Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 14, Issue. 10, October 2025, pg.94 – 109

Interval-Oriented Feature Modelling in Writer-Dependent Offline Signature Verification

Annapurna H[1]; D. S. Guru[2]; K. S. Manjunatha[3]

[1,2]Department of Studies in Computer Science, Manasagangotri, University of Mysore, Mysuru, Karnataka, 570006, India
[3]Department of Computer Science, Government First Grade College, Nanjangud, Karnataka, 571301, India

[1] annapurnah2015@gmail.com; [2] dsg@compsci.uni-mysore.ac.in; [3] kowshik.manjunath@gmail.com

DOI: https://doi.org/10.47760/ijcsmc.2025.v14i10.008

Abstract: As technology advances, automated signature verification systems have become essential due to the increasing use of digital signatures in various fields, such as banking, finance, legal document verification, forensic departments, and industry. The main objective of these systems is to determine whether a given signature of an individual is genuine or forged, thereby authenticating their identity. This paper presents an approach for offline signature verification that uses writer-dependent features. In this work, signature samples are analyzed using a combination of global and grid features. Writer-dependent features are then selected using a filter-based feature selection technique. These features are represented as Interval-oriented feature vectors, which effectively preserve intra-class variations. Additionally, this representation scheme minimizes the number of reference feature vectors that need to be stored in the knowledge base, thereby speeding up the matching process. A symbolic classifier is adopted to enhance the efficiency of the signature verification process. Two sets of experiments are carried out: first, using a common feature set and a global threshold across all writers, and second, using writer-dependent characteristics. Extensive experiments are conducted on two benchmarking offline signature datasets, CEDAR and MCYT, and the results demonstrate the effectiveness of the proposed model.

Keywords: Decision-threshold, Interval-oriented feature vector, Offline signature, Symbolic classifier, Writer-dependent features.

I. INTRODUCTION

As technology advances, biometric authentication systems are becoming increasingly popular due to their high reliability in authentication compared to traditional methods like PINs, passwords, or smart cards. These systems utilize unique physiological or behavioural characteristics such as face, iris, hand geometry, fingerprint, signature, voice, and gait to authenticate individuals. Among these, handwritten signatures are the most commonly used behavioural biometric trait for authentication purposes. Automated signature verification systems have become essential with the increasing use of digital signatures in various fields, such as banking, finance, legal document verification, forensic departments, and industries [116]. The main goal of any signature verification system is to determine whether a given signature of an individual is genuine or forged, thereby authenticating their identity [40]. Hence, developing an efficient signature verification model suitable for real-time applications is crucial.

Biometric sensors or high-resolution cameras are used to collect signatures through either online (dynamic) or offline (static) modes. Offline signatures are captured using optical scanners or cameras, whereas online signatures are recorded using tools such as digitizing tablets, intelligent pens, and personal digital assistants. In the online mode, dynamic features of signatures, such as velocity, pressure, position, and acceleration, can be recorded during the signature acquisition process [35][84]. Offline signature verification, however, is more challenging than online verification due to the absence of dynamic features, the limited availability of signature samples, and the higher susceptibility to forgery. As a result, automatic signature verification in biometrics continues to be an active area of research [3].

This work explores the significance of writer-dependent characteristics for offline signature verification. When verifying a signature manually, human experts typically identify unique characteristics for each writer and apply different similarity thresholds to determine the genuineness of a signature. Verification requires distinct features for each individual because signature styles vary widely. To address this, we propose an offline signature verification approach based on writer-dependent characteristics, including features, feature dimensions, and decision-thresholds. In our approach, global and grid features are used to represent each signature sample. Writer-dependent features are then selected using a filter-based feature selection method. An Interval-oriented representation scheme is employed, effectively preserving intra-class variations [29]. Finally, a symbolic classifier is utilized to establish the authenticity of test signature samples during the verification process. To demonstrate the effectiveness of writer-dependent traits, we conducted two sets of experiments: one using a common feature set for all writers, and the other using writer-dependent characteristics. Our proposed model is tested on two benchmark offline signature datasets: CEDAR and MCYT.

The remainder of the paper is organized as follows: Section 2 provides a brief literature review, followed by Section 3, which explains the steps involved in the proposed model. Section 4 outlines the experimental setup, datasets, and results, while Section 5 presents a comparative analysis of the proposed model. Section 6 highlights directions for future work, and Section 7 concludes with a summary of the proposed approach.

II. LITERATURE REVIEW

Over the past 50 years, researchers have explored various models for offline signature verification. Signature verification is a multi-stage process that includes preprocessing, feature extraction, representation, and verification [18].

A. *Preprocessing*

The process typically begins with preprocessing techniques, which include binarization [26,37,71,104], filtering [2,37,63,92], inversion [66,106], morphological operations [26,37], scaling [5,26,66,104], and cropping [104].

B. *Feature extraction*

Machine learning algorithms and deep convolutional neural networks (DCNNs) are commonly used to derive features from signature images [35]. Handcrafted features are extracted using machine learning algorithms, which include local features [5,26,48,89,94,111,117] or global features [10,21,42,76,89,92,112]. Researchers have applied various mathematical transformations to extract both local and global features from signature images, such as the Discrete Fourier Transform (DFT) [75,77], Discrete Radon Transform, Hough Transform [88], Gabor Wavelet Transform [96], Fractal Transform [115], Curvelet Transform [28,30], and Contourlet Transform [77]. Deep features, on the other hand, are extracted through deep convolutional neural networks. Various deep architectures have been used in the literature to extract these features from signature images, including VGG-16 [6,19], ResNet-50 [19], Deep Convolutional Generative Adversarial Networks (DCGANs) [110], one-class convolutional neural networks [91], Siamese Neural Networks (SNN) [17,41], Capsule Neural Networks [74], Signet [31], and architectures like VGG-16, Inception-v3, ResNet-50, and Xception [27].

C. *Representation*

The feature extraction step generates a feature vector from the training sample signatures. However, a large number of features describing a signature can result in high computational costs for classification and proximity measurements. To address this issue, researchers have proposed various representation schemes for offline signatures, including interval-valued representations [3,72,78], bag-of-visual-words (BoVW) [67,68], cluster-based [73], graph-based [57], sparse-based [113], and Histogram of Templates (HOT) [90]. For offline signatures, an effective representation scheme should account for intra-class variations to minimize computational complexity.

D. Verification

The verification process confirms the authenticity of a test signature sample after matching. Pattern recognition techniques are employed in the matching process to authenticate a signature [18]. During matching, the features of a test signature are compared with those of the reference signatures of the claimed writer stored in a knowledge base. Researchers have proposed various matching techniques for offline signature authentication, including distance-based classifiers [7,34,103], nearest neighbor classifiers [9,50,65], Support Vector Machines (SVMs) [10,11,26,28], neural networks [1,46,47,95,99], and Hidden Markov Models (HMMs) [16,43,44,103]. More recently, deep convolutional neural networks (DCNNs) have been widely explored for verification [25,39,54,55,64,84].

E. Approaches

Various strategies have been proposed in the literature to improve system reliability and reduce error rates. In the literature, two main approaches are commonly used for offline signature verification: writer-independent (WI) and writer-dependent (WD) methods [13,25,28]. In the writer-independent approach, the system is trained once using a common set of features, a common decision threshold, and a shared classifier for all writers. This approach offers advantages such as reduced training time and no impact on system performance when writers are added or removed [17,28, 51,55,74,85,93,112].

In contrast, the writer-dependent approach requires training the system individually for each writer using writer-specific characteristics. While this method is more effective and closely resembles the verification process performed by human experts, it necessitates retraining whenever a new writer is added [32]. A few WD models for offline signature verification can be found in the literature. These models leverage writer dependency at different levels, such as the feature level [12,114] and the classifier level [100]. However, in most cases, the same set of features and classifiers are used for all writers, with writer dependency limited to the decision threshold. Some researchers have also explored hybrid models that combine WI and WD approaches to enhance offline signature verification [8,20,31,87]. Other approaches include fusion-based methods [12,23,80,107,109] and fuzzy-based techniques [24,36,82]. Most of the works in the literature are based on fully supervised learning approaches. However, self-supervised approaches have recently been introduced, which are capable of learning representations from unlabeled datasets [15,60,61].

