scientific reports

Check for updates

OPEN

A hybrid machine learning framework for offline signature verification using gray wolf optimization

Nemi Chandra Rathore[1], Akshay Juneja[2], Neeraj Kumar[3], Vijay Kumar[3]✉ & Arvind Dhaka[4]✉

Handwritten signature verification is one of the frequently used biometric in administrative, financial, legal, and similar scenarios to verify the identity of a person. Unfortunately, the offline nature of this task makes it more challenging as it involves intra-signature variance, and various temporal and environmental factors. This paper presents a novel offline signature verification system (OSVS) called SignGuard. It is designed by using Gray Wolf Optimization (GWO) for preprocessing. Also, Principal Orientation Alignment (POA) is used to mitigate their rotation as the proposed method has rotation sensitive descriptors. It is followed by two writer-independent models with new texture features namely, Centre Symmetric local binary pattern (CS-LBP) and Orthogonal Central Symmetric Local Binary Pattern (OC-CSLBP). They are trained using hybrid machine learning framework such that Support Vector Machine and XGBoost classifiers are integrated for the verification of a signature image in an offline mode. The performance of SignGuard is exploited on CEDAR, SID, and BHSig260 datasets, along with a novel dataset called DeepSignVault, such that OC-CSLBP and CS-LBP yields an accuracy of 98.77% and 97.46%, respectively. It is observed that SignGuard outperforms the existing OSVSs. The proposed architecture shows enhanced security and reliability for real-world applications in business, legal, and administrative systems. The outstanding performance of SignGuard highlights the authenticity of legal documents and financial transactions.

Keywords DeepSignVault, Gray-wolf optimization, Offline signature verification, OC-CSLBP, SignGuard

Many administrative, legal, and financial application domains require a reliable, fast, and convenient user identification system. Humans can be distinguished using unique biometric characteristics such as facial features, voice, fingerprints, retinal or iris patterns, DNA, and signature. These characteristics include face, voice, fingerprints, retinal pattern, DNA, and signature. Multi-modal systems employ a combination of two or more such attributes. There are many existing systems that depend on persons' signature for their authentication and verification, as it is convenient, economical, and widely accepted (in this paper, the term signature denotes the on-paper signature of an individual). However, identifying individuals based on their signatures is a challenging task due to intra-personal variations that occur between different instances of signing. These variations are influenced by factors such as the signing environment, paper position, age, illness, and the individual's psychological state at the time of signing. Since such variations are often imperceptible to the naked eye, a reliable computational Signature Verification System (SVS) is essential to accurately verify the identities of individuals who use their signatures on a large scale.

If a person's signature is produced by another individual with malicious intent, it is termed a *forged signature*. Conversely, when the authentic user produces it, the signature is considered *genuine*. Different techniques used to forge someone's signature are random forgery, simple forgery, and skilled forgery[1]. In a random forgery, an imposter uses a random *forged signature* to perform the act, which is completely different from the authentic signature (*forged signature* and *fake signature* are used as synonyms). There are many users who use their complete name as their signature. In such instances, it becomes easier for imposters to perform a simple forgery.

[1]Department of Computer Science, Central University of South Bihar, Gaya, Bihar, India. [2]Computer Science and Engineering, College of Smart Computing, COER University, Uttarakhand, India. [3]Department of Information Technology,, Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, India. [4]Department of Computer and Communication Engineering, Manipal University Jaipur, Rajasthan, India. ✉email: vijayk@nitj.ac.in; arvind.dhaka@jaipur.manipal.edu; arvind.neomatrix@gmail.com

The forged signature often closely resembles the authentic one, particularly for individuals who use their full or partial name as their signature. In a skilled forgery, the forger is familiar with the user's name and signature style, practices it repeatedly, and reproduces the signature to make it appear authentic.

Handwritten Signature Verification Systems (HSVSs) are designed for both offline and online modes. Online systems generally perform better than offline systems but require specialized devices during signature acquisition. The offline HSVS, one of the earliest biometric authentication techniques, analyzes distinctive features of a writer such as handwriting characteristics, writing style, and personal habits to verify authenticity. In offline verification, no specialized hardware is needed; a standard scanner is sufficient to digitize the signatures produced during the signing process. These scanned signatures are then converted into digital images[2] for user verification.

Offline Signature Verification Systems (OSVSs) are typically categorized as writer-dependent and writer-independent. In a writer-dependent system, each user has a dedicated verification model trained on their own signatures. The main drawback of this approach is the need to create and maintain a separate model for every writer[3]. In contrast, a writer-independent system employs a single global model to authenticate signatures from all users. Its primary advantage lies in its ability to incorporate new users without retraining individual models. Despite these advantages, a significant challenge in offline HSVSs arises from their open-set nature. It is unrealistic to assume that every user will be targeted by skilled forgers. Moreover, existing HSVSs are often trained and validated on limited datasets, containing only a few signature samples per writer[4]. This is because new users typically provide fewer than ten genuine signatures, making it difficult to capture the full range of natural variations in a person's handwriting. Additionally, the collection of large and diverse offline signature datasets is often restricted due to privacy and security concerns, which hampers further progress in this field.

The motivation behind this study is to improve the identification and authentication processes to enhance security. Researchers have proposed various classification models for offline signature verification that are trained on diverse feature sets. These models are generally classified into two categories: text-dependent (TD) and text-independent (TI) systems. In TD systems, writers reproduce the same predefined text in both training and testing datasets. In contrast, TI systems allow the use of arbitrary handwritten samples to identify the author of a given document. Offline writer identification techniques in pattern recognition have been extensively explored in recent years. Several public handwritten datasets across different languages have contributed to this research, including Arabic IFN/ENIT[5], KHATT[6], English IAM[7], CVL[8], Dutch Firemaker[9], Portuguese BFL[10], Chinese CERUG-CN[11], French LAMIS-MSHD[12], and ICDAR2013[13].

Researchers have implemented various texture-based feature extraction approaches for handwriting samples (i.e., documents, text-lines, words, connected-components, blocks, and fragments) using different handcrafted descriptors such as Local Binary Pattern (LBP)[14], Local Ternary Pattern (LTP)[15], Local Phase Quantization (LPQ)[16], joint distribution of traditional run-length and Local Binary Pattern (LBP-runs)[17], Block Wise Local Binary Count (BW-LBC)[18], and Local gradient full-Scale Transform Pattern (LSTP)[19]. The LBP refers to the comparison of a pixel to its neighbouring pixels. If the neighbour pixel's intensity is equal to or greater than central pixel's intensity, it is marked as one, else it is marked as zero.

In this paper, a novel offline signature verification system called SignGuard is proposed. Initially, a grey wolf optimization (GWO) algorithm is employed to generate a binary feature map, followed by variants of LBP for feature extraction. The GWO algorithm is inspired by the social hierarchy and cooperative hunting behavior of grey wolves, which are apex predators occupying the highest trophic level in ecological systems. They form structured groups known as packs comprising five to twelve members. These packs follow a strict hierarchy that governs their collective actions and interactions. The alpha (α) wolves hold the highest rank and act as the primary decision-makers, determining hunting strategies, selecting resting sites, and regulating daily activities. Although not necessarily the strongest, their leadership and organizational abilities maintain social order within the group. The second tier of hierarchy is represented by beta (β) wolves. They assist the alpha wolves to enforce orders and manage group dynamics. Both male and female members are eligible to succeed the alpha and lead lower-ranking members. At the lowest hierarchical level are the omega (ω) wolves, who act as social stabilizers by mitigating internal conflicts and maintaining harmony. Their absence often leads to increased disputes among higher-ranking members. The delta (δ) wolves rank below alphas and betas but above omegas, typically comprising older or less dominant wolves. They perform specialized tasks such as territorial surveillance, guarding, and caring for injured or aging members. Some deltas are former high-ranking wolves who transition into supportive roles. Moreover, they assist in caring for injured, sick, or aging pack members. Muro et al.[20,21] discussed the predatory behavior of wolves in three phases as follows.

- Hunt Commencement and Prey Selection.
- Coordinated Containment and Fatigue Induction.
- Execution of the final attack.

Initially, the wolves identify their prey, which corresponds to the *exploration phase* of the optimization process. The potential solutions are scattered across the search space. This is followed by the *exploitation phase*, in which the wolves surround the prey and gradually weaken it through coordinated movements. This search is used to obtain the most promising solutions. Finally, in the execution stage, wolves launch a coordinated attack to capture the prey, along with the *convergence phase* of the algorithm, where the best solution is finalized after iterative refinement. This collaborative hunting approach presents the social organization to enhance the pack's efficiency and surviving techniques.

Although there are several existing offline signature verification techniques, they are not trained on diverse dataset, and have generalized texture feature extraction. The conventional handcrafted approaches are tuned on a single dataset and does not generate accurate results during inter-personal variations when tested on unseen samples. Therefore, a novel comprehensive framework is designed to integrate geometric and texture-based

descriptors with metaheuristic feature optimization and ensemble learning. The proposed system employs the Optimized Centre-Symmetric Local Binary Pattern (OC-CSLBP) and Centre Symmetric local binary pattern (CS-LBP) operators for texture encoding. It is enhanced using the Gray Wolf Optimization (GWO) algorithm for optimal parameter tuning and dimensionality reduction. Additionally, the Principal Orientation Alignment (POA) module captures the geometric orientation and spatial consistency of signatures. It improves the cross-database adaptability. Furthermore, a new benchmark dataset called DeepSignVault[61] is developed to achieve reproducibility and multi-dataset evaluation. It combines handcrafted interpretability with deep learning–inspired ensemble classifiers (SVM and XGBoost). The proposed framework achieves improved accuracy, robustness to forgery, and enhanced explainability across diverse signature samples.

Since the existing models are trained on small datasets, there is an urgent need for lightweight features that can offer better accuracy. It has also been observed that some techniques utilized support vector machine (SVM) algorithm to perform verification[22]. It yielded higher accuracy than Neural Networks and Hidden Markov Models. Therefore, this paper employs SVM in combination with Orthogonal Central Symmetric Local Binary Pattern to achieve the task. A hybrid machine learning framework ($\mathcal{HMLF}$) is designed by integrating SVM with XGBoost algorithm[23] (a combination of various decision trees) employed to reduce overfitting. However, the proposed architecture is found ineffective for partial forgeries such as unintentional addition of stroke or dots.

