

Article

A Genetic Algorithm Based One Class Support Vector Machine Model for Arabic Skilled Forgery Signature Verification

Ansam A. Abdulhussien ^{1,2,*} , Mohammad F. Nasrudin ¹ , Saad M. Darwish ³ 
and Zaid Abdi Alkareem Alyasseri ^{4,5} 

¹ Centre of Artificial Intelligence, Faculty of Information Sciences and Technology,
University Kebangsaan Malaysia, Bangi 50300, Malaysia; mfn@ukm.edu.my

² Information Technology Center, Iraqi Commission for Computers and Informatics, Baghdad 10009, Iraq

³ Institute of Graduate Studies and Research, University of Alexandria, Alexandria 21526, Egypt;
saad.darwish@alexu.edu.eg

⁴ Information Technology Research and Development Center (ITRDC), University of Kufa, Najaf 540011, Iraq;
zaid.alyasseri@uokufa.edu.iq

⁵ College of Engineering, University of Warith Al-Anbiyaa, Karbala 56001, Iraq

* Correspondence: p102907@siswa.ukm.edu.my

Abstract: Recently, signature verification systems have been widely adopted for verifying individuals based on their handwritten signatures, especially in forensic and commercial transactions. Generally, feature extraction and classification tremendously impact the accuracy of system authentication. Feature extraction is challenging for signature verification systems due to the diverse forms of signatures and sample circumstances. Current signature verification techniques demonstrate promising results in identifying genuine and forged signatures. However, the overall performance of skilled forgery detection remains rigid to deliver high contentment. Furthermore, most of the current signature verification techniques demand a large number of learning samples to increase verification accuracy. This is the primary disadvantage of using deep learning, as the figure of signature samples is mainly restricted to the functional application of the signature verification system. In addition, the system inputs are scanned signatures that comprise noisy pixels, a complicated background, blurriness, and contrast decay. The main challenge has been attaining a balance between noise and data loss, since some essential information is lost during preprocessing, probably influencing the subsequent stages of the system. This paper tackles the aforementioned issues by presenting four main steps: preprocessing, multifeature fusion, discriminant feature selection using a genetic algorithm based on one class support vector machine (OCSVM-GA), and a one-class learning strategy to address imbalanced signature data in the practical application of a signature verification system. The suggested method employs three databases of signatures: SID-Arabic handwritten signatures, CEDAR, and UTSIG. Experimental results depict that the proposed approach outperforms current systems in terms of false acceptance rate (FAR), false rejection rate (FRR), and equal error rate (EER).

Keywords: offline signature verification system; preprocessing; feature fusion; forgery detection; Arabic signature; one-class support vector machine



Citation: Abdulhussien, A.A.; Nasrudin, M.F.; Darwish, S.M.; Alyasseri, Z.A.A. A Genetic Algorithm Based One Class Support Vector Machine Model for Arabic Skilled Forgery Signature Verification. *J. Imaging* **2023**, *9*, 79. <https://doi.org/10.3390/jimaging9040079>

Academic Editor: Silvia Liberata Ullo

Received: 16 December 2022

Revised: 25 February 2023

Accepted: 9 March 2023

Published: 29 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A signature is one of the most important human attributes. It is often used as proof of identity on legal documents like bank checks, credit cards, and wills. An effective automatic system can handle many fraud issues and other daily crimes. There are two different kinds of signature verification scenarios: online and offline. An online signature verification system uses tablets, PDAs, iPads, and smartphones to evaluate the signature image. The system has a dynamic nature, operating on features such as writing, orientation, pen tip positions, momentum, velocity, pressure, etc. [1,2].

An offline verification validates signatures by employing an optical detector to collect signatures on paper. This approach contains static data such as inclination, boundary, signature length and altitude, baseline, pressure, and size [3]. Offline verification is more intricate than online verification due to the absence of dynamic parameter information. Moreover, in an OSV system, signatures are obtained from various devices, so the resolution of training and testing samples is not the same, resulting in intraclass variation [4]. The raw signature may contain additional pixels known as noises or may not be in perfect working order, necessitating preprocessing.