F. Findings

The literature reveals that researchers have proposed both writer-independent (WI) and writer-dependent (WD) models for offline signature verification. However, most offline signature verification models are WI, employing the same set of features and classifiers for all writers. Conversely, WD models also use the same set of features and classifiers but train the system separately for each writer. In contrast, human experts typically identify different sets of features and apply distinct matching strategies for different writers. This makes WD models more practical as they effectively reduce the risk of fraudulent activities, thereby creating a demand for such systems.

To address these issues, this paper introduces a Interval-oriented feature modeling in writer-dependent of offline signature verification, building on our earlier work [59], where symbolic data concepts were initially employed to represent features. In our previous work [59], we used 150 features to represent a signature and applied fuzzy-C means clustering to aggregate training samples for each writer into a compact representation. In this work, we compute a new set of 70 features. In both studies, writer dependency is incorporated at the levels of features, feature dimensions, and thresholds. Features are represented using an Interval-oriented representation scheme, and verification is performed using a symbolic classifier.

The key contributions of this work include:

- Proposing an offline signature verification system leveraging writer-dependent characteristics.
- Exploring writer dependency at three levels: features, feature dimensions, and decision thresholds.
- Adopting an Interval-oriented representation scheme to effectively capture intra-class variations and employing a symbolic classifier for the verification step.

III. PROPOSED METHOD

The six main stages of the proposed model are shown in Fig. 1.

A. Preprocessing
B. Extraction of features
C. Selection of writer-dependent features
D. Creation of an interval-oriented feature vector
E. Fixation of writer-dependent characteristics
F. Signature verification

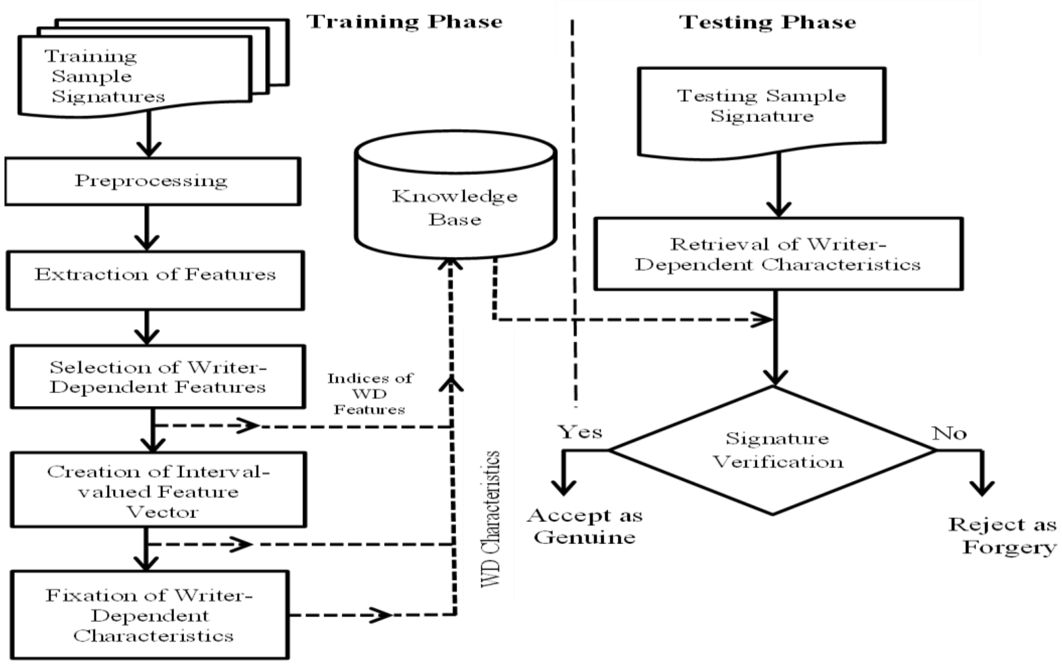

Fig. 1 Block diagram of the proposed model

A. *Preprocessing*

Preprocessing prepares the signature area for the feature extraction stage. The thresholding technique [71] is applied to convert the input signature image into a binary format. Noise in the signature image is reduced using a median filter [102] on the binarized image. Since variations in pen and ink thickness can affect the signature's appearance, morphological operations are employed to minimize these differences. After geometric alignment, the processed image is inverted to reduce computational complexity. The signature area is then cropped to eliminate the surrounding background, and the image is resized to 256 × 256 pixels.

B. *Extraction of features*

Feature extraction is performed after preprocessing the signature images. In this work, we extract both global and grid features, computing a total of 70 features, out of which 25 are global and 45 are grid-based. The details of the global features are provided in Table I. To extract grid features, we apply nine different 3×3 masks, sliding them over the preprocessed image. For each mask, five global statistical features viz., standard deviation, skewness, variance, contrast, and homogeneity are computed. This process results in the extraction of 45 (9 × 5) grid features per signature sample. In total, 70 features are derived from each signature sample.

TABLE I
DETAILS OF GLOBAL FEATURES COMPUTED

Feature	Feature
Signature area [46]	Baseline shift [81]
Centre feature	Variance of row vector [62]
Normalized area [49]	Energy
Minor axis length	Outline signature area [38]
Major axis plus minor axis length	Horizontal centre of the signature [83]
Maximum horizontal projection plus Maximum vertical projection	Vertical centre of the signature [83]
Diameter	Convex area
Radii	Total number of background pixels
Major axis length	Perimeter
Number of edge-points [83]	Equi-diameter
Global centroid	Mean of row vector [62]
Homogeneity [101]	Correlation [101]

Let $\{S_1, S_2, S_3, \dots, S_n\}$ be the set of n training signature samples of writer W_j $(j = 1,2,3, \dots, N)$, where N denotes the number of writers. Let $F_i = \{f_{i1}, f_{i2}, f_{i3}, \dots, f_{im}\}$, where $i = 1,2,3, \dots, n$ be the feature vector representing the i^{th} training signature sample of a writer W_j and m is the total number of features extracted. Hence, feature dimension of each writer is $n \times m$. Therefore, the feature dimension of the entire dataset will be $N \times n \times m$.

C. *Selection of writer-dependent features*

Once the features are extracted, a filter-based feature selection technique is applied to identify writer-dependent features. In this work, we employ the Multi-Cluster Feature Selection (MCFS) [14] technique, which is well-suited for multi-clustered data like signatures, where signatures naturally form clusters for each writer. Additionally, MCFS is a simple and computationally efficient feature selection method. MCFS computes a relevancy score for each feature based on its contribution to preserving the cluster structure of a writer. After calculating these scores, the features are ranked in descending order. The top-ranked features are then selected for verification, with the selected features varying for each writer. This results in the identification of writer-dependent features. For instance, Tables II and III show the indices of the top 10 most relevant features selected by MCFS for the first ten writers in the CEDAR and MCYT datasets, using 50% of the training signature samples. It is evident from these tables that the selected feature indices differ from writer to writer, demonstrating the variability in writer-dependent feature selection.

Let $\{f_1, f_2, f_3, \dots, f_d\}$, $d < m$ be the set of features selected for writer W_j and d is the number of top-ranked features selected with high relevancy scores. To represent each writer, we have recommended d number of features. Therefore, after selecting the writer-dependent features, each writer is represented using a feature matrix of dimension $n \times d$, where n represents the number of training signature samples of the respective writer. Finally, selection of writer-dependent features reduces the feature dimension of entire dataset from $N \times n \times m$ to $N \times n \times d$. The indices of the d features selected are different for different writer, thereby resulting in selection of writer-dependent features.

TABLE II
THE INDICES OF THE TOP 10 FEATURES SELECTED FOR THE FIRST TEN WRITERS OF THE CEDAR DATASET

Writer #	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
1	8	2	46	36	16	57	14	20	15	31
2	8	14	3	57	31	26	21	46	15	40
3	6	26	14	60	31	16	57	40	3	8
4	3	40	26	8	53	10	31	46	16	20
5	2	40	26	46	6	36	14	3	8	31
6	8	3	26	53	40	2	16	57	31	20
7	5	3	57	16	26	40	2	11	6	14
8	60	15	19	14	31	20	16	40	18	26
9	6	3	15	60	14	40	57	5	16	20
10	15	14	8	20	26	40	16	3	6	31

TABLE III
THE INDICES OF THE TOP 10 FEATURES SELECTED FOR THE FIRST TEN WRITERS OF THE MCYT DATASET

Writer #	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
1	3	14	16	57	6	20	36	46	31	26
2	3	5	16	57	46	20	26	14	15	11
3	16	21	3	20	5	60	53	46	36	14
4	3	26	21	5	31	46	40	11	57	60
5	3	14	57	36	16	46	21	53	20	26
6	16	57	31	36	21	53	20	14	12	26
7	26	31	21	14	16	3	46	20	36	6
8	3	40	46	16	53	36	14	26	57	20
9	3	16	19	5	14	31	57	26	11	20
10	3	60	26	53	20	31	36	6	40	16

D. *Creation of an Interval-oriented feature vector for selected writer-dependent features*

Once the writer-dependent features are selected for all writers, each writer is represented using a conventional feature matrix of dimension n × d. Signature databases typically contain samples from multiple writers collected over time. The signatures of each writer form a distinct class, resulting in multiple classes with significant intra-class variations. Interval-oriented representation is more effective in preserving intra-class variations than conventional representation, where each feature is represented as a single crisp value [29]. Therefore, in this work, we adopt an Interval-oriented representation scheme to represent writer-dependent features.