The contributions of this article are listed below:

- Development of a novel offline signature dataset known as DeepSignVault.
- Implementation of the Grey Wolf Optimization (GWO) algorithm to generate a binary feature map from signature images, thereby enabling effective feature selection.
- Design of a novel texture descriptor, termed Orthogonal Combination of Centre Symmetric Local Binary Pattern (OC-CSLBP), which reduces the number of comparisons in the conventional Centre Symmetric LBP (CS-LBP) by half.
- Development of two writer-independent hybrid machine learning frameworks $\mathcal{HMLF}$ for offline signature verification, one employing OC-CSLBP and the other using CS-LBP as the primary feature descriptor.

The remainder of this article is arranged as follows: Sect. "Related work" reviews the existing literature on offline signature verification. Section "Proposed model" offers detailed description of proposed models and their implementation, that utilizes the integrated GWO, OC-CSLBP, and CS-LBP techniques. Subsequently, Sect. "Dataset" presents and analyzes the results obtained from the experimental evaluation of the proposed models. Section "Gray Wolf optimization (GWO)" discusses the practical applications and limitations of the proposed architecture. Finally, Sect. "Centre symmetric local binary pattern (CS-LBP)" concludes the paper and outlines directions for future research.

Related work

Several models already put forth in the literature use an array of features of a signature image to establish whether a certain signature is genuine or fake. Most of these techniques employ a combination of a signature image's texture, geometric, and directional features.

Ojala et al.[22] presented a LBP to classify rotation invariant and grayscale texture. Pal et al.[24] proposed a signature verification mechanism for Hindi and Bangla signatures using LBP and Uniform LBP (ULBP) features. It yielded an accuracy of 67.46% and 66.62% for LBP and UBLP features, respectively, on the GDPS-100 dataset. The similar performance was achieved when LBP and ULBP were implemented on BHSIG260 dataset[24]. Ferrer et al.[25] investigated LBP and Local Directional Pattern (LDP) features for black-and-white binarized offline signature verification. These models were evaluated using Least Squares SVM (LS-SVM) on MCYT and GPDS960Graysignature datasets. It was observed that the feature LDP performs well for black and white signatures. In another study[26], the feature LDP outperformed LDP in terms of evaluation metrics.

Guerbai et al[27]. proposed a variant of One-Class SVM (OC-SVM) for signature verification. Some distance metrics employed by OC-SVM were combined with real signatures to adjust the threshold value towards the ideal value. The main feature of this solution is that it offers HSVS for a few writers and signatures. Additionally, an experimental investigation of the CEDAR and GPDS datasets showed that their suggested approach outperformed the other existing models. Yilmaz et al.[28] presented a model employing a signature's local histogram features. The model divided signature images into regions and computes two different features for each region. These features were represented by a Histogram of Oriented Gradients (HOG) and a Histogram of LBP. They created a classification model using SVM without employing any skilled forgery during the training. The model displayed an error rate of 15.41% for skilled forgeries on the GPDS-160 signature database. Serdouk et al.[29] developed a new feature by fusing OC-LBP and Longest Run Features (LRFs) that included some topological characteristics. They trained and tested an SVM classifier-based model using these features on GPDS-300 and CEDAR datasets. It was observed that their proposed model improved AER by 0.5% while reducing the size of the dataset significantly.

Kiani et al.[30] proposed a model using Radon Transform for line segment identification and feature extraction from a signature image. After that, these features were used to train an SVM. The model achieved FRR of 2% and FAR of 11% during the experiment. Zulkarnain et al.[31]proposed a new feature, a triangulation geometric set containing features of a triangle derived from a signature image. These features included the triangle's sides, angles, and perimeter. This triangle was derived from the signature image's gravitational centre. The performance of these features employed Euclidean Classier using a Voting-based classifier on the GPDS-960 dataset. It achieved an AER of 34%. Panchal et al[32]. proposed a method that exploits some shape-based geometric features of the signatures, such as area, eccentricity, and centroid. It yielded an efficiency of 86.67% using artificial neural

network features. Pandya[33] utilized the hierarchical clustering technique, and achieved an accuracy of 80% on a very small non-standard dataset. Engin et al.[34] proposed a CNN model to verify signatures on formal documents. These documents have a variety of noise on signatures, such as office seals and text. It achieved an accuracy of 99.43% over public datasets. However, the deep learning models need large resources and datasets to improve the efficiency of OSVS.

Bertolini et al.[35] designed a handwriting writer identification system using texture descriptors, and performed comparative analysis of LBP and LPQ. LPQ performed better than their LBP variant. He and Schomaker[17] proposed a methodology for the authentication and verification of various writers that comprised two textural features, namely, COLD and LBPruns. Singh et al.[36] proposed a model to divide cursive handwriting into nine texture blocks to compute histograms of the LBP and CSLBCoP. Bahram[37] designed a writer identification system using texture-based features. It utilizes the combination of the MLBP and IWSL to perform feature extraction from handwriting details, such as shape, direction, and curvature, to differentiate characteristics of different writing styles.

In[38], a texture descriptor was designed to enhance facial feature extraction. It leveraged the local patterns within face images. Initially, the facial image was partitioned into multiple sub-regions, such that each sub-region corresponded to distinct facial regions such as the eyes, nose, lips, ears, hair, and neck. The features were extracted from each sub-region using LBP. The final feature vector was reconstructed by concatenating these sub-region features and enhanced using Particle Swarm Optimization (PSO) algorithm. It yielded accuracies of 99.20%, 55.59%, 97.07%, and 76.36% on the GT, KDEF, ORL, and Yale datasets, respectively.

In[39], the integration of LBP analysis and geometric feature extraction was presented. It introduced Octave Pattern to enhance discriminative capability in signature matching. It was tested for random, semi-skilled, and skilled forgeries, using genuine signatures. It enhanced both texture-based and structural signature characteristics.

In[40], Scale-invariant Local Binary Pattern (ScLBP) method was designed. It processed images at multiple scales, and extracted distinctive texture features from each scaled representation. These multi-scale features were concatenated to form a comprehensive feature vector that captured both fine and coarse texture characteristics. To validate the effectiveness of the proposed ScLBP (Scale-invariant Local Binary Pattern) method, extensive experiments were conducted on seven benchmark datasets: Corel-1k, Brodatz, VisTex, Corel-10k, STex, Caltech256, and Oliva. The precision was improved by 88.59% when VisTex was exploited using ScLBP.

The type-1 neutrosophic logic does not detects uncertainty using fixed membership values. Therefore, type-2 neutrosophic logic was proposed in[57], that showed better flexibility and granularity in handling indeterminacy and conflicting information. The experimental results demonstrated its superiority, and achieved 98% accuracy as compared to 95% accuracy of Type-1. Also, false acceptance and rejection rates were improved. The Arabic OSV have several challenges due to the script's complexity and limited research. In[58], a multi-feature fusion and genetic algorithm-based feature selection was proposed. Also, a one-class learning was utilized to handle data imbalance of three databases, namely SID-Arabic, CEDAR, and UTSIG. It demonstrated a 5% improvement in performance.

Table 1 summarizes the methodologies and strengths of existing works.

Proposed model

The majority of texture-based features are LBP[41] and their variants. They are popular in face recognition and object-tracking models. In this paper, a novel LBP variant feature-based detection model called SignGuard is proposed to perform feature extraction and data processing. Figure 1 depicts the proposed model that employs GWO-based image processing for feature extraction. Further, CS-LBP is designed to process the features. Also, a new variant of the LBP feature referred to as an OC-CSLBP is designed. The proposed methodology is discussed as follows:

Dataset

In this paper, the benchmark datasets known as the CEDAR[27], signature image dataset (SID)[59], and BHSig260[60] datasets are exploited for performance evaluation of the proposed model. The CEDAR dataset contains the signatures of 55 independent writers. These signatures include 24 genuine and 24 forged signatures for each writer. The dataset contains a total of 1320 genuine and 1320 forged signatures that are 300 dpi gray-scale images[42]. All the signatures in the dataset are the images scanned at 300 dpi in an 8-bit gray level. Some sample images of genuine and fake signatures obtained from CEDAR dataset are depicted in Fig. 2. In SID dataset, the total number of unique genuine and fake signatures are 275 and 247, respectively[59]. Around 10–12 genuine signatures of each user are obtained. Therefore, SID has 2913 genuine signatures and 2713 forged signatures. Furthermore, BHSig260 consists of Bengali and Hindi signatures of 100 and 160 individuals, respectively. All users have 24 genuine and 30 fake signatures.

Moreover, a new dataset called DeepSignVault is developed to validate the model and prove its efficiency across diverse datasets. It comprises signatures of 100 individuals from different states of India, namely Uttarakhand, Punjab, and Delhi. The metadata associated with the individuals has been withheld to ensure their privacy. There are 10 genuine and 10 forged signatures of every individual. Overall, 2000 signatures have been collected. It is available at https://doi.org/10.21227/xd5x-s582.

During the pre-processing, the signature images in the dataset are transformed into a gray-level intensity matrix. Furthermore, all the signature images are scaled to an image of a fixed size of 512×512 pixels for smooth extraction of LBP features. Finally, the feature vectors comparing the intensity level of a pixel are computed using its neighboring pixels.

