C., Marinai, S., Chaudhuri, B. B., & Blumenstein, M. (2018). Offline bengali writer verification by pdf-cnn and siamese net. *2018 13th iapr international workshop on document analysis systems (DAS)* (pp. 381–386). IEEE. DOI: 10.1109/DAS.2018.33

[12] Maergner, P., Pondenkandath, V., Alberti, M., Liwicki, M., Riesen, K., Ingold, R., & Fischer, A. (2019). Combining graph edit distance and triplet networks for offline signature verification. *Pattern recognition letters*, *125*, 527–533. https://www.sciencedirect.com/science/article/pii/S0167865519301850

[13] Ghosh, R. (2021). A recurrent neural network based deep learning model for offline signature verification and recognition system. *Expert systems with applications*, *168*, 114249. https://www.sciencedirect.com/science/article/pii/S0957417420309659

[14] Yapıcı, M. M., Tekerek, A., & Topaloğlu, N. (2021). Deep learning-based data augmentation method and signature verification system for offline handwritten signature. *Pattern analysis and applications*, *24*(1), 165–179. https://doi.org/10.1007/s10044-020-00912-6

[15] Aubin, V., Mora, M., & Santos, M. (2022). A new approach for writer verification based on segments of handwritten graphemes. *Logic journal of the igpl*, *30*(6), 965–978. https://doi.org/10.1093/jigpal/jzac006

[16] Khan, M. A., Mohammad, N., Brahim, G. Ben, Bashar, A., & Latif, G. (2022). Writer verification of partially damaged handwritten Arabic documents based on individual character shapes. *PeerJ computer science*, *8*, 1–28. https://peerj.com/articles/cs-955/

[17] Sadr, H., & Nazari Soleimandarabi, M. (2022). ACNN-TL: attention-based convolutional neural network coupling with transfer learning and contextualized word representation for enhancing the performance of sentiment classification. *The journal of supercomputing*, *78*(7), 10149–10175. https://doi.org/10.1007/s11227-021-04208-2

[18] Javidi, M., & Jampour, M. (2020). A deep learning framework for text-independent writer identification. *Engineering applications of artificial intelligence*, *95*, 103912. https://www.sciencedirect.com/science/article/pii/S0952197620302463

[19] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., … & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of big data*, *8*(1), 8-53. https://doi.org/10.1186/s40537-021-00444-8

[20] Chakladar, D. Das, Kumar, P., Roy, P. P., Dogra, D. P., Scheme, E., & Chang, V. (2021). A multimodal-siamese neural network (MSNN) for person verification using signatures and EEG. *Information fusion*, *71*, 17–27. https://www.sciencedirect.com/science/article/pii/S1566253521000105

[21] Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: a comprehensive survey and benchmark. *Neurocomputing*, *503*, 92–108. https://www.sciencedirect.com/science/article/pii/S0925231222008426

[22] Beke, A., & Kumbasar, T. (2019). Learning with type-2 fuzzy activation functions to improve the performance of deep neural networks. *Engineering applications of artificial intelligence*, *85*, 372–384. https://www.sciencedirect.com/science/article/pii/S0952197619301551

[23] Zhang, Q., Pei, M., Chen, M., & Jia, Y. (2018). Vehicle verification based on deep siamese network with similarity metric. *Advances in multimedia information processing-PCM 2017* (pp. 773–782). Cham: springer international publishing. https://doi.org/10.1007/978-3-319-77380-3_74

[24] Tian, Y., Su, D., Lauria, S., & Liu, X. (2022). Recent advances on loss functions in deep learning for computer vision. *Neurocomputing*, *497*, 129–158. https://www.sciencedirect.com/science/article/pii/S0925231222005239

[25] Wang, F., & Liu, H. (2021). Understanding the behaviour of contrastive loss. *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 2495–2504). IEEE. DOI: 10.1109/CVPR46437.2021.00252

[26] De Boer, P. T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of operations research*, *134*(1), 19–67. https://doi.org/10.1007/s10479-005-5724-z

[27] Marti, U. V., & Bunke, H. (2002). The IAM-database: an english sentence database for offline handwriting recognition. *International journal on document analysis and recognition*, *5*(1), 39–46. https://doi.org/10.1007/s100320200071

[28] El Abed, H., & Margner, V. (2007). The ifn/enit-database-a tool to develop arabic handwriting recognition systems. *2007 9th international symposium on signal processing and its applications* (pp. 1–4). IEEE. DOI: 10.1109/ISSPA.2007.4555529