Let $\{f_{i1}, f_{i2}, f_{i3}, \dots, f_{id}\}$, $i = 1,2,3, \dots, n$ be the feature vector representing the i^{th} training signature sample of writer W_j and d is the number of features selected. Interval-oriented feature vector for a writer W_j is created by computing mean and standard deviation of each of the d features selected for all the n training signature samples of writer W_j.

Let $Mean(f_{jk})$; $k = 1,2,3,\ldots,d$ be the mean of the k^{th} feature values obtained from all n training signature samples of writer W_j.

$$\text{i.e.,} \qquad Mean\left(f_{jk}\right) = \frac{1}{n}\sum_{i=1}^{n}(f_{ik}) \qquad (1)$$

Similarly, let $Stdev\left(f_{jk}\right)$; $k = 1,2,3,\ldots,d$ be the standard deviation of the k^{th} feature values obtained from all n training signature samples of writer W_j.

$$\text{i.e.,} \qquad Stdev\left(f_{jk}\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(f_{ik} - Mean\left(f_{jk}\right)\right)^2} \qquad (2)$$

Now, the k^{th} feature values of writer W_j is represented in the form of an interval $\left[f_{jk}^{-}, f_{jk}^{+}\right]$, where $\left[f_{jk}^{-}\right]$ denotes the lower limit of the k^{th} feature values of writer W_j and it is computed as,

$$f_{jk}^{-} = Mean(f_{jk}) - Stdev(f_{jk}) \qquad (3)$$

Similarly, $\left[f_{jk}^{+}\right]$ denotes the upper limit of the k^{th} feature values of writer W_j and it is computed as,

$$f_{jk}^{+} = Mean(f_{jk}) + Stdev(f_{jk}) \qquad (4)$$

This results in creation of an Interval-oriented feature vector for a writer W_j and is given by,

$$F_j = \left\{\left[f_{j1}^{-}, f_{j1}^{+}\right], \left[f_{j2}^{-}, f_{j2}^{+}\right], \left[f_{j3}^{-}, f_{j3}^{+}\right], \ldots, \left[f_{jd}^{-}, f_{jd}^{+}\right]\right\} \qquad (5)$$

In a conventional representation, we need to store the feature vectors of all the training signature samples of each writer. Here, only one Interval-oriented feature vector needs to be stored for each writer instead of feature vectors of all the training samples. This reduces the feature dimension of reference feature vectors to be stored in the knowledgebase from $N \times n \times d$ to $N \times d$ because of N writers, which helps to speed up the matching process.

E. *Fixation of writer-dependent characteristics*

In this work, we utilize writer-dependent characteristics, namely features, feature dimension, and decision threshold. After generating a symbolic Interval-oriented feature vector for each writer, these vectors are stored in the knowledge base along with the indices of selected features for each writer, eliminating the need for feature selection during verification. The False Acceptance Rate (FAR) and False Rejection Rate (FRR) are estimated for each writer by varying the feature dimension at different threshold values. Here, the threshold represents the percentage of features in a test signature that must match the corresponding Interval-oriented features of a reference signature from the claimed writer. After estimating the FAR and FRR for each writer at different feature dimensions under varying thresholds, the Equal Error Rate (EER) is determined using the Receiver Operating Characteristic (ROC) curve. To fix up the writer-dependent characteristics, the minimum EER criterion is applied. Specifically, for each writer, we identify the feature dimension at which the EER is lowest and designate it as the writer-dependent feature dimension. The corresponding threshold is then set as the writer-dependent threshold. This process results in defining writer-dependent characteristics for each writer, including features, feature dimension, and decision threshold. Once determined, these characteristics are stored in the knowledge base for use during verification.

F. *Signature Verification*

To establish the authenticity of a given test signature, writer dependent characteristics fixed for the claimed writer are retrieved from the knowledge base during verification. Then, the features of the test signature are compared with the corresponding interval values of the features of the reference Interval-oriented feature vector of a claimed writer by means of an Interval-oriented classifier. The genuineness of a test signature sample is determined as follows.

Let T_S be a test signature, claimed to be of a writer W_j and it is described by a feature vector $\{f_{t1}, f_{t2}, f_{t3}, \ldots, f_{tm}\}$. During verification, the indices of d features selected for a writer W_j are retrieved from the knowledgebase. Then out of m features only those corresponding d features of test signature T_S are compared with corresponding d Interval-oriented features of the reference signatures. To keep track of the number of features of a test signature fall within the corresponding Interval-oriented features of the respective reference signatures, we use a counter called acceptance count (A_c) [78] and it is defined as

$$A_C = \sum_{k=1}^{d} C\left(f_{tk}, \left[f_{jk}^{-}, f_{jk}^{+}\right]\right) \qquad (6)$$

Where,

$$C\left(f_{tk}, [f_{jk}^{-}, f_{jk}^{+}]\right) = \begin{cases} 1, & if\left(f_{tk} \geq f_{jk}^{-}\ and\ f_{tk} \leq f_{jk}^{+}\right) \\ 0, & otherwise \end{cases} \quad (7)$$

Initially, A_c is set to zero. If a feature of a test signature falls within the corresponding Interval-oriented feature of a reference signature, then value of A_c is incremented by one. After comparing all the features of a test signature, the value of total acceptance count is compared to the predefined similarity threshold computed for the corresponding writer. If it is greater than the predefined similarity threshold, the test signature is accepted as genuine; otherwise, it is rejected as a forgery.

IV. EXPERIMENTATION DETAILS AND RESULTS

In this section, we discuss the dataset utilized for experimentation, training set, testing set details, and the results obtained. The proposed model is evaluated on two benchmarking offline signature datasets: CEDAR [45], and MCYT [70]. The brief description of these datasets are provided in Table IV. For experimentation, the signature dataset is divided into two sets: training and testing.

TABLE IV
THE DESCRIPTION OF CEDAR AND MCYT DATASETS

Dataset	**Total No. of writers**	**No. of genuine signatures per writer**	**No. of skilled forgery signatures per writer**	**Total No. of signatures**
CEDAR	55	24	24	2640
MCYT	75	15	15	2250

The training set consists only genuine signatures, while the testing set includes all random and skilled forgeries along with the remaining genuine signatures. We have used three performance metrics to evaluate the performance of the proposed approach, viz., the FRR, the FAR, and the EER. Experimentation is conducted by varying the percentage of training samples from 30% to 70%. Each time, the training samples are selected randomly to effectively capture intra-class variations. Skilled forgeries refer to forgeries created by professionals with extensive practice. Random forgeries are generated for each writer by selecting one genuine signature from every other writer in the system for testing purposes. In this work, we performed two sets of experiments. The first experiment is conducted without selecting writer-dependent features, using a common set of features and a global threshold for all writers. The second experiment incorporates writer-dependent features.

In the first set of experiments, the training signature samples of each writer are represented using the same number of features. These features are then transformed into an Interval-oriented vector, as explained in Section 3.4. For each writer, FAR and FRR are calculated by varying the percentage of training samples from 30% to 70% and adjusting the decision threshold from 0.1 to 1.0. The EER is estimated for each percentage of training and testing samples, and the threshold that results in the lowest EER is set as the global threshold for all writers. Finally, the system's overall EER is determined by averaging the EER values of all writers. We conducted 10 trials using different training and testing samples. Table V presents the maximum, minimum, and average EER obtained from these trials when a common set of features and a global threshold are used for all writers in the CEDAR and MCYT datasets.