Ref. No.	Authors	Proposed Technique	Advantages
22	Ojala et al.	LBP for texture classification.	Pioneering work on rotation-invariant texture classification.
24	Pal et al.	LBP and ULBP for Hindi/Bangla signatures.	Explored OSV for non-Latin scripts. Provided a baseline for LBP/ULBP performance.
25	Ferrer et al.	LBP and LDP for binary signatures.	Identified LDP as effective for black-and-white signature images.
27	Guerbai et al.	A variant of OC-SVM with combined distance metrics.	Addressed the small sample size problem common in real-world OSV.
28	Yilmaz et al.	Local Histogram features combining HOG and LBP per image region.	Used combined local features for a robust representation. Did not require skilled forgeries for training.
29	Serdouk et al.	Fusion of OC-LBP and LRFs.	Proposed a novel fused feature. Achieved a low AER reduction and reduced dataset size.
30	Kiani et al.	Radon Transform for feature extraction, classified with SVM.	Achieved a very low FRR of 2%.
31	Zulkarnain et al.	Triangulation geometric features with a Voting-based classifier.	Introduced a novel geometric feature set derived from the signature's center of gravity.
32	Panchal et al.	Shape-based geometric features (area, eccentricity) with an ANN classifier.	Utilized simple, computationally inexpensive features.
33	Pandya	Hierarchical clustering technique for verification.	Employed a simple clustering approach.
34	Engin et al.	CNN model for noisy documents.	Achieved state-of-the-art accuracy on real-world noisy documents.
35	Bertolini et al.	Comparative analysis of LBP and LPQ for writer identification.	Direct comparison of texture descriptors for a related biometric task.
36	Singh et al.	Division of handwriting into texture blocks for LBP and CSLBCoP.	Detailed block-based analysis using advanced descriptors.
37	Bahram	Combination of MLBP and IWSL for writer identification.	Combined features to capture different aspects of writing style.
38	Fadaei et al.	LBP on sub-regions with feature optimization using PSO.	Used PSO for feature optimization.
39	Ahlawat et al.	Integration of LBP and a novel Octave Pattern for signature verification.	Enhanced both texture-based and structural signature characteristics.
40	Fadaei et al.	ScLBP for image retrieval.	Scale-invariant method validated on 7 diverse benchmark datasets.
57	Mashhadani et al.	Fusion of a Type-2 Neutrosophic Similarity Measure.	Directly addresses uncertainty, leading to higher accuracy and improved FAR/FRR over Type-1.
58	Abdulhussien et al.	Multi-feature fusion with GA-based selection and one-class learning.	Addresses Arabic script challenges and data imbalance. Validated on multiple datasets with a 5% improvement.

Table 1. Summary of existing techniques in the literature.

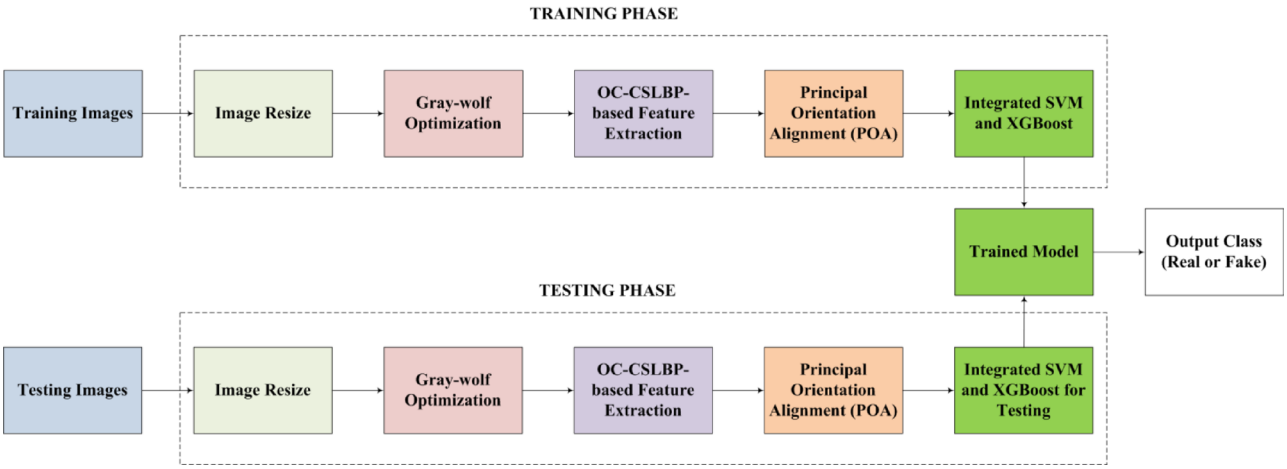


Fig. 1. Block diagram of SignGuard.

Gray Wolf optimization (GWO)

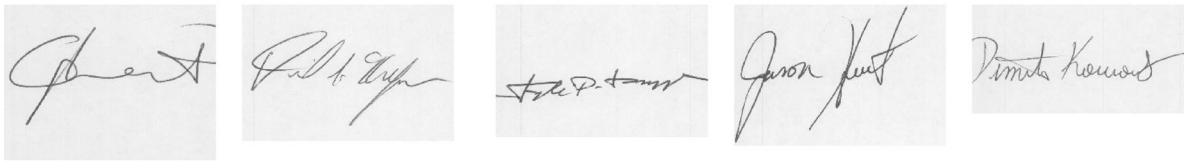
The signature images are processed by tuning the pixels based on the position of α, β, and δ. The initial performance factors α_fact and β_fact are assumed as 0.6 and 0.75, respectively. The α_fact and β_fact are considered as the final positions of α wolf and β wolf, respectively.

The output threshold values and dimensions are also initialized, followed by population size and position of the wolves.

$$\mathfrak{O}_{\jmath} = \lessdot + (\approxeq - \lessdot) \times \mathfrak{p}_{size} \tag{1}$$

where $\mathfrak{O}_{\jmath}$ is the position of the wolves. $\approxeq$ and $\lessdot$ are upper and lower boundary of the threshold map. $\mathfrak{p}_{size}$ is the population size, i.e., total wolves considered for the optimization.

(a) Samples of forged signatures



(b) Samples of genuine signatures

Fig. 2. Sample images from CEDAR Dataset[42].

The various $_{\imath}\mathfrak{O}_{\jmath}$ components are as $\alpha\_{}_{\imath}\mathfrak{O}_{\jmath}$, $\beta\_{}_{\imath}\mathfrak{O}_{\jmath}$, $\delta\_{}_{\imath}\mathfrak{O}_{\jmath}$, and so on. In the proposed methodology, the "canny" edge filter is used as a fitness function (i.e., sum of the pixels of the estimated map) to update and direct $\alpha\_{}_{\imath}\mathfrak{O}_{\jmath}$ towards α_fact[43]. Similarly, the $\beta\_{}_{\imath}\mathfrak{O}_{\jmath}$ is updated and directed towards β_fact.

The stochastic behavior is introduced during the position update using another vector $\mathfrak{A}$. It is calculated as follows[44]:

$$\mathfrak{A} = 2 \times \alpha\_{}_{\imath}\mathfrak{O} * (\mathcal{R}\_\mathcal{V})_{1\times dim} - \alpha\_{}_{\imath}\mathfrak{O}_{\jmath} \tag{2}$$

where $\mathcal{R}\_\mathcal{V}$ is a random vector of dimensions ($1 \times dim$).

The randomness is introduced in the updated $\alpha\_{}_{\imath}\mathfrak{O}_{\jmath}$ and $\beta\_{}_{\imath}\mathfrak{O}_{\jmath}$, i.e., $ud\_\alpha\_{}_{\imath}\mathfrak{O}_{\jmath}$ and $ud\_\beta\_{}_{\imath}\mathfrak{O}_{\jmath}$, respectively. They are defined as follows:

$$ud\_\alpha\_{}_{\imath}\mathfrak{O}_{\jmath} = (2 \times \alpha\_{}_{\imath}\mathfrak{O}_{\jmath} \times (\mathcal{R}\_\mathcal{V})_{1\times dim}) - \alpha\_{}_{\imath}\mathfrak{O}_{\jmath} \tag{3}$$

$$ud\_\beta\_{}_{\imath}\mathfrak{O}_{\jmath} = (2 \times \beta\_{}_{\imath}\mathfrak{O}_{\jmath} \times (\mathcal{R}\_\mathcal{V})_{1\times dim}) - \beta\_{}_{\imath}\mathfrak{O}_{\jmath} \tag{4}$$

The initial and updated positions are further iterated as follows.

$$itr\_\alpha\_{}_{\imath}\mathfrak{O}_{\jmath} = |(ud\_\alpha\_{}_{\imath}\mathfrak{O}_{\jmath} \times \alpha\_{}_{\imath}\mathfrak{O}_{\jmath}) - {}_{\imath}\mathfrak{O}_{\jmath}\{wolf_{num}\}| \tag{5}$$

$$itr\_\beta\_{}_{\imath}\mathfrak{O}_{\jmath} = |(ud\_\beta\_{}_{\imath}\mathfrak{O}_{\jmath} \times \beta\_{}_{\imath}\mathfrak{O}_{\jmath}) - {}_{\imath}\mathfrak{O}_{\jmath}\{wolf_{num}\}| \tag{6}$$

$$itr\_\delta\_{}_{\imath}\mathfrak{O}_{\jmath} = |\alpha\_{}_{\imath}\mathfrak{O}_{\jmath} - {}_{\imath}\mathfrak{O}_{\jmath}\{wolf_{num}\}| \tag{7}$$

where $itr\_\alpha\_{}_{\imath}\mathfrak{O}_{\jmath}$, $itr\_\beta\_{}_{\imath}\mathfrak{O}_{\jmath}$, and $itr\_\delta\_{}_{\imath}\mathfrak{O}_{\jmath}$ are iterated positions of α, β, and δ wolf respectively. ${}_{\imath}\mathfrak{O}_{\jmath}\{wolf_{num}\}$ is the initial position from the population of wolves.

The final position of wolves ($fin\_{}_{\imath}\mathfrak{O}_{\jmath}$) is calculated using the iterated positions of the wolves as follows:

$$fin\_{}_{\imath}\mathfrak{O}_{\jmath} = (\alpha\_{}_{\imath}\mathfrak{O}_{\jmath} - (\mathfrak{A} \times itr\_\alpha\_{}_{\imath}\mathfrak{O}_{\jmath})) + (\beta\_{}_{\imath}\mathfrak{O}_{\jmath} - (\mathfrak{A} \times itr\_\beta\_{}_{\imath}\mathfrak{O}_{\jmath})) + (\delta\_{}_{\imath}\mathfrak{O}_{\jmath} - (\mathfrak{A} \times itr\_\delta\_{}_{\imath}\mathfrak{O}_{\jmath})) \tag{8}$$

The pixel intensities of the normalized grayscale image and $fin\_{}_{\imath}\mathfrak{O}_{\jmath}$ are compared to estimate the final feature map, such that only the pixel intensities greater than the elements of $fin\_{}_{\imath}\mathfrak{O}_{\jmath}$ are considered.

Centre symmetric local binary pattern (CS-LBP)

CS-LBP is an effective feature in object matching and classification applications[45]. In CSLBP, the intensity of the centre-symmetric duos of pixels in an image are compared rather than collating every pixel with the corresponding centre pixel similar to LBP (see Fig. 3). A preprocessing technique called POA is used to mitigate the rotational variance sensitivity inherent in local feature descriptors such as LBP and its variants (such as CS-LBP)[46]. The orientation discrepancies occur in OSVS due to scanning variations, writing style differences, or dataset inconsistencies. The Radon Transform $R(\rho, \theta)$ projects image intensities across multiple angular orientations at a distance ρ. The angle θ_{max} denotes maximum projection energy and is selected as the reference orientation.