Additionally, variations in the original signer's signature are related to document orientation, signature degeneration, illness, illegible signatures, pen width, age, etc. As a result, preprocessing is a fundamental stage to improve the input data or isolate raw data samples into a standard format appropriate for the feature extraction stage. Nevertheless, it is crucial to balance noise removal and data loss, since a certain amount of relevant information may be lost during preprocessing, impairing the accuracy of subsequent system stages [5]. Therefore, proposing an algorithm that can effectively remove noise while maintaining relevant information is crucial.

Generally, the handwritten signature verification system determines whether the query signature is genuine or forged [2]. There are three forms of forgery: random forgery, simple (unskilled), and skilled forgery. In random forging, the forger uses a signature without knowing the original user's name or signature. This signature shape is distinct from the genuine signature. The simple forger knows the user's name, but is uninformed of the signature's pattern. In this case, the forms of the original and forged signatures may not be similar.

On the other hand, the skilled forger learns the signature form and professionally mimics the signature with practice. This form of imitation is more challenging to detect since it is comparable to an authentic signature [6]. Figure 1 shows the samples of signatures. The most challenging aspect of the signature verification process is the substantial intra-class variance across signatures from the same individual, in contrast to low intra-class between forgery and genuine. Second, no comprehensive system can recognize all scripts.

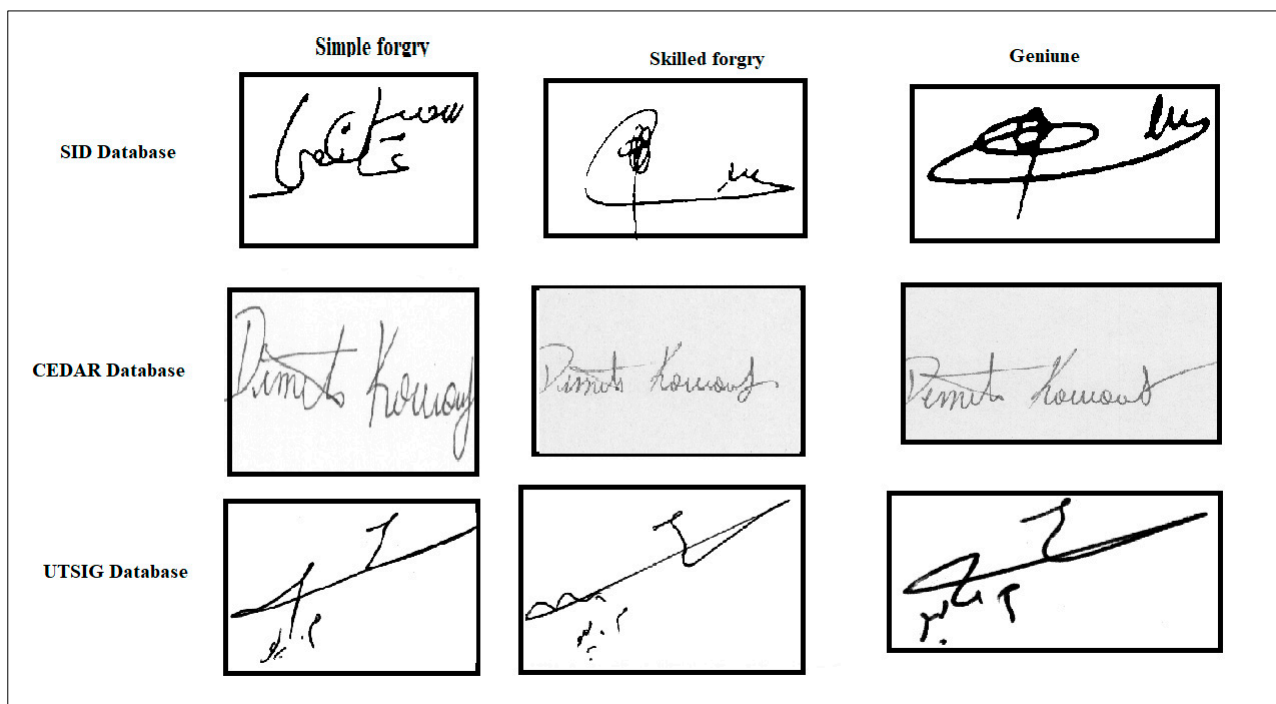


Figure 1. Sample of signatures.