TABLE V
THE MAXIMUM, MINIMUM, AND AVERAGE VALUES OF EER OBTAINED USING A COMMON SET OF FEATURES AND GLOBAL THRESHOLDS ON CEDAR, AND MCYT

Dataset	**% of Training Samples**	With Skilled Forgeries, Threshold = 0.5			With Random Forgeries, Threshold = 0.4		
		Max. EER	Min. EER	Avg. EER	Max. EER	Min. EER	Avg. EER
CEDAR	**30%**	18.52	16.53	17.53	9.21	8.72	8.96
	40%	17.41	15.29	16.35	8.45	7.11	7.78
	50%	16.85	15.14	16.00	7.86	6.26	7.06
	60%	14.77	13.97	14.37	6.18	5.24	5.71
	70%	13.51	12.01	12.76	5.05	4.09	4.57
MCYT	**30%**	19.49	18.03	18.76	9.11	8.12	8.61
	40%	17.23	13.71	15.47	8.75	7.24	7.99
	50%	15.32	12.65	13.99	7.86	6.26	7.06
	60%	14.69	12.17	13.43	6.64	5.74	6.19
	70%	13.98	11.61	12.80	7.43	5.11	6.27

The minimum EER obtained using a common feature set and a global threshold for CEDAR, and MCYT datasets are given in Table VI. The second set of experimentation is carried out using writer-dependent features instead of using the same number of features and a fixed threshold for all writers. For each writer, writer-dependent features are selected by varying the feature dimensions from 5 to 50 in steps of 5 and threshold values from 0.1 to 1.0 in steps of 0.1, using the MCFS feature selection technique, as discussed in Section 3.3. After selecting the writer-dependent features, the training samples of each writer are represented as an Interval-oriented feature vector, as explained in Subsection 3.4.

In the next step, we estimate the FAR and FRR for each writer at each feature dimension, with thresholds ranging from 0.1 to 1.0. Then, the EER is estimated for each writer using the ROC curve at each feature dimension. The feature dimension at which the EER is minimized is set as the writer-dependent feature dimension, and the corresponding threshold is set as the writer-dependent threshold for the respective writer. The writer-dependent characteristics viz., features, feature dimension, and decision threshold are then fixed, as explained in Section 3.5.

TABLE VI
MINIMUM EER OBTAINED USING A COMMON FEATURE SET AND A GLOBAL THRESHOLD FOR CEDAR, AND MCYT DATASETS

Dataset	**CEDAR**		**MCYT**	
% of Training Samples	With Skilled Forgeries Threshold = 0.5	With Random Forgeries Threshold =0.4	With Skilled Forgeries Threshold =0.5	With Random Forgeries Threshold =0.4
30%	16.53	8.72	18.03	8.12
40%	15.29	7.11	13.71	7.24
50%	15.14	6.26	12.65	6.26
60%	13.97	5.24	12.17	5.74
70%	12.01	4.09	11.61	5.11

To determine the authenticity of a test signature, we use the acceptance count, which is calculated for each writer, as explained in Section 3.6. The EER is determined for each writer based on their unique characteristics, such as features, feature dimension, and threshold. Finally, we calculate the EER of the entire system by averaging the EER of all writers. At the verification step, both skilled and random forgeries are tested. Experimentation is repeated for ten trials with different training and testing samples, which are selected randomly in each trial. Table VII shows the maximum, minimum, and average EER obtained from all ten trials using writer-dependent features for the CEDAR and MCYT datasets. In this second set of experiments, the threshold and feature dimensions vary from writer to writer. Hence, they are not shown in Table VII. The minimum EER obtained for the CEDAR and MCYT datasets using writer-dependent characteristics is presented in Table VIII.

TABLE VII
THE MAXIMUM, MINIMUM, AND AVERAGE VALUES OF EER OBTAINED USING WRITER DEPENDENT CHARACTERISTICS ON CEDAR, AND MCYT DATASETS

Dataset	**% of Training Samples**	With Skilled Forgeries			With Random Forgeries		
		Max. EER	Min. EER	Avg. EER	Max. EER	Min. EER	Avg. EER
CEDAR	**30%**	14.72	13.3	14.01	6.86	5.61	6.23
	40%	13.79	12.48	13.13	6.72	5.3	6.01
	50%	12.23	11.05	11.64	5.48	4.98	5.23
	60%	11.97	9.14	10.55	4.73	3.99	4.36
	70%	10.17	8.56	09.36	4.24	3.32	3.78
MCYT	**30%**	15.21	13.04	14.12	6.42	5.71	6.06
	40%	14.84	12.36	13.60	5.74	4.02	4.88
	50%	13.98	11.79	12.88	4.60	3.71	4.155
	60%	12.68	11.06	11.87	3.45	2.39	2.92
	70%	10.76	9.25	10.05	3.21	2.02	2.61

TABLE VIII
MINIMUM EER OBTAINED USING WRITER DEPENDENT CHARACTERISTICS FOR CEDAR, AND MCYT DATASETS

Dataset	**CEDAR**		**MCYT**	
% of Training Samples	With Skilled Forgeries	With Random Forgeries	With Skilled Forgeries	With Random Forgeries
30%	13.3	5.61	13.04	5.71
40%	12.48	5.3	12.36	4.02
50%	11.05	4.98	11.79	3.71
60%	9.14	3.99	11.06	2.39
70%	8.56	3.32	9.25	2.02

To compare the performance of the model with and without writer-dependent characteristics, the minimum EER obtained on the CEDAR and MCYT datasets is presented in Table IX. From Table IX, it can be observed that the error rate decreases when the model is trained with a sufficient number of training samples. However, in the case of an offline signature verification system, as the number of writers increases, the computational complexity also increases. This is due to high intra-class variations and low inter-class variations. The graphical representations of the results obtained with skilled forgeries on the CEDAR and MCYT datasets, as presented in Table IX, are shown in Fig. 2 and Fig. 3, respectively.

TABLE IX
THE EER OBTAINED BY USING WRITER DEPENDENT CHARACTERISTICS AND WITH USING A COMMON SET OF FEATURES AND THRESHOLDS FOR CEDAR, AND MCYT DATASETS

Experiment	Dataset	**CEDAR**		**MCYT**	
	% of Training Samples	With Skilled Forgeries	With Random Forgeries	With Skilled Forgeries	With Random Forgeries
With Writer Dependent Characteristics	**30%**	13.3	5.61	13.04	5.71
	40%	12.48	5.3	12.36	4.02
	50%	11.05	4.98	11.79	3.71
	60%	9.14	3.99	11.06	2.39
	70%	8.56	3.32	9.25	2.02
With Common Features and Threshold	**30%**	16.53	8.72	18.03	8.12
	40%	15.29	7.11	13.71	7.24
	50%	15.14	6.26	12.65	6.26
	60%	13.97	5.24	12.17	5.74
	70%	12.01	4.09	11.61	5.11

From Table IX, it can be observed that the error rate decreases when the model is trained with a sufficient number of training samples. However, in the case of an offline signature verification system, as the number of writers increases, the computational complexity also rises. This is due to high intra-class variations and low inter-class variations.

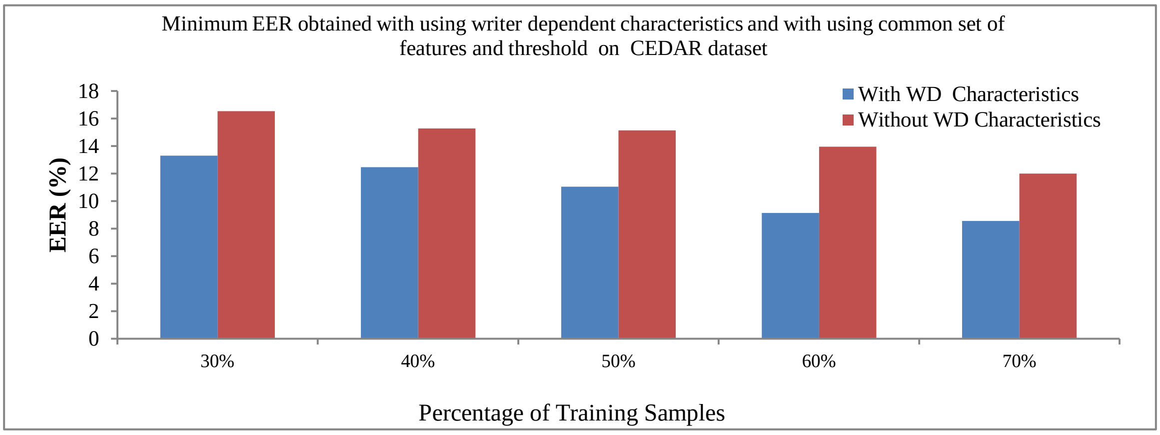

Fig. 2 Minimum EER obtained with using writer-dependent characteristics and with using common set of features and threshold on CEDAR dataset