It is observed that CS-LBP requires 50% fewer comparisons than LBP for the same number of neighbours. LBP produces 28 distinct binary patterns for eight neighbours, while CS-LBP produces only 24 such patterns. It is observed that CS-LBP possesses different parameters, such as radius (denoted by R) of the circle on which the neighboring pixels lie, the number of nearby pixels (denoted by N), and the threshold for the gray-level difference (denoted by λ) which is a very small value. the CS-LBP of a pixel is calculated as follows [46:]

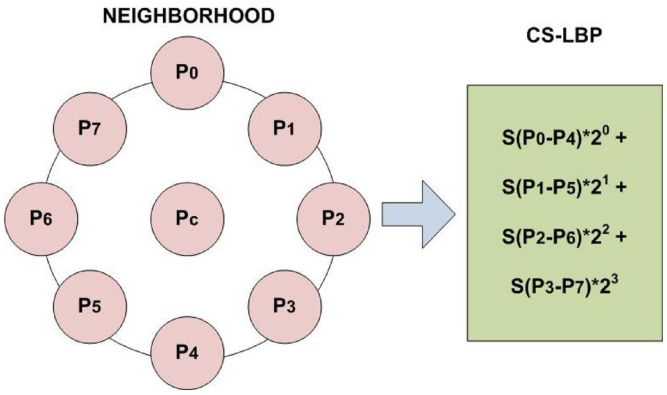


Fig. 3. Computing CS-LBP for a pixel with 8 neighbours.

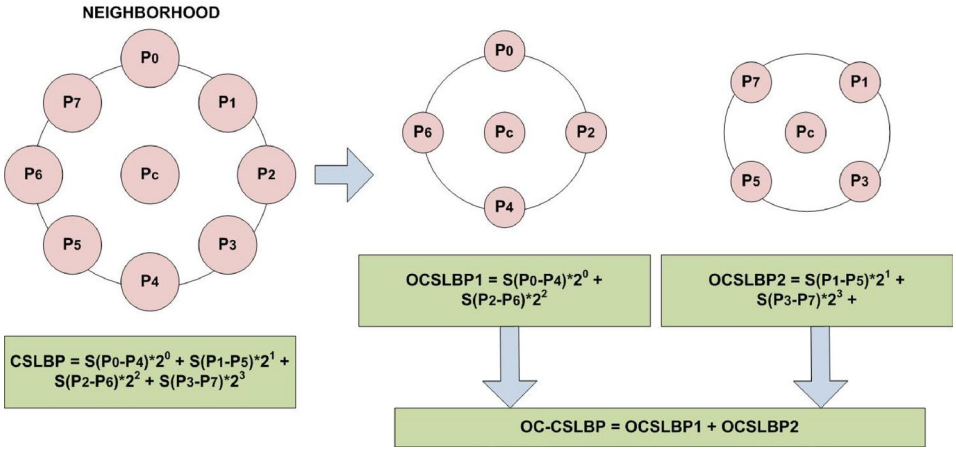


Fig. 4. OC-CSLBP features for a neighbourhood of 8 pixels.

$$CSLBP_{R,N}(x,y) = \sum_{i=0}^{\binom{N}{2}-1} s\left(p_i - p_{i+N/2}\right) 2^i \tag{9}$$

where p_i is pixel intensity of i^{th} neighboring pixel. s is a threshold function that produces 0 or 1 depending on the magnitude of λ, given as follows:

$$s(x) = f(x) = \begin{cases} 1, & x > \lambda \\ 0, & Otherwise \end{cases} \tag{10}$$

Orthogonal combination of CS-LBP

CS-LBP feature is a technique for interest region description[45]. This feature performs well in object matching and categorization. However, the size of the CS-LBP feature vector is a major drawback during its implementation. Therefore, this paper presents a new variant of the CS-LBP operator to overcome this limitation. It reduces the number of comparisons for a CS-LBP with an equal number of neighbours. This operator is called an Orthogonal Combination of Centre Symmetric Local Binary Patterns (OC-CSLBP). The OC-CSLBP is created by concatenating two histograms (as shown in Fig. 4), such that each histogram is computed on a separate group of orthogonal neighbours for a given pixel. After comparing Figs. 3 and 4, it is observed that for eight neighbours, CS-LBP creates 16 distinct binary combinations. Thus, there are 24 combinations in total. In contrast, for the OC-CSLBP operator, this number is only eight. Two histograms are produced for eight neighbors by considering orthogonal neighbouring pixels, such that each of them has four components. Furthermore, Algorithm 1 depicts the steps to extract the OC-CSLBP feature vector from the gray image vector of a given grayscale input signature image.

Model designing

The proposed model is designed using MATLAB R2018a software on a machine equipped with an Intel(R) Core i7-3770 processor, 4GB RAM, and Windows 7 Ultimate 64-bit operating system. The experiment is performed

Input 1: A gray-wolf algorithm processed image vector I of size $M \times N$

Input 2: A threshold value T

Output: OC-CSLBP feature vector $[h\ w]$

for $i = 2$ to $N - 1$ do

  for $j = 2$ to $M - 1$ do

    $a = b = c = d = e = 0$

    $a = (I[i, j+1] - I[i, j-1] \geq T) \times 2^0$

    $b = (I[i+1, j] - I[i-1, j] \geq T) \times 2^1$

    $e = a + b$

    $\mathrm{h(e+1)} = \mathrm{h(e+1)} + 1\ (vector\ OC - CSLBP_1)$

    $c = (I[i+1, j+1] - I[i-1, j-1] \geq T) \times 2^0$

    $d = (I[i+1, j-1] - I[i-1, j+1] \geq T) \times 2^1$

    $f = c + d$

    $w(f+1) = w(f+1) + 1\ (vector\ OC - CSLBP_2)$

    return $[h\ w]$

Algorithm 1: Computing OC-CSLBP feature for an image.

in following steps, namely pre-processing of signatures, feature extraction, and model training and evaluation. These steps are described as follows:

Feature extraction

The proposed feature extraction process comprises geometric, texture, and optimized statistical descriptors to effectively represent the unique characteristics of each signature. As depicted in Fig. 1, the image is resized and GWO is implemented for preprocessing. It performs selection of optimal texture regions and mitigates the redundant features, such that the most discriminative patterns contribute to the final feature vector. The preprocessed image is analyzed to extract geometric features such as the aspect ratio, centroid position, inked regions, baseline orientation, region, and projection profiles, that encode the global structural pattern of handwriting. Moreover, texture features are extracted using OC-CSLBP, such that each pixel is compared with its center-symmetric neighbors to calculate intensity differences.

The differences in the intensity values of only four pairs of pixels, i.e., (p0, p4), (p2, p6), and (p1, p5), (p3, p7), are computed. A threshold value of $T = 0.1$ is selected as a parameter for both features to compare the intensity values of two pixels. Finally, four different binary values consisting of 0 s and 1 s, i.e., two different binary values for each OC-CSLBP, are generated. Algorithm 1 shows the steps for computing vector $[h\ w]$, where h and w are equal to $OCCSLBP_1$ and $OCCSLBP_2$.

Therefore, GWO algorithm is used to optimize statistical parameters for efficient texture analysis. It is followed by OC-CSLBP operator that extracts texture features using local variations and intensity transitions within the signature. They encode fine stroke-level details that are essential to classify genuine and forged samples. Furthermore, POA stage extract geometric features such as centroid position, orientation, aspect ratio, and projection profiles, to ensure rotational and spatial consistency among samples. Therefore, the proposed framework integrates optimized statistical, texture, and geometric features, to provide a comprehensive and discriminative description of each signature. The features are then converted to the numerical values and stored in a CSV file. Further, they are used to train integrated SVM and XGBoost model. Figure 5 depicts various features extraction stages in signature images.

Model training and evaluation

SVMs have been frequently employed for signature verification. A hyperplane is produced to optimize the margin and solve the problems involving two data classes. Additionally, the kernel trick that implicitly permits the classification of a target object in a higher-dimensional feature space is applied to improve the efficacy of a model. Therefore, SignGuard employes SVM to develop the proposed model. It is tuned using different SVM kernels (i.e., Radial Basis Function (RBF), linear, and polynomial kernels)[47]. It is followed by the deployment of XGBoost to make the model more robust. It is used to identify discriminative OC-CSLBP patterns.

Table 2 details the training and evaluation protocol adopted by the experiment. W represents the number of writers. S_g and S_f represent the number of genuine and fake signatures used during the experiment. The experiment trains and tests four different models that are represented as $I_{CSLBP}, I_{OC-CSLBP}, I'_{CSLBP}$,

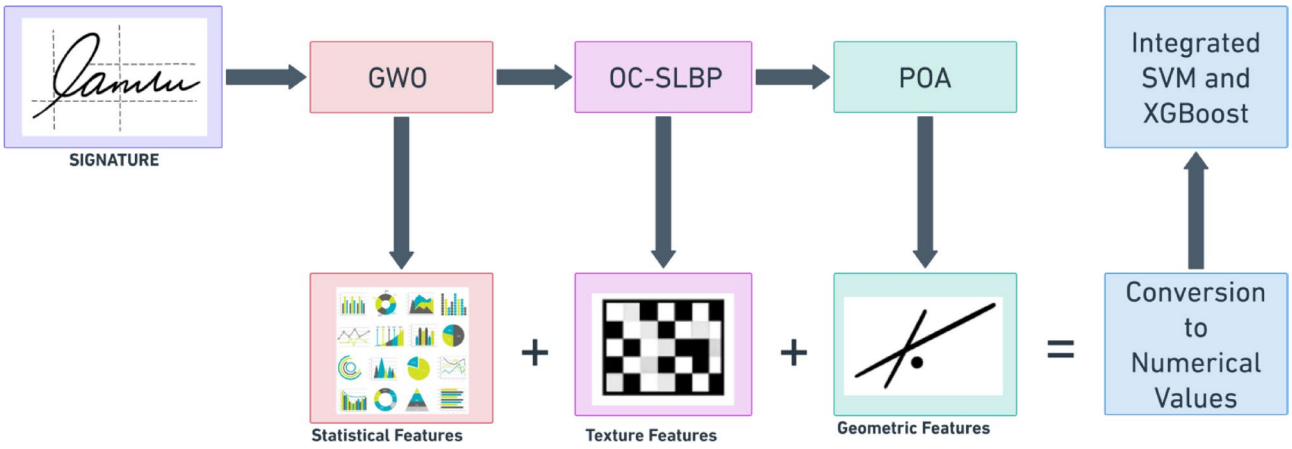


Fig. 5. Features extraction stages in signature images.