According to the literature, signature verification systems have been the subject of substantial study in several languages, including English, Hindi, Bangla, and Chinese. However, studies on Arabic signatures are still limited, and improvement is slow because the language is more difficult to analyze than others. Arabic script is characterized by its cursive form, changing letter shapes, joining and non-joining characters, delayed strokes, and ligatures [7,8].

Recently, researchers have focused on two major processes: feature extraction and verification methods. Feature extraction methods depended on handcraft, such as statistical, geometric, structural, and automatic deep learning. In several cases, researchers have combined multiple methods to improve performance. Fusion is the process of integrating multiple patterns of components into a single matrix using a fusion approach, such as score-level fusion and high-priority index feature fusion [9].

The problem with earlier approaches was that they typically employed a combination of features without considering correlations and discriminants, such that the generated feature vector could not recognize a skilled forgery. In addition, fusion techniques were used to increase system complexity and computational time. Moreover, the current deep learning approaches are defective in the signature verification domain because deep training models necessitate large data samples and effort in selecting images suitable for learning construction [8]. Deep learning also necessitates substantial computer resources, such as expensive GPUs.

This research aims to propose an offline Arabic signature verification system that can recognize skilled forgery and genuine signatures at a highly accurate rate and low FAR, FRR, and EER. First, efficient preprocessing techniques such as image cropping, denoising, binarization, and removing stray isolated pixels are recommended to decrease noise while maintaining essential data.

The contributions of this study include the following:

1. Efficient preprocessing techniques are recommended to decrease noise while maintaining essential data.
2. Hybrid feature types have been proposed to solve the low inter-class variability between authentic and skilled forgery and the high intra-class variability in each individual's signature.
3. The early serial concatenation fusion approach (ESCF) integrates multiscale information without prejudice complication.
4. Propose GA_OCSVM to improve feature selection and tackle the potential correlation between fused features
5. Settle the problem of unbalanced and restricted forgery samples by using one-class classification.

The extensive computing time and storage capacity are unnecessary for the proposed approach.

This paper has the following organization: Section 2 presents the related studies. The framework for the proposed signature verification system is given in Section 3. The experimental results and comparisons of the proposed methodology to previous research are presented in Section 4. The summary of the proposed work is presented in Section 5. The conclusion and recommendations for further work are provided in Section 6.

2. Related Works

Signatures are typically simple or unconventional, and have no distinct characteristics that are hard to lose or forget compared to other biometric features [10]. Consequently, signatures on checks, card payments, and legal documents are often used and accepted as evidence of authorship or approval. Signatures are currently authenticated in various environments [11]; however, the rapid progress of computer technology has attracted the attention of researchers in automated signature verification and authenticity detection. OSV has significantly evolved in the last decade; researchers have employed various methodologies and techniques to accomplish high performance, superior accuracy, and

efficiency in offline signature verification. Signature verification methods are typically divided into template matching, statistical, and structural approaches [12].

In the template matching approach, the pattern of test signatures is compared with templates already stored in the database. Dynamic time warping (DTW) is the most often utilized for this purpose, ref. [13] proposed a crowdsourcing experiment to develop a human baseline for signature recognition and a new attribute-based automated signature verification system based on FDE analysis. The technique combines the DTW algorithm with an attribute-based approach to improve accuracy with 5% EER. The authors of [14] proposed a graph-based system for signature verification. This approach combines DTW with linear time graph dissimilarity to measure the polar graph embedding distance (PGEd) called structural DTW (SDTW). They used a sliding window approach to compare PGEd at various local positions on several subgraphs. The resulting distance matrix was used to find an optimal alignment between the sequences of subgraphs using DTW. The authors applied the proposed method to standard GPDS-75 and MCYT-75 datasets.

However, statistical models are employed in the vast preponderance of signature verification systems, such as distance-based classification, support vector machine (SVM), deep learning, and other classification techniques. The distance-based approach is one of the most straightforward and reliable approaches for identifying query and reference signatures because this approach lacks parameters and model training [15]. Nevertheless, distance-based techniques are not interested in the influence of general variability on distance and often exhibit random fluctuations of varying sizes. The most prominent methods used in this domain are Euclidean distance, city block distance, Chi-square distance, Manhattan distance, and Hamming distance. On the SUSIG dataset, a Hadamard transform-based technique was developed [16]. The Hadamard matrix was generated from the extracted features, and then the Euclidean and Manhattan distances were employed for feature comparison and verification.