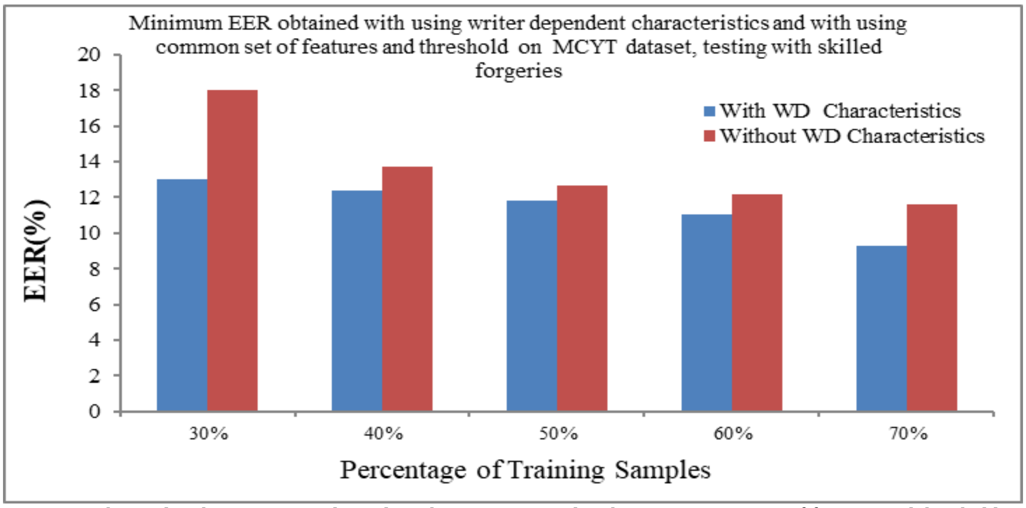

Fig. 3 Minimum EER obtained with using writer-dependent characteristics and with using common set of features and threshold on MCYT dataset

From Figures 2 and 3, it is clear that the EER obtained using writer-dependent characteristics is lower than the EER obtained with a common set of features and a global threshold across all splits of training and testing, as well as for both skilled and random forgeries, across all three offline signature datasets. This demonstrates the significance of writer-dependent features for offline signature verification systems.

V. Comparative Analysis

In this section, we compare the performance of the proposed model with other state-of-the-art approaches reported in the literature on the CEDAR and MCYT datasets. Comparing offline signature verification models is challenging due to variations in feature extraction and classification techniques, differences in training and testing sample sizes, performance measures, and other factors. Therefore, we limit our comparative study to models that have been validated on the CEDAR and MCYT datasets.

The EER of various models tested on the CEDAR and MCYT datasets is presented in Table X and Table XI, respectively. However, some entries in these tables are marked with a (-) because the authors did not report the number of training samples used to obtain those results. Table X shows that, compared to the other models listed, our proposed model achieves a lower EER than models [45,98] but a higher EER than the remaining models. This is because our approach relies on conventional machine learning, utilizing handcrafted features and a traditional classifier. In contrast, models [25,57,85,105] are based on convolutional neural networks, while models [11,52,92] use high-dimensional feature vectors, which contribute to their improved performance.

TABLE X
Comparison of the performances of various approaches on the CEDAR dataset

Model	No. of training samples	EER/AER
[98]	24	21.50
[45]	-	21.90
[105]	-	08.50
[53]	24	08.33
[33]	04	08.27
[11]	12	07.84
[85]	-	06.75
[52]	16	06.02
[57]	10	05.91
[28]	12	05.60
[91]	-	04.94
[25]	-	04.69
[92]	12	04.67
Proposed	**17**	**08.56**

TABLE XI
COMPARISON OF THE PERFORMANCES OF VARIOUS APPROACHES ON THE MCYT DATASET

Model	**No. of training samples**	**EER/AER**
[04]	10	22.13
[79]	-	19.80
[78]	-	17.61
[108]	12	15.41
[97]	05	13.44
[33]	10	12.44
[58]	10	12.01
[41]	-	11.51
[69]	10	09.87
[12]	10	09.26
[58]	10	09.15
[101]	10	07.08
[92]	10	05.96
[25]	-	05.58
Proposed	**10**	**9.25**

Based on the results presented in Table XI, it is clear that our proposed model has a lower EER than the other models presented in [04,12,33,58,69,78,79,97,108]. However, our model has a higher EER compared to models [25,58], which are based on deep neural networks, and model [69], which is a hybrid approach utilizing principal component analysis (PCA), discrete radon transform (DRT), and probabilistic neural network (PNN). Additionally, models [33,92] employ high-dimensional feature vectors. Nevertheless, our model is conventional, straightforward, and easy to use.

VI. FUTURE WORKS

In this work, we propose an offline signature verification system utilizing writer-dependent handcrafted features. In the future, we aim to dynamically exploit writer-dependent characteristics using deep features.

VII. CONCLUSION

This paper explores the applicability of writer-dependent characteristics for offline signature verification. Writer dependency is utilized at three levels: features, feature dimensions, and decision-thresholds. Writer-dependent features are selected using a filter-based feature selection technique. An Interval-oriented symbolic representation is adopted to represent these features, effectively capturing intra-class variations. A symbolic classifier is employed to establish the authenticity of a test signature. Extensive experiments are conducted on two standard benchmarking offline signature datasets, CEDAR and MCYT. Two sets of experiments are performed to demonstrate the effectiveness of writer-dependent characteristics. The first set uses a common feature set and a global threshold, while the second set employs writer-dependent features, dimensions, and decision thresholds. The experimental results highlight the effectiveness of writer-dependent characteristics in offline signature verification.

Conflicts of Interest

The authors have no conflicts of interest to declare.

Funding

No funding is provided for the preparation of manuscripts

REFERENCES

[1] Abdala, M. A., and Yousif, N. A. (2009). Offline signature recognition and verification based on artificial neural network. Eng and Tech. Journal, 27(7), 20.

[2] AbdelRaouf, A., and Salama, D. (2018). Handwritten signature verification using haar cascade classifier approach. 2018 13th International Conference on Computer Engineering and Systems (ICCES), 319–326.

[3] Alaei, A., Pal, S., Pal, U., and Blumenstein, M. (2017). An Efficient Signature Verification Method Based on an Interval Symbolic Representation and a Fuzzy Similarity Measure. IEEE Transactions on Information Forensics and Security, 12(10), 2360–2372. https://doi.org/10.1109/TIFS.2017.2707332

[4] Alonso-Fernandez, F., Fairhurst, M. C., Fierrez, J., and Ortega-Garcia, J. (2007). Automatic measures for predicting performance in off-line signature. 2007 IEEE International Conference on Image Processing, 1, I–369.

[5] Alsuhimat, F. M., and Mohamad, F. S. (2023). Offline signature verification using long short-term memory and histogram orientation gradient. Bulletin of Electrical Engineering and Informatics, 12(1), 283–292.

[6] Alvarez, G., Bryant, M., Sheffer, B., and Bryant, M. (2016). Offline signature verification with convolutional neural networks. Technical Report, Stanford University, 8.

[7] Arathi, M., and Govardhan, A. (2014). An efficient offline signature verification system. International Journal of Machine Learning and Computing, 4(6), 533.

[8] Avola, D., Bigdello, M. J., Cinque, L., Fagioli, A., and Marini, M. R. (2021). R-SigNet: Reduced space writer-independent feature learning for offline writer-dependent signature verification. Pattern Recognition Letters, 150, 189–196. https://doi.org/10.1016/j.patrec.2021.06.033

[9] Azmi, A. N., Nasien, D., and Omar, F. S. (2017). Biometric signature verification system based on freeman chain code and k-nearest neighbor. Multimedia Tools and Applications, 76, 15341–15355.

[10] Batool, F. E., Attique, M., Sharif, M., Javed, K., Nazir, M., Abbasi, A. A., Iqbal, Z., and Riaz, N. (2020). Offline signature verification system: a novel technique of fusion of GLCM and geometric features using SVM. Multimedia Tools and Applications, 1–20.