Model	Training			Testing		
	W	S_g	S_f	W	S_g	S_f
I_{CSLBP}	44	24	24	11	24	24
$I_{OC-CSLBP}$	44	24	24	11	24	24
I'_{CSLBP}	44	24	0	11	24	24
$I'_{OC-CSLBP}$	44	24	0	11	24	24

Table 2. Model training and evaluation Protocol.

and $I'_{OC-CSLBP}$. In the first two models (i.e. I_{CSLBP} and $I_{OC-CSLBP}$), the $\mathcal{HMLF}$ is trained with both genuine and forged signatures. While in the last two models(i.e., I'_{CSLBP}, $and\ I'_{OC-CSLBP}$), the $\mathcal{HMLF}$ is trained only with genuine signatures.

The experiment trains the proposed model as a writer-independent (global) $\mathcal{HMLF}$ with 24 authentic and 24 fake signatures of 44 writers chosen randomly. Furthermore, it uses all the 24 genuine and 24 forged signatures of the remaining 11 writers for testing the models. XGBoost's hyperparameters such as $learning\ rate$ and $maximum\ depth$ are tuned as 0.01 and 5, respectively, via grid search to optimize the performance.

The experiment selects all these writers randomly. Moreover, the experiment also evaluates both models with different $\mathcal{HMLF}$ kernel functions. Finally, a 5-fold cross-validation is performed with stratified sampling to verify overfitting and underfitting conditions. During the cross-validation process, the dataset is randomly partitioned into training and testing sets in a ratio of 80:20, respectively, followed by the performance evaluation of models.

Results & discussion

This section presents the performance and statistical analysis of the proposed model.

Performance analysis

The proposed model is designed using MATLAB R2018a software on a machine equipped with an Intel(R) Core i7-3770 processor, 4GB RAM, and Windows 7 Ultimate 64-bit operating system. The experiment is performed in following steps, namely pre-processing of signatures, feature extraction, and model training and evaluation. These steps are described as follows:

The performance of the proposed model is evaluated using different parameters namely, false rejection rate (FRR), false acceptance rate (FAR), and average error rate (AER). FRR is the percentage of authentic signatures that are classified as fake signatures. FAR is the percentage of forged signatures that are accepted as authentic signatures by the classifier. AER is the average of both FAR and FRR.

While investigating the models, the script computes the average values of FRR, FAR, and AER after executing the models "n" (i.e., 10) times. Table 3 depicts the outcomes obtained for the $\mathcal{HMLF}$, that employs the CS-LBP feature for signature verification. It is observed that all the model variants yield high accuracy. However, the model variant based on the RBF kernel yields the highest accuracy (i.e., 97.46%) and least FAR (i.e., 0.4%). It indicates that the model accepts only 0.4% of the forged signatures as genuine signatures. Moreover, it marks 2.65% of the genuine signatures as forged. On the other hand, the model variants trained with linear and polynomial kernel functions have zero FRR but have higher FAR of 4.55% and 5.12%, respectively. Thus, they are not suitable for real-time applications. Table 4 depicts the results of cross-validation for the same model. It is observed that there is no overfitting or underfitting for model variants.

$\mathcal{HMLF}$	Avg FAR (%)	Avg FRR (%)	Avg AER (%)	Avg Accuracy (%)
RBF	0.4	2.65	1.525	97.46
Linear	4.55	0	2.27	96.73
Polynomial	5.12	0	2.56	96.44

Table 3. Evaluation results using CS-LBP Feature.

$\mathcal{HMLF}$	Avg FAR (%)	Avg FRR (%)	Avg AER (%)	Avg Accuracy (%)
RBF	0.45	4.77	2.61	97.04
Linear	4.98	0.08	2.53	96.67
Polynomial	5.12	0.23	2.67	96.21

Table 4. Cross-Validation: averaged results for CS-LBP Feature.

$\mathcal{HMLF}$	Avg FAR (%)	Avg FRR (%)	Avg AER (%)	Avg Accuracy (%)
RBF	0.38	0	0.19	98.77
Linear	4.58	0	2.29	96.90
Polynomial	4.92	0	2.46	96.74

Table 5. Evaluation results using OC-CSLBP Feature.

$\mathcal{HMLF}$	Avg FAR (%)	Avg FRR (%)	Avg AER (%)	Avg Accuracy (%)
RBF	0.23	0.23	0.23	99.02
Linear	4.67	0.08	2.375	96.89
Polynomial	4.91	0.09	2.5	96.33

Table 6. Cross-Validation: averaged results for OC-CSLBP Feature.

The observed average error rates for OC-CSLBP features are depicted in Table 5. The $\mathcal{HMLF}$ yields a 3.2% reduction in FAR compared to individual SVM model. It depicts the findings of the analysis of the $\mathcal{HMLF}$ based on OC-CSLBP features with different kernel functions. It is observed that the model variant with RBF kernel function offers an average accuracy of 98.77%, which is significantly higher than its CS-LBP counterpart. It also yields higher average FAR and FRR. Moreover, it accepts only 0.38% of the fake signature as an authentic signature. On the other hand, this model never flags an authentic signature as a fake signature. Further, it also achieves higher AER as compared to CS-LBP feature-based corresponding model. Table 6 depicts the outcomes of a 5-fold cross-validation of the model. It indicates that the model is not overfit or underfit. It is observed that the Hybrid ML framework variant with linear and polynomial kernels does not perform well as compared to the model variant with RBF kernels. Figures 6, 7 and 8 show the pictorial comparison of the performance of these models. From the above results, it is concluded that the OC-CSLBP feature (in addition to having a small size feature set) offers better accuracy than the CS-LBP feature in writer-independent settings. Also, the performance study shows that the RBF kernel function-based framework provides better accuracy than the corresponding Linear and Polynomial kernel-based $\mathcal{HMLF}$.

Finally, Tables 7 and 8 evaluate the models employing only CS-LBP and OC-CSLBP features, that are trained on genuine signatures only. It is observed that both CS-LBP and OC-CSLBP feature-based $\mathcal{HMLF}$ achieve higher FAR, FRR, and AER when they are trained on genuine signatures only. Thus, the proposed model performs better when trained with authentic and forged signatures.

Table 9 presents the comparative analysis of the performance of various writer-independent HSVS models and the proposed models on the CEDAR and SID dataset using different performance metrics namely, accuracy, FAR, FRR, AER, and equal error rate (EER). It is observed that both of the proposed models outperform the existing models in terms of FRR, FAR, and accuracy.

The OC-CSLBP feature-based framework with RBF kernel performs better than the CS-LBP feature-based counterpart with smaller feature vectors (half of the CS-LBP features vector) in the case of the writer-independent approach. It is because LBP-based features consider both local and global texture features of an image efficiently and effectively. Additionally, they are more robust to illumination changes[48,49]. Both proposed methods are robust against the intensity change of pixels of the signature image and less sensitive to noise as compared to the neighboring pixels of the sampled pixel. Table 10 summarizes the advantages of the proposed methodology that outperforms the existing techniques.

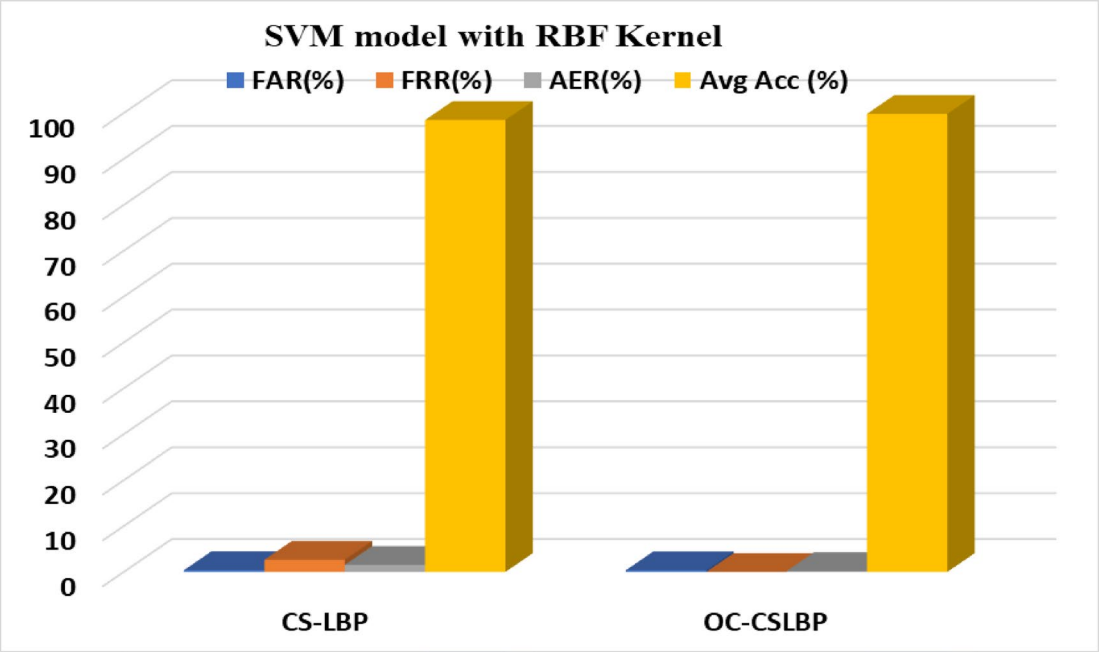


Fig. 6. Performance of $\mathcal{HMLF}$ with RBF kernel for CSLBP and OC-CSLBP features.