In [17], the Euclidian distance was employed to compare stored and new feature vectors. The investigation used one hundred and eight signature samples from participants. Global thresholding was used to convert images to grayscale, and the median filter was utilized to remove noise. The Canny edge detector was employed to identify signature edges. Seven hundred moments of invariants were calculated for five samples, and the standard deviation was used to generate a feature vector.

Similarly, SVM and deep learning techniques such as convolutional neural networks (CNNs) are the most often used classifiers in OSV. SVM performs well in high-dimensional spaces, regardless of whether the dimensionality exceeds the sample quantity. It is memory efficient because it uses a subset of training images (support vectors) in the decision function [18]. However, SVM is mathematically and computationally complex. Ref. [19] used the SVM and shape correspondence approaches for signature verification. Pixels were correlated using an adaptive weight that included Euclidean and shape context distances. Plate spline transformation was used to convert the query signature plane to the reference signature plane. On the GPDS signature dataset, the system achieved an accuracy of 89.58%. The authors in [19] used a decision tree classifier and a Local Binary Pattern feature extraction. Two collected datasets with 100 and 260 authors were employed to evaluate the performance of the system. The system produced a FAR of 7.0% and 11% for simple and skilled forging signatures, respectively.

Moreover, the authors in [20] introduced a dynamic signature verification technique (DSVT) using mutual compliance (MC) between the security system and the biometric device. The security system was responsible for online and offline signature approval using personal inputs from the user. The signing bit, key, and size were used as security metrics to verify both modes using classifier learning. The verification was based on stored online/offline signatures using certificates provided for authentication.

The E-signature was conducted based on the user's specific inputs. The user authenticity was examined based on stored online/offline signatures using certificates and authentication during manual sessions. A traditional tree classifier was used to distinguish

the dynamic verification between online and offline signatures. The success rate of the suggested strategy was 0.893%, while the failure rate was 8.58%.

The [21] compared SVM with five machine-learning classifiers, i.e., boosted tree, random forest classifier (RFC), K-nearest neighbor, multilayer Perceptron, and naïve Bayes classifier, utilizing four image-based characteristics. The BHsig260 dataset (Bangla and Hindi) was used in the proposed work, which included signatures from 55 Hindi and Bangla users. The offline Hindi signature verification accuracy using MLP with 20 sample sizes was 72.3%. The accuracy for Bangla was 79% using RFC with two signature samples, while KNN and SVM obtained above 92%.

In addition, various deep-learning techniques have been proposed for online and offline signature verification. In the offline signatures system, ref. [22] employed CNNs in a two-stage method. Feature representations were learned in the writer-independent phase by discriminatively training a CNN to identify authors. These CNN characteristics were then utilized for training writer-dependent classifiers (SVMs) to recognize differences between genuine and skilled signatures. Moreover, they tested this method using four distinct feature representation versions of AlexNet and VGG networks [23]. Kohonen neural networks were proposed to construct an offline signature verification system, which was a form of self-organizing map [24]. The intra-variability of an individual's signatures is quantified using their competitive learning power. The proposed system achieved FAR and FRR for the genuine samples of 2.8% and 5%, respectively, for simple and random forgeries.

The researchers in [25] also used CNN to verify a Bengali handwritten signature. Two handwritten signature databases were used as experimental data for the training system. The first database contained 800 handwritten signature images of 40 students at the Fergana branch of the Muhammad al-Khwarizmi Tashkent University; each student had 10 genuine and 10 forged signatures. The second database was a public Bengali handwritten signature database, which included 100 people with 24 authentic and 30 skilled signatures. The average accuracy achieved for the first database was 90.04% on images of size 250×150 , and 97.50% for the second database on images of size 250×150 .

The researchers in [20] proposed an offline signature verification system using a multi-size assembled attention swin-transformer (MSAAST) network. The main modules included the resize, swin-transformer, and attention block