[11] Bharathi, R. K., and Shekar, B. H. (2013). Off-line signature verification based on chain code histogram and Support Vector Machine. Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013, 2063–2068.

[12] Bhunia, A. K., Alaei, A., and Roy, P. P. (2019). Signature verification approach using fusion of hybrid texture features. Neural Computing and Applications, 31(12), 8737–8748. https://doi.org/10.1007/s00521-019-04220-x

[13] Bouamra, W., Djeddi, C., Nini, B., Diaz, M., and Siddiqi, I. (2018). Towards the design of an offline signature verifier based on a small number of genuine samples for training. Expert Systems with Applications, 107, 182–195. https://doi.org/10.1016/j.eswa.2018.04.035

[14] Cai, D., Zhang, C., and He, X. (2010). Unsupervised feature selection for multi-cluster data. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 333–342.

[15] Chattopadhyay, S., Manna, S., Bhattacharya, S., and Pal, U. (2022). Surds: Self-supervised attention-guided reconstruction and dual triplet loss for writer independent offline signature verification. 2022 26th International Conference on Pattern Recognition (ICPR), 1600–1606.

[16] Coetzer, J., Herbst, B. M., and Du Preez, J. A. (2004). Offline signature verification using the discrete radon transform and a hidden Markov model. Eurasip Journal on Applied Signal Processing, 2004(4), 559–571. https://doi.org/10.1155/S1110865704309042

[17] Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Llados, J., and Pal, U. (2017). SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification. 1, 1–7. http://arxiv.org/abs/1707.02131

[18] Diaz, M., Ferrer, M. A., Impedovo, D., Malik, M. I., Pirlo, G., and Plamondon, R. (2019). A perspective analysis of handwritten signature technology. ACM Computing Surveys, 51(6). https://doi.org/10.1145/3274658

[19] Engin, D., Kantarci, A., Arslan, S., and Ekenel, H. K. (2020). Offline signature verification on real-world documents. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 808–809.

[20] Eskander, G. S., Sabourin, R., and Granger, E. (2013). Hybrid writer-independent writer-dependent offline signature verification system. IET Biometrics, 2(4), 169–181. https://doi.org/10.1049/iet-bmt.2013.0024

[21] Ferrer, M. A., Alonso, J. B., and Travieso, C. M. (2005). Offline geometric parameters for automatic signature verification using fixed-point arithmetic. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6), 993–997. https://doi.org/10.1109/TPAMI.2005.125

[22] Fick, C., Coetzer, J., and Swanepoel, J. (2017). Efficient curve-sensitive features for offline signature verification. 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference, PRASA-RobMech 2016, 1–6.

[23] Fierrez-Aguilar, J., Alonso-Hermira, N., Moreno-Marquez, G., and Ortega-Garcia, J. (2004). An off-line signature verification system based on fusion of local and global information. Biometric Authentication: ECCV 2004 International Workshop, BioAW 2004, Prague, Czech Republic, May 15th, 2004. Proceedings, 295–306.

[24] Ganapathi, G., and Nadarajan, R. (2013). A fuzzy hybrid framework for offline signature verification. Pattern Recognition and Machine Intelligence: 5th International Conference, PReMI 2013, Kolkata, India, December 10-14, 2013. Proceedings 5, 121–127.

[25] 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://doi.org/10.1016/j.eswa.2020.114249

[26] Goon, L. W., and Eng, S. K. (2021). Offline Signature Verification System Using SVM Classifier with Image Pre-processing Steps and SURF Algorithm. Journal of Physics: Conference Series, 2107(1), 12069.

[27] Gosai, D., Vyas, S., Patel, S., Barot, P., and Suthar, K. (2022). Handwritten Signature Verification Using Convolution Neural Network (CNN). International Conference on Advancements in Smart Computing and Information Security, 90–106.

[28] Guerbai, Y., Chibani, Y., and Hadjadji, B. (2015). The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters. Pattern Recognition, 48(1), 103–113. https://doi.org/10.1016/j.patcog.2014.07.016

[29] Guru, D. S., and Nagendraswamy, H. S. (2007). Symbolic representation of two-dimensional shapes. Pattern Recognition Letters, 28(1), 144–155.

[30] Hadjadji, B., Chibani, Y., and Nemmour, H. (2017). An efficient open system for offline handwritten signature identification based on curvelet transform and one-class principal component analysis. Neurocomputing, 265, 66–77. https://doi.org/10.1016/j.neucom.2017.01.108

[31] Hafemann, L. G., Sabourin, R., and Oliveira, L. S. (2017). Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recognition, 70, 163–176. https://doi.org/10.1016/j.patcog.2017.05.012

[32] Hafemann, L. G., Sabourin, R., and Oliveira, L. S. (2018). Offline handwritten signature verification - Literature review. Proceedings of the 7th International Conference on Image Processing Theory, Tools and Applications, IPTA 2017, 2018-Janua, 1–8.

https://doi.org/10.1109/IPTA.2017.8310112

[33] Hafemann, L. G., Sabourin, R., and Oliveira, L. S. (2019). Meta-learning for fast classifier adaptation to new users of signature verification systems. IEEE Transactions on Information Forensics and Security, 15, 1735–1745.

[34] Hamadene, A., and Chibani, Y. (2016). One-Class Writer-Independent Offline Signature Verification Using Feature Dissimilarity Thresholding. IEEE Transactions on Information Forensics and Security, 11(6), 1226–1238. https://doi.org/10.1109/TIFS.2016.2521611

[35] Hameed, M. M., Ahmad, R., Mat, L., and Murtaza, G. (2021). Signal Processing : Image Communication Machine learning-based offline signature verification systems : A systematic review. Signal Processing: Image Communication, 93(April 2020), 116139. https://doi.org/10.1016/j.image.2021.116139

[36] Hanmandlu, M., Yusof, M. H. M., and Madasu, V. K. (2005). Off-line signature verification and forgery detection using fuzzy modeling. Pattern Recognition, 38(3), 341–356. https://doi.org/10.1016/j.patcog.2004.05.015

[37] Houtinezhad, M., and Ghaffari, H. R. (2022). Off-line signature verification system using features linear mapping in the candidate points. Multimedia Tools and Applications, 81(17), 24815–24847.

[38] Huang, K., and Yan, H. (1997). Off-line signature verification based on geometric feature extraction and neural network classification. Pattern Recognition, 30(1), 9–17. https://doi.org/10.1016/S0031-3203(96)00063-5

[39] Hung, P. D., Bach, P. S., Vinh, B. T., Tien, N. H., and Diep, V. T. (2022). Offline handwritten signature forgery verification using deep learning methods. In Smart Trends in Computing and Communications: Proceedings of SmartCom 2022 (pp. 75–84). Springer.

[40] Impedovo, D., and Pirlo, G. (2008). Automatic signature verification: The state of the art. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 38(5), 609–635. https://doi.org/10.1109/TSMCC.2008.923866

[41] Jagtap, A. B., Sawat, D. D., Hegadi, R. S., and Hegadi, R. S. (2020). Verification of genuine and forged offline signatures using Siamese Neural Network (SNN). Multimedia Tools and Applications, 79(47–48), 35109–35123. https://doi.org/10.1007/s11042-020-08857-y

[42] Jain, A., Singh, S. K., and Singh, K. P. (2021). Signature verification using geometrical features and artificial neural network classifier. Neural Computing and Applications, 33(12), 6999–7010. https://doi.org/10.1007/s00521-020-05473-7

[43] Justino, E. J. R., Bortolozzi, F., and Sabourin, R. (2001). Off-line signature verification using HMM for Random, simple and skilled forgeries. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2001-Janua(c), 1031–1034. https://doi.org/10.1109/ICDAR.2001.953942

[44] Justino, E. J. R., El Yacoubi, A., Bortolozzi, F., and Sabourin, R. (2000). An off-line signature verification system using HMM and graphometric features. Proc. of the 4th International Workshop on Document Analysis Systems, 211–222.

[45] Kalera, M. K., Srihari, S., and Xu, A. (2004). Offline Signature Verification and Identification. International Journal of Pattern Recognition and Artificial Intelligence, 18(7), 1339–1360.