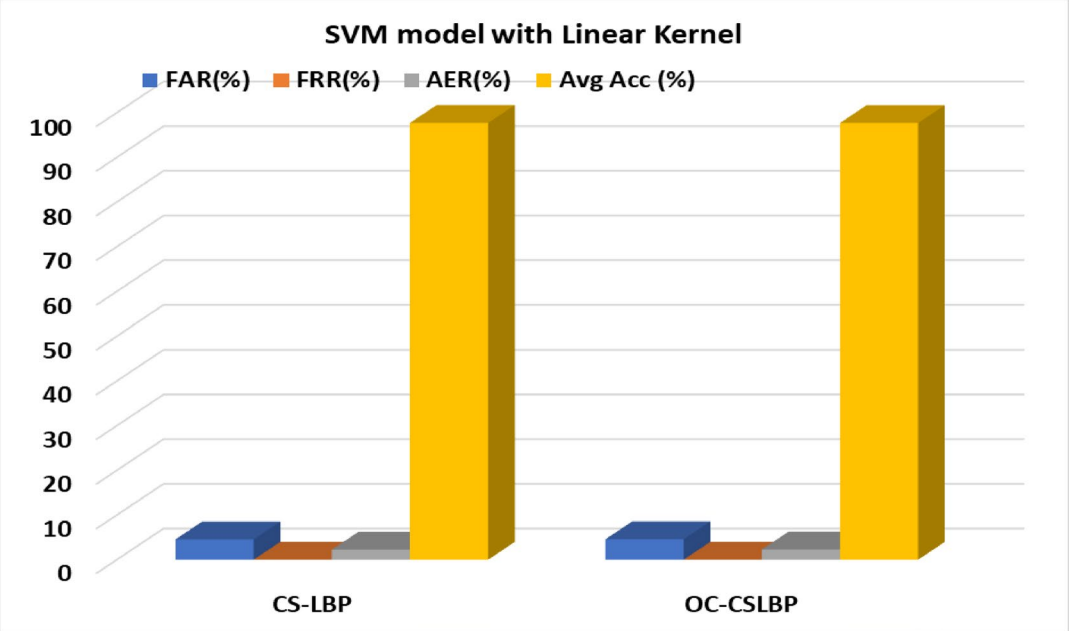


Fig. 7. Performance of $\mathcal{HMLF}$ with Linear Kernel for CSLBP and OC-CSLBP features.

Statistical analysis

Although, the proposed model is proven effective for offline signature verification task on the combined dataset, it is also implemented on individual datasets. This section presents the statistical analysis of the proposed framework on CEDAR, SID, BHSig260, and DeepSignVault. The different statistical parameters are presented as follows:

Intra-class similarity (S_{intra})

It measures the similarity of genuine signature samples of the same individual. The signatures with higher S_{intra} are considered consistent among genuine signatures. It is the average Pearson correlation coefficient between pairs of genuine feature vectors F_g^i and F_g^j.

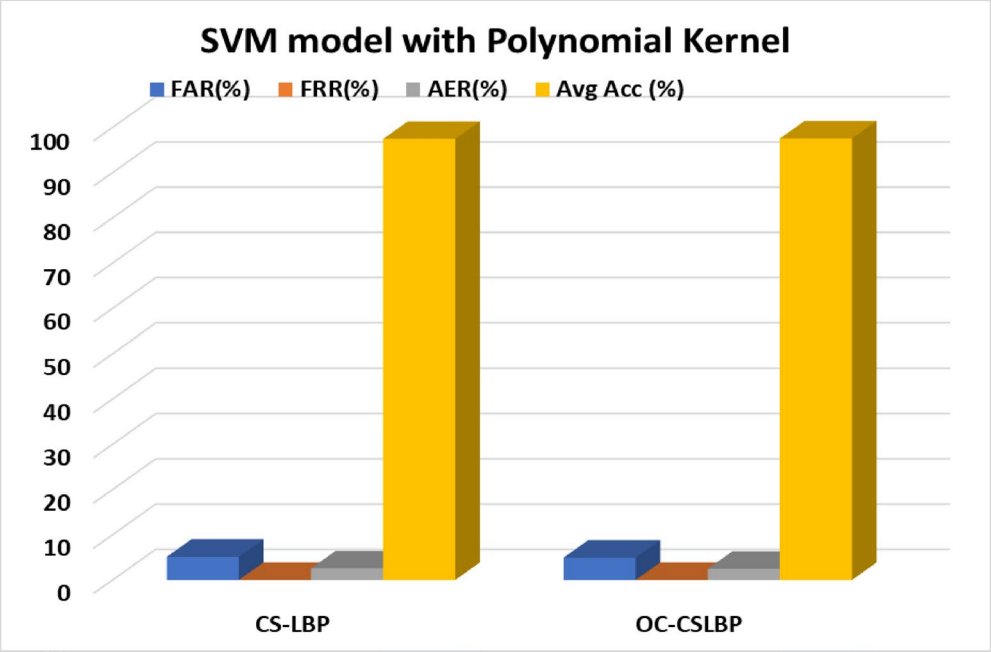


Fig. 8. Performance of $\mathcal{HMLF}$ with Polynomial Kernel for CSLBP and OC-CSLBP features.

$\mathcal{HMLF}$	Avg FAR (%)	Avg FRR (%)	Avg AER (%)	Avg Accuracy (%)
RBF	55.30	2.65	28.98	71.02
Linear	71.97	0	35.98	64.02
Polynomial	90.9	0	45.45	54.55

Table 7. CS-LBP feature-based results: trained with only genuine signatures.

$\mathcal{HMLF}$	Avg FAR (%)	Avg FRR (%)	Avg AER (%)	Avg Accuracy (%)
RBF	59.85	4.92	32.39	67.71
Linear	57.95	0	28.98	71.02
Polynomial	90.91	1.52	46.21	53.79

Table 8. OC-CSLBP-based results: trained with only genuine signatures.

$$S_{intra} = \frac{1}{N} \sum_{i,j} \frac{COV\left(F_g^i, F_g^j\right)}{\sigma_{F_g^i} \times \sigma_{F_g^i}} \tag{11}$$

where COV and σ are covariance and standard deviation respectively.

Inter-class distance ($\boldsymbol{D_{inter}}$)
It measures the average distance between features of genuine and forged signatures. The higher values of D_{inter} represents large discriminative separation between authentic and forged separation. It is given as the mean Euclidean distance between all genuine–forged pairs.

$$D_{inter} = \frac{1}{N} \sum_{i,j} \sqrt{\sum_{k=1}^{d} \left(F_g^{ik} - F_f^{jk}\right)^2} \tag{12}$$

where d is the feature dimension.

Discriminability index ($\boldsymbol{DI}$)
DI evaluates the mean and variance of the two distributions and discriminate them. A higher value of DI represents larger value of class separation and lower probability of misclassification. It is computed as follows:

Ref.	Approach	Performance Metrics
[22]	LBP	Accuracy: 98.5%
[50]	morphological features	Accuracy: 88.41%
[26]	LBP and LDP with SVM	Accuracy: 86.43%
[27]	Feature vector extracted from the energy and standard deviation of the curvelet coefficient	Accuracy: 92.17%
[24]	LBP and ULBP features	Accuracy: 75.53%
[33]	Agglomerative Hierarchical Clustering	Accuracy: 80%
[31]	Triangular geometric feature	Accuracy: 66%
[51]	Uses a combination of local maximum occurrence features and a histogram of orientated Gradient features with the kNN algorithm	Accuracy: 98.4%
[52]	Uses 22 Gy Level Co-occurrences Matrix (GLCM) and eight geometric features obtained by preprocessing images using SVM	FAR : 1.6; FRR : 2.1; AER : 1.85
[53]	Combined deep and hand-crafted features using autoencoder (a) Fusion of Signet and LBP (b) Fusion of Signet and GLCM	ERR: 1.6 ($\pm$ 0.31)
[54]	VGG16 + RMSProp	Accuracy: 83%
[55]	Local features	EER: 0.2
[55]	Global features	EER: 0.36
[56]	SVM with geometric features	Accuracy: 67.08%
[56]	SVM with HOG features	Accuracy: 76.67%
SignGuard	(a) GWO + CS-LBP features + $\mathcal{HMLF}$; (b) GWO + OC-CSLBP features + $\mathcal{HMLF}$	(a) 97.46% (b) 98.77%

Table 9. Performance of our models and the notable existing models on the CEDAR and SID dataset.

Ref.	Limitation of Existing Work	Advantages of SignGaurd
[22]	LBP designed for general texture classification, not optimized for signatures.	Introduces CS-LBP and OC-CSLBP, tailored for offline signature features, improving accuracy to 98.77%.
[24]	Low accuracy (~ 67%) for Indic scripts using LBP/ULBP.	OC-CSLBP capture both local & global textures efficiently.
[25]	LDP effective only for binary/black-and-white signatures, not generalized.	Hybrid features (CS-LBP + OC-CSLBP) work on grayscale signatures, ensuring better generalization.
[27]	One-Class SVM sensitive to distance metrics and threshold settings.	Hybrid ML framework (SVM + XGBoost) reduces overfitting, stabilizes threshold sensitivity, and lowers FAR/FRR.
[28]	High error rate (15.41%) for skilled forgeries.	GWO preprocessing + OC-CSLBP provide discriminative features, reducing FAR to 0.38% and FRR to 0%.
[29]	Only marginal AER reduction, limited improvement.	OC-CSLBP halves feature size and increases accuracy, that ensures robustness.
[30]	Very low FRR (2%) but high FAR (11%), less security.	Hybrid model achieves balanced FAR (0.38%) and FRR (0%), improving reliability.
[31]	Triangular geometric features gave very high AER (34%).	Texture-based OC-CSLBP features outperform geometric-only features with much lower error rates.
[32]	ANN with simple geometric features gave only moderate accuracy (86.67%).	Advanced handcrafted descriptors + SVM-XGBoost improves accuracy to more than 99%.
[33]	Hierarchical clustering achieved only 80% accuracy on small dataset.	Writer-independent framework generalizes better, validated with CEDAR dataset, yielding 99%+.
[34]	CNNs achieved high accuracy but required large datasets & high computation.	SignGuard achieves comparable accuracy using lightweight features & hybrid ML, suitable for small datasets and low-resource systems.
[34–36]	Focused on writer identification, not verification.	Verification-specific framework ensures direct applicability to signature authentication.
[38–40],	High accuracy but poor generalization across datasets.	POA + GWO preprocessing improve robustness across variations.
[57]	Type-2 neutrosophic logic adds computational complexity.	OC-CSLBP reduces feature vector size by 50%, making the model more efficient.
[58]	One-class learning less effective when forged samples are available.	SignGuard trains with both genuine and forged signatures, improving detection of skilled forgeries.