[46] Karouni, A., Daya, B., and Bahlak, S. (2011). Offline signature recognition using neural networks approach. Procedia Computer Science, 3, 155–161. https://doi.org/10.1016/j.procs.2010.12.027

[47] Kiran, P., Parameshachari, B. D., Yashwanth, J., and Bharath, K. N. (2021). Offline signature recognition using image processing techniques and back propagation neuron network system. SN Computer Science, 2(3), 196.

[48] Kovari, B., and Charaf, H. (2013). A study on the consistency and significance of local features in off-line signature verification. Pattern Recognition Letters, 34(3), 247–255. https://doi.org/10.1016/j.patrec.2012.10.011

[49] Kruthi, C., and Shet, D. C. (2014). Offline signature verification using support vector machine. Proceedings - 2014 5th International Conference on Signal and Image Processing, ICSIP 2014, 3–8. https://doi.org/10.1109/ICSIP.2014.5

[50] Kumar, A., and Bhatia, K. (2021). k-NN based writer independent offline signature verification system. 2021 International Conference on Technological Advancements and Innovations (ICTAI), 612–616.

[51] Kumar, A., and Rastogi, R. (2022). Writer-Independent Offline Signature Verification using LBP and NN. 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), 1, 776–779.

[52] Kumar, M. M., and Puhan, N. B. (2014). Off-line Signature verification: Upper and lower envelope shape Analysis using chord moments. IET Biometrics, 3(4), 347–354. https://doi.org/10.1049/iet-bmt.2014.0024

[53] Kumar, R., Sharma, J. D., and Chanda, B. (2012). Writer-independent off-line signature verification using surroundedness feature. Pattern Recognition Letters, 33(3), 301–308. https://doi.org/10.1016/j.patrec.2011.10.009

[54] Liao, W., Lin, Z., Chen, Z., Zou, G., and Chen, L. (2023). Research on offline signature verification based on image domain transfer Siamese network. International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2023), 12707, 50–57.

[55] Longjam, T., Kisku, D. R., and Gupta, P. (2023). Multi-scripted Writer Independent Off-line Signature Verification using Convolutional Neural Network. Multimedia Tools and Applications, 82(4), 5839–5856.

[56] Maergner, P., Pondenkandath, V., Alberti, M., Liwicki, M., Riesen, K., Ingold, R., and Fischer, A. (2018). Offline signature verification by combining graph edit distance and triplet networks. Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, S+ SSPR 2018, Beijing, China, August 17–19, 2018, Proceedings 9, 470–480.

[57] Maergner, P., Pondenkandath, V., Alberti, M., Liwicki, M., Riesen, K., Ingold, R., and Fischer, A. (2019). Combining graph edit distance and triplet networks for offline signature verification. Pattern Recognition Letters, 125, 527–533. https://doi.org/10.1016/j.patrec.2019.06.024

[58] Maergner, P., Riesen, K., Ingold, R., and Fischer, A. (2017). A structural approach to offline signature verification using graph edit distance. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 1, 1216–1222.

[59] Manjunatha, K. S., Guru, D. S., and Annapurna, H. (2017). Interval-valued writer-dependent global features for off-line signature verification. In Mining Intelligence and Knowledge Exploration: 5th International Conference, MIKE 2017, Hyderabad, India, December 13–15, 2017, Proceedings 5 (pp. 133-143). Springer International Publishing.

[60] Manna, S., Chattopadhyay, S., Bhattacharya, S., and Pal, U. (2022). SWIS: Self-Supervised Representation Learning for Writer Independent Offline Signature Verification. 1, 1411–1415. https://doi.org/10.1109/icip46576.2022.9897562
[61] Mayda, M. F., and Musdholifah, A. (n.d.). Siamese-Network Based Signature Verification using Self Supervised Learning. IJCCS (Indonesian Journal of Computing and Cybernetics Systems), 17(2).
[62] Mhatre, P. M., and Maniroja, M. (2011). Offline signature verification based on statistical features. International Conference and Workshop on Emerging Trends in Technology 2011, ICWET 2011 - Conference Proceedings, Icwet, 59–62. https://doi.org/10.1145/1980022.1980034
[63] Mizukami, Y., Yoshimura, M., Miike, H., and Yoshimura, I. (2002). An off-line signature verification system using an extracted displacement function. Pattern Recognition Letters, 23(13), 1569–1577. https://doi.org/10.1016/S0167-8655(02)00121-6
[64] Mosaher, Q. S., and Hasan, M. (2022). Offline handwritten signature recognition using deep convolution neural network. European Journal of Engineering and Technology Research, 7(4), 44–47.
[65] Murtopo, A. A., Priyatna, B., and Mayasari, R. (2022). Signature Verification Using The K-Nearest Neighbor (KNN) Algorithm and Using the Harris Corner Detector Feature Extraction Method. Buana Information Technology and Computer Sciences (BIT and CS), 3(2), 35–40.
[66] Noor, F., Mohamed, A. E., Ahmed, F. A. S., and Taha, S. K. (2020). Offline Handwritten Signature Recognition using Convolutional Neural Network Approach. 2020 International Conference on Computing, Networking, Telecommunications and Engineering Sciences Applications (CoNTESA), 51–57.
[67] Okawa, M. (2016). Offline signature verification based on bag-of-visual words model using KAZE features and weighting schemes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 184–190.
[68] Okawa, M. (2018). From BoVW to VLAD with KAZE features: Offline signature verification considering cognitive processes of forensic experts. Pattern Recognition Letters, 113, 75–82. https://doi.org/10.1016/j.patrec.2018.05.019
[69] Ooi, S. Y., Teoh, A. B. J., Pang, Y. H., and Hiew, B. Y. (2016). Image-based handwritten signature verification using hybrid methods of discrete Radon transform, principal component analysis and probabilistic neural network. Applied Soft Computing Journal, 40, 274–282. https://doi.org/10.1016/j.asoc.2015.11.039
[70] Ortega-Garcia, J., Fierrez-Aguilar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., Satue, A., Hernaez, I., Igarza, J.-J., and Vivaracho, C. (2003). MCYT baseline corpus: a bimodal biometric database. IEE Proceedings-Vision, Image and Signal Processing, 150(6), 395–401.
[71] Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62–66.
[72] Pal, S., Alaei, A., Pal, U., and Blumenstein, M. (2015). Interval-valued symbolic representation based method for off-line signature verification. Proceedings of the International Joint Conference on Neural Networks, 2015-September. https://doi.org/10.1109/IJCNN.2015.7280518
[73] Pandya, V. (2019). Offline Signature Verification using Clustering Technique. Proceedings of the 2019 the International Conference on Pattern Recognition and Artificial Intelligence, 92–95.
[74] Parcham, E., Ilbeygi, M., and 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(July), 115649. https://doi.org/10.1016/j.eswa.2021.115649
[75] Parodi, M., Gómez, J. C., and Belaïd, A. (2011). A circular grid-based rotation invariant feature extraction approach for off-line signature verification. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 1289–1293. https://doi.org/10.1109/ICDAR.2011.259
[76] Parziale, A., Diaz, M., Ferrer, M. A., and Marcelli, A. (2019). SM-DTW: Stability Modulated Dynamic Time Warping for signature verification. Pattern Recognition Letters, 121, 113–122. https://doi.org/10.1016/j.patrec.2018.07.029
[77] Pourshahabi, M. R., Sigari, M. H., and Pourreza, H. R. (2009). Offline handwritten signature identification and verification using contourlet transform. SoCPaR 2009 - Soft Computing and Pattern Recognition, 670–673. https://doi.org/10.1109/SoCPaR.2009.132
[78] Prakash, H. N., and Guru, D. S. (2009a). Bi-Interval Valued Symbolic Feature Vector for Off-line Signature Verification: A Fusion Based Approach. IICAI, 1887–1900.
[79] Prakash, H. N., and Guru, D. S. (2009b). Relative orientations of geometric centroids for off-line signature verification. Proceedings of the 7th International Conference on Advances in Pattern Recognition, ICAPR 2009, 201–204. https://doi.org/10.1109/ICAPR.2009.28
[80] Prakash, H. N., and Guru, D. S. (2010). Offline signature verification: An approach based on score level fusion. International Journal of Computer Applications, 1(18), 52–58.
[81] Qi, Y., and Hunt, B. R. (1994). Signature verification using global and grid features. Pattern Recognition, 27(12), 1621–1629.
[82] Quek, C., and Zhou, R. W. (2002). Antiforgery: a novel pseudo-outer product based fuzzy neural network driven signature verification system. Pattern Recognition Letters, 23(14), 1795–1816.
[83] Ramachandra, A. C., Rao, J. S., Raja, K. B., Venugopla, K. R., and Patnaik, L. M. (2009). Robust offline signature verification based On global features. 2009 IEEE International Advance Computing Conference, IACC 2009, March, 1173–1178. https://doi.org/10.1109/IADCC.2009.4809181
[84] Rateria, A., and Agarwal, S. (2018). Off-line signature verification through machine learning. 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 1–7.
[85] Ren, C., Zhang, J., Wang, H., and Shen, S. (2023). Vision Graph Convolutional Network for Writer-Independent Offline Signature Verification. 2023 International Joint Conference on Neural Networks (IJCNN), 1–7.
[86] Rohilla, S., Sharma, A., and Singla, R. K. (2014). Online Signature Verification at Sub-trajectory Level BT - Advanced Computing, Networking and Informatics- Volume 2 (M. Kumar Kundu, D. P. Mohapatra, A. Konar, and A. Chakraborty (eds.); pp. 369–374). Springer International Publishing.

[87] Rokade, S., Singh, S. K., Bansod, S., and Pal, P. (2023). An Offline Signature Verification Using Deep Convolutional Neural Networks. 2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 1–4.

[88] Ruiz-Del-Solar, J., Devia, C., Loncomilla, P., and Concha, F. (2008). Offline signature verification using local interest points and descriptors. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5197 LNCS, 22–29. https://doi.org/10.1007/978-3-540-85920-8_3

[89] Schafer, B., and Viriri, S. (2009). An off-line signature verification system. ICSIPA09 - 2009 IEEE International Conference on Signal and Image Processing Applications, Conference Proceedings, 95–100. https://doi.org/10.1109/ICSIPA.2009.5478727

[90] Serdouk, Y., Nemmour, H., and Chibani, Y. (2017). Handwritten signature verification using the quad-tree histogram of templates and a support vector-based artificial immune classification. Image and Vision Computing, 66, 26–35.

[91] Shariatmadari, S., Emadi, S., and Akbari, Y. (2019). Patch-based offline signature verification using one-class hierarchical deep learning. International Journal on Document Analysis and Recognition, 22(4), 375–385. https://doi.org/10.1007/s10032-019-00331-2

[92] Sharif, M., Khan, M. A., Faisal, M., Yasmin, M., and Fernandes, S. L. (2020). A framework for offline signature verification system: Best features selection approach. Pattern Recognition Letters, 139(February), 50–59. https://doi.org/10.1016/j.patrec.2018.01.021

[93] Sharma, N., Gupta, S., Mohamed, H. G., Anand, D., Mazón, J. L. V., Gupta, D., and Goyal, N. (2022). Siamese convolutional neural network-based twin structure model for independent offline signature verification. Sustainability, 14(18), 11484.

[94] Sharma, P. (2014). Offline signature verification using surf feature extraction and neural networks approach. Int J Comput Sci Inf Technol, 5(3), 3539–3541.

[95] Shekar, B. H., and Bharathi, R. K. (2014). Off-line signature verification based on principal component analysis and multi-layer perceptrons. Recent Advances in Intelligent Informatics: Proceedings of the Second International Symposium on Intelligent Informatics (ISI'13), August 23-24 2013, Mysore, India, 101–109.

[96] Sigari, M. H., Pourshahabi, M. R., and Pourreza, H. R. (2012). An ensemble classifier approach for static signature verification based on multi-resolution extracted features. International Journal of Signal Processing, Image Processing and Pattern Recognition, 5(1), 21–36.

[97] Soleimani, A., Araabi, B. N., and Fouladi, K. (2016). Deep Multitask Metric Learning for Offline Signature Verification. Pattern Recognition Letters, 80, 84–90. https://doi.org/10.1016/j.patrec.2016.05.023

[98] Srihari, S. N., Xu, A., and Kalera, M. K. (2004). Learning strategies and classification methods for off-line signature verification. Proceedings - International Workshop on Frontiers in Handwriting Recognition, IWFHR, 161–166. https://doi.org/10.1109/IWFHR.2004.61

[99] Tahir, N. M. T., Ausat, A. N., Bature, U. I., Abubakar, K. A., and Gambo, I. (2021). Off-line Handwritten Signature Verification System: Artificial Neural Network Approach. International Journal of Intelligent Systems and Applications, 13(1), 45–57. https://doi.org/10.5815/ijisa.2021.01.04

[100] Tanko, O., Senkyire, I. B., Kwao, K., Aluze-Ele, S. H. A., Boahen, E. K., and Frimpong, S. A. (2023). A Writer-Dependent Approach for Offline Signature Verification Using Feature Learning and One-Class Support Vector Machine. 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 1–6.

[101] Vargas, J. F., Ferrer, M. A., Travieso, C. M., and Alonso, J. B. (2011). Off-line signature verification based on grey level information using texture features. Pattern Recognition, 44(2), 375–385. https://doi.org/10.1016/j.patcog.2010.07.028

[102] Wang, Z., and Zhang, D. (1999). Progressive switching median filter for the removal of impulse noise from highly corrupted images. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 46(1), 78–80.

[103] Wen, J., Fang, B., Tang, Y. Y., and Zhang, T. P. (2009). Model-based signature verification with rotation invariant features. Pattern Recognition, 42(7), 1458–1466. https://doi.org/10.1016/j.patcog.2008.10.006

[104] Xamxidin, N., Yao, Z., Aysa, A., and Ubul, K. (2022). Multilingual Offline Signature Verification Based on Improved Inverse Discriminator Network. Information, 13(6), 293.

[105] Xing, Z.-J., Yin, F., Wu, Y.-C., and Liu, C.-L. (2018). Offline signature verification using convolution Siamese network. Proc.SPIE, 10615, 106151I. https://doi.org/10.1117/12.2303380

[106] Yilmaz, M. B., and Ozturk, K. (2018). Hybrid user-independent and user-dependent offline signature verification with a two-channel CNN. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2018-June, 639–647. https://doi.org/10.1109/CVPRW.2018.00094

[107] YIlmaz, M. B., and Yanikoğlu, B. (2016). Score level fusion of classifiers in off-line signature verification. Information Fusion, 32, 109–119. https://doi.org/10.1016/j.inffus.2016.02.003

[108] Yilmaz, M. B., Yanikoglu, B., Tirkaz, C., and Kholmatov, A. (2011). Offline signature verification using classifier combination of HOG and LBP features. 2011 International Joint Conference on Biometrics, IJCB 2011. https://doi.org/10.1109/IJCB.2011.6117473

[109] Yılmaz, M. B., and Yanıkoğlu, B. (2016). Score level fusion of classifiers in off-line signature verification. Information Fusion, 32, 109–119.

[110] Zhang, Z., Liu, X., and Cui, Y. (2016). Multi-phase offline signature verification system using deep convolutional generative adversarial networks. Proceedings - 2016 9th International Symposium on Computational Intelligence and Design, ISCID 2016, 2, 103–107.
https://doi.org/10.1109/ISCID.2016.2033

[111] Zois, E. N., Alewijnse, L., and Economou, G. (2016). Offline signature verification and quality characterization using poset-oriented grid features. Pattern Recognition, 54, 162–177. https://doi.org/10.1016/j.patcog.2016.01.009

[112] Zois, E. N., Alexandridis, A., and Economou, G. (2019). Writer independent offline signature verification based on asymmetric pixel relations and unrelated training-testing datasets. Expert Systems with Applications, 125, 14–32.

[113] Zois, E. N., Papagiannopoulou, M., Tsourounis, D., and Economou, G. (2018). Hierarchical dictionary learning and sparse coding for static signature verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 432–442.
[114] Zois, E. N., Zervas, E., Tsourounis, D., and Economou, G. (2020). Sequential motif profiles and topological plots for offline signature verification. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 13245–13255.
[115] Zouari, R., Mokni, R., and Kherallah, M. (2014). Identification and verification system of offline handwritten signature using fractal approach. 12–15.
[116] Upadhyay, A., Nadar, S., and Jadhav, R. (2020). Comparative study of SVM & KNN for signature verification. Journal of Statistics and Management Systems, 23(2), 191-198.
[117] Salama, M. A., and Hussein, W. B. (2017). An enhanced handwritten signature verification model applied on off-line benchmark data sets. International Journal of Image and Data Fusion, 8(4), 332-353.