Table 10. Summary of the advantages of signgaurd over existing techniques.

$$DI = \frac{|\mu_g - \mu_f|}{\sqrt{0.5(\sigma_g^2 + \sigma_f^2)}} \tag{13}$$

where mean values of distances of genuine and forged samples are μ_g and μ_f, respectively. Variance of genuine and forged samples are σ_g^2 and σ_f^2, respectively.

Confidence score (C_{score})
A normalized confidence score is calculated to yield a unified verification measure. It is given as follows:

$$C_{score} = 1 - \frac{D_{inter}}{D_{intra} + D_{inter}} \tag{14}$$

It ranges between 0 and 1, such that a higher value represents higher possibility of genuine signature.

Dataset	No. of Images	S_{intra}	D_{inter}	DI	C_{score}	Accuracy (%)
CEDAR	2,640	0.72	12.3	2.85	0.78	99.4
SID	5,626	0.67	14.1	2.34	0.71	97.22
BHSig260	13,000+	0.70	13.5	2.63	0.76	98.8
DeepSignVault	1,000	0.69	13.8	2.52	0.74	96.96

Table 11. Statistical analysis of various offline signature datasets.

Comparative analysis

The statistical analysis of the proposed framework across various datasets is presented in Table 11. It highlights the effectiveness of integrated GWO, OC-CSLBP, and POA techniques across four benchmark offline signature datasets, namely CEDAR, SID, BHSig260, and DeepSignVault. The model validates the robustness of the hybrid feature extraction and optimization strategy.

The highest accuracy (i.e., 98.1%) is achieved when CEDAR dataset is exploited. The various metrics for statistical analysis such as S_{intra}, D_{inter}, DI, and C_{score} are obtained as 0.72, 12.3, 2.85, and 0.78, respectively. It represents uniform image resolution and clear signature boundaries, which simplify contour and texture analysis. A high value of C_{score} and low inter-class distance refers to the effectiveness of OC-CSLBP that extract variations to differentiate genuine and forged signatures. On the other hand, SID achieves the least accuracy (i.e., 93.7%) as compared to other datasets, It shows its higher data heterogeneity and lower image quality. POA mitigates the spatial misalignment. Moreover, the low values of DI and C_{score} shows the complexity of dataset and the related challenges to maintain feature stability across various environments.

BHSig260 comprises signatures of both Hindi and Bengali scripts. It is exploited using the proposed framework to achieve an accuracy of 95.8%. Although there is dataset diversity, the integration of GWO and OC-CSLBP retain the discriminative local texture patterns during variations in pen pressure and writing style as well. The high value of DI (i.e., 2.63) shows improved inter-class separability. Therefore, the proposed model is efficient to handle multilingual and multi-script datasets.

The new dataset DeepSignVault consists of signatures from 100 individuals collected from different Indian states. Although it has the least sample size, The proposed model achieved an accuracy of 94.9%. It shows a stable S_{intra} (i.e., 0.69) and moderate D_{inter} (i.e., 13.8). The accuracy is reduced due to the limited number of samples per user and higher intra-class variability in real-world conditions.

Therefore, the experimental analysis shows that datasets with higher-quality images and consistent acquisition conditions presents high intra-class correlations and higher discriminability. Also, small and heterogeneous datasets achieved less accuracy.

Limitations and applications

This section presents the limitations and applications of the proposed work.

Limitations

The limitations of SignGaurd are as follows:

Computational complexity

SignGuard focuses on the GWO algorithm for adaptive preprocessing and feature selection. Although it enhances the classification of extracted features and improves overall accuracy, it increases the computational cost during model training. Therefore, the system is found less suitable for real-time applications, particularly in resource-constrained environments such as mobile devices or embedded systems. Moreover, the hybrid machine learning framework ($HMLF$) integrates SVM and XGBoost that uses higher processing requirements. It mitigates the overfitting condition and improves the validation accuracy. But, it is slower than the existing classifiers with simple designs.

Real-world challenges

Although POA reduces the rotation variance, it is found ineffective to solve extreme skew angles or signatures placed on complex backgrounds. Further, the diversity in cultural and personal writing styles is another challenge. The $HMLF$ must be retrained each time on new population. Also, it is difficult to detect unintentional stroke or dots in real-time. However, if it is a part of signature of an individual, then it is detected during the training process. But if it is unintentional, then the model does not consider it.

Therefore, the deployment of SignGuard in real-world environments have computational constraints, and uncontrolled signing conditions.

Applications

The OSVS has multiple applications in business, legal, and administrative systems.

Business applications

The SignGuard system can be used to transform authentication and security protocols within the business sector. The primary aim is to validate cheques, loan agreements, and letters of authorization. The high accuracy rate reduces the risk of financial fraud that occurs due to forged signatures. Therefore, it will protect both institutions and their clients from losses. Furthermore, it secures the internal processes, such as approval of purchase orders,

expense reports, and confidential internal documents. The SignGuard enhances operational efficiency by automating and securing the signature verification process. It reduces the need for manual oversight, and design a more robust, trustworthy framework for commercial transactions. It can also be deployed in retail banking for on-the-spot verification of customer signatures against archived specimens, as it streamlines the customer service while bolstering security.

Legal applications
The authenticity of a signature is most important parameter in the legal domain. It validates contracts, wills, affidavits, and other legally binding documents. The SignGuard will be an effective system for forensic document examiners and legal professionals to verify the signatures, along with the visual inspection. It uses robust texture features and hybrid machine learning techniques to minimize human error and bias. It validates if the signature withstands legal scrutiny. It can be proven effective to resolve disputes over wills or contested contracts to prevent lengthy and costly litigation. The law firms and courts can use it to perform preliminary checks on document authenticity. It ensures the integrity of the legal process and upholds the enforceability of agreements.

Administrative applications
The government and public administrative bodies preserve a large volume of paperwork that requires citizen signatures, including application forms and identity documents to permits and benefit claims. The SignGuard verifies the signatures in an offline mode, that makes it ideal for use in various offices and agencies. It prevents the identity theft and welfare fraud, that ensures the delivery of services and benefits to the correct individuals. Therefore, the administrative bodies can automate the verification step that leads to faster processing times and mitigates administrative burdens. It enhances the public trust in government systems. The technology ensures the integrity of official records, i.e., from driver's license applications to voter registration forms. It will safeguard the entire administrative ecosystem against forgery and misrepresentation.

Conclusion and future scope
This section presents the conclusion and future scope.

Conclusion
The offline signature verification system (OSVS) suffers from discrepancies due to intra-signature variability and rotational inconsistencies. This paper presents SignGuard, a novel OSVS to address key challenges in authentication of handwritten signature. It presents the integration of GWO and POA. GWO is used for adaptive preprocessing, while POA is used for rotation correction. SignGuard enhances the robustness of rotation-sensitive descriptors using novel features namely, CS-LBP and OC-CSLBP. The hybrid machine learning framework designed using integrated SVM-XGBoost classifiers further improves classification by combining the advantages of both, i.e., margin optimization using SVM and ensemble learning using XGBoost. SignGuard is validated on CEDAR, SID, BHSig260, and a novel dataset called DeepSignVault. The DeepSignVault comprises signatures of 100 participants, with 10 real and 10 forged signatures of each participant. The SignGaurd is found superior than existing techniques by yielding 98.77% accuracy using OC-CSLBP, and 97.46% accuracy using CS-LBP. It outperforms existing OSVS methods. Thus, it mitigates the forgery attempts while maintaining the computational efficiency.

However, SignGuard have certain limitations. The POA technique for rotation correction is ineffective for signatures with extreme skew angles or complex backgrounds. Although, the GWO technique is effective for feature selection, it increases the computational time during training. Thus, it is not effective for real-time implementation on resource-constrained devices. Furthermore, the $\mathcal{HMLF}$ is not effective across diverse cultural writing styles without extensive retraining.

Future scope
In future, SignGuard can be trained to handle multi-script signature verification, including non-Latin scripts such as Arabic or Chinese. It incorporates language-specific feature descriptors. The model can be enhanced by integrating deep learning architectures, such as Siamese networks or vision transformers, to enhance forgery detection accuracy. It will reduce the dependency on handcrafted features. The optimization techniques like model quantization or neuromorphic computing implementation will enable efficient deployment on edge devices. The system's robustness will be further improved by incorporating adversarial training to counter generative forgeries. Thus, SignGuard will become a versatile solution for global banking, legal, and forensic authentication needs.

Data availability
The datasets used in this paper are either publicly available or created by the authors. The datasets are available online as follows:1) CEDAR – https://www.kaggle.com/datasets/shreelakshmigp/cedardataset2) SID – https://www.kaggle.com/datasets/tienen/handwritten-signature-verification3) BHSig260 – https://www.kaggle.com/datasets/ishanikathuria/handwritten-signature-datasets4) DeepSignVault - https://dx.doi.org/10.21227/xd5x-s582.

Received: 19 June 2025; Accepted: 9 January 2026
Published online: 03 February 2026

References

1. Vargas, J. F., Ferrer, M. A., Travieso, C. & Alonso, J. B. Off-line signature verification based on grey level information using texture features. *Pattern Recognit.* **44** (2), 375–385 (2011).
2. Hafemann, L. G., Sabourin, R. & Oliveira, L. S. Offline handwritten signature verification — Literature review, in *Proc. 7th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, IEEE, 1–8. (2017).
3. Kumar, R., Sharma, J. D. & Chanda, B. Writer-independent off-line signature verification using surroundedness feature. *Pattern Recognit. Lett.* **33** (3), 301–308 (2012).
4. Kumar, A. & Bhatia, K. A survey on offline handwritten signature verification system using writer dependent and independent approaches, in *Proc. 2nd Int. Conf. Adv. Comput., Commun., Autom. (ICACCA)*, 1–6. (2016).
5. Pechwitz, M. et al. IFN/ENIT database of handwritten Arabic words, in *7th Colloque Int. Francophone sur l'Ecrit et le Document (CIFED)*, 129–136. (2002).
6. Mahmoud, S. A. et al. KHATT: Arabic Offline Handwritten Text Database, in *Proc. Int. Conf. Front. Handwrit. Recognit.*, IEEE, 449–454. (2012).
7. Marti, U. V. & Bunke, H. The IAM-database: an english sentence database for offline handwriting recognition. *Int. J. Doc. Anal. Recogn.* **5**, 39–46 (2002).
8. Kleber, F. et al. CVL-DataBase: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting, in *Proc. 12th Int. Conf. Document Anal. Recogn. (ICDAR)*, 560–564. (2013).
9. Schomaker, L. & Vuurpijl, L. A Benchmark Data Set and a Comparison of Two Systems, Tech. Rep., NICI, Nijmegen, (2000).
10. Freitas, C. O. A. et al. Brazilian Forensic Letter Database, in *Proc. 11th Int. Workshop Front. Handwrit. Recogn. (IWFHR-11)*, (2008).
11. He, S., Wiering, M. & Schomaker, L. Junction detection in handwritten documents and its application to writer identification. *Pattern Recogn.* **48** (12), 4036–4048 (2015).
12. Djeddi, C. et al. LAMIS-MSHD: A Multiscript Offline Handwriting Database, in *Proc. 14th Int. Conf. Front. Handwrit. Recogn.*, IEEE, 93–97. (2014).
13. Louloudis, G. et al. ICDAR2013 competition on writer identification, in *Proc. 12th Int. Conf. Document Anal. Recogn.*, IEEE, 1397–1401. (2013).
14. Ojala, T., Pietikäinen, M. & Maenpaa, T. Multiresolution grayscale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24** (7), 971–987 (2002).
15. Tan, X. & Triggs, B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans. Image Process.* **19** (6), 1635–1650 (2010).
16. Ojansivu, V. & Heikkilä, J. Blur Insensitive Texture Classification Using Local Phase Quantization, in *Image Signal Process. (ICISP)*, Springer, 236–243. (2008).
17. He, S. & Schomaker, L. Writer identification using curvature-free features. *Pattern Recogn.* **63**, 451–464 (2017).
18. Chahi, A. et al. Block wise local binary count for off-Line text-independent writer identification. *Expert Syst. Appl.* **93**, 1–14 (2018).
19. Chahi, A. et al. Local gradient full-scale transform patterns based off-line text-independent writer identification. *Appl. Soft Comput.* **92**, 106277 (2020).
20. Muro, C. et al. Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. *Behav. Process.* **88** (3), 192–197 (2011).
21. Juneja, A., Kumar, V. & Singla, S. K. Single Image Dehazing Using Grey Wolf Optimization, in *Congr. Smart Comput. Technol.*, Springer, 333–344. (2022).
22. Ojala, T., Pietikainen, M. & Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell., 24 (7), 971–987
23. R. Ospina et al., Application of nonparametric quantifiers for online handwritten signature verification: A statistical learning approach, *Stat. Anal. Data Min.: ASA Data Sci. J.*, 17 (2), e11673, 2024. (2002).
24. Pal, S. et al. Performance of an Off-Line Signature Verification Method Based on Texture Features on a Large Indic-Script Signature Dataset, in *Proc. 12th IAPR Workshop Document Anal. Syst. (DAS)*, IEEE, 72–77. (2016).
25. Ferrer, M. A. et al. Signature verification using local directional pattern (LDP), in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, IEEE, 336–340. (2010).
26. Ferrer, M. A., Morales, A. & Vargas, J. F. Off-line signature verification using local patterns, in *Proc. CONATEL*, 1–6. (2011).
27. Guerbai, Y., Chibani, Y. & Hadjadji, B. The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters. *Pattern Recognit.* **48** (1), 103–113 (2015).
28. Yilmaz, M. B. et al. Offline signature verification using classifier combination of HOG and LBP features, in *Int. Joint Conf. Biometrics (IJCB). IEEE*, 1–7. (2011).
29. Serdouk, Y. et al. Combination of OC-LBP and Longest Run Features for Off-Line Signature Verification, in *Proc. 10th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, IEEE, 84–88. (2014).
30. Kiani, V. et al. Offline signature verification using local radon transform and support vector machines. *Int. J. Image Process.* **3** (5), 184–194 (2009).
31. Zulkarnain, Z. et al. Triangular geometric feature for offline signature verification. *Int. J. Comput. Electr. Autom. Control Inf. Eng.* **10** (3), 485–488 (2016).
32. Panchal, S. T. & Yerigeri, V. V. Offline signature verification based on geometric feature extraction using artificial neural network. *IOSR J. Electron. Commun. Eng.* **13** (3), 53–59 (2018).
33. Pandya, V. Offline Signature Verification using Clustering Technique, in *Proc. Int. Conf. Pattern Recognit. Artif. Intell.*, ACM, 92–95. (2019).
34. Engin, D. et al. Offline Signature Verification on Real-World Documents, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 3518–3526. (2020).
35. Bertolini, D. et al. Texture-based descriptors for writer identification and verification. *Expert Syst. Appl.* **40** (6), 2069–2080 (2013).
36. Singh, P. et al. Writer identification using texture features: A comparative study. *Comput. Electr. Eng.* **71**, 1–12 (2018).
37. Bahram, T. A texture-based approach for offline writer identification. *J. King Saud Univ. Comput. Inf. Sci.* **34** (8), 5204–5222 (2022).
38. Fadaei, S. et al. An efficient texture descriptor based on local patterns and particle swarm optimization algorithm for face recognition. *J. Supercomput.* **80** (17), 25345–25376 (2024).
39. Ahlawat, S. et al. Offline signature verification using local binary pattern and octave pattern, in *Proc. 5th Int. Conf. Graph. Image Process. (ICGIP)*, 2014, 213–218.
40. Ahlawat, S. et al. Offline signature verification using local binary pattern and octave pattern, in *Proc. 5th Int. Conf. Graph. Image Process. (ICGIP)*, 2014, 213–218. S. Fadaei et al., Content-based image retrieval using multi-scale averaging local binary patterns, *Digit. Signal Process.*, 146, 104391, 2024. (9069).
41. Humeau-Heurtier, A. Texture feature extraction methods: A survey. *IEEE Access.* **7**, 8975–9000 (2019).
42. Prakash, S. G. CEDAR-Dataset, *Kaggle*, [Online]. (2021). Available: https://www.kaggle.com/datasets/shreelakshmigp/cedardataset
43. Ding, L. & Goshtasby, A. On the canny edge detector. *Pattern Recogn.* **34** (3), 721–725 (2001).
44. Mirjalili, S. et al. Grey Wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014).
45. Heikkilä, M. et al. Description of interest regions with local binary patterns. *Pattern Recogn.* **42** (3), 425–436 (2009).
46. Singla, A. & Mittal, A. Exploring offline signature verification techniques: A survey based on methods and future directions. *Multimed Tools Appl.* **84** (6), 2835–2875 (2025).

47. Panja, S. et al. Kernel functions of svm: A comparison and optimal solution, in *Int. Conf. Adv. Informatics Comput. Res.*, Springer, 88–97. (2018).
48. Ahonen, T. et al. Face recognition with local binary patterns, in Comput. Vis. – ECCV 2004, Springer, 469–481. (2004).
49. Lopez, L. S. Local Binary Patterns applied to Face Detection and Recognition, Polytechnic Univ. of Catalonia, [Online]. (2010). Available: https://upcommons.upc.edu/handle/2099.1/10772
50. Kumar, R. et al. A Writer-Independent Off-line Signature Verification System based on Signature Morphology, in *Proc. Int. Conf. Intell. Interact. Technol. Multimedia*, 261–265. (2010).
51. Rexit, A. et al. Multilingual handwritten signature recognition based on High-Dimensional feature fusion. *Information* **13** (10), 496 (2022).
52. Batool, F. E. et al. Offline signature verification system: a novel technique of fusion of GLCM and geometric features using SVM, *Multimed. Tools Appl.*, 1–20.
53. X. Zhao et al., Fusing deep and hand-crafted features by deep canonically correlated contractive autoencoder for offline signature verification, *Pattern Recogn.*, 111834, 2025.
54. M. R. Swamy et al., Deep learning approaches for online signature authentication: a comparative study of pre-trained CNN models, *Eng. Res. Express*, 7 (1), 015230, 2025. (2024).
55. Malik, M. I. et al. Evaluation of local and global features for offline signature verification. *CEUR Workshop ProcAppl Sci.* **76811**, 26–30 (20112021).
56. Y. Zhou et al., "Handwritten signature verification method based on improved combined features," *Appl. Sci.*, 11, 5867, 2021.
57. Mashhadani, S. S. et al. Fusion of Type-2 neutrosophic similarity measure in signatures verification systems: A new forensic document analysis paradigm. *Intell. Autom. Soft Comput.* **39** (5), 1–17 (2024).
58. Abdulhussien, A. A. et al. One-class Arabic signature verification: A progressive fusion of optimal features. *Comput. Mater. Contin.* **75** (1), 219–242 (2023).
59. Pashkin, Z. Handwritten signature verification, *Kaggle.com*, (2022). https://www.kaggle.com/datasets/tienen/handwritten-signature-verification (accessed Sep. 20, 2025).
60. Pal, S., Alaei, A., Pal, U. & Blumenstein, M. Performance of an Off-line Signature Verification Method Based on Texture Features on a Large Indic-script Signature Dataset, *Proc. 12th IAPR Workshop on Document Analysis Systems (DAS)*, 1–6, (2016).
61. Goyal, P. et al. DeepSignVault: A comprehensive offline signature image database for biometric authentication and deep learning applications. *IEEE Dataport Oct.* **22** https://doi.org/10.21227/xd5x-s582 (2025).

Author contributions

Author Contributions: Conceptualization, N.C.R, A.J., N.K.,V.K., & A.D.; methodology, N.C.R, A.J., N.K.; software, N.C.R, A.J., N.K.,V.K.; validation, N.K.,V.K., & A.D.; formal analysis, N.K.,V.K., & A.D.;investigation, V.K., & A.D.; resources, N.K.,V.K., & A.D.; data curation, V.K., & A.D.; writing—original draft preparation, N.C.R, A.J., N.K.,V.K.; writing—review and editing, N.C.R, A.J., N.K.,V.K., & A.D.; visualization, V.K., & A.D.; project administration, V.K., & A.D.; funding acquisition, A.D. All authors have read and agreed to the published version of the manuscript.

Funding

Open access funding provided by Manipal University Jaipur. Manipal University Jaipur, India.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to V.K. or A.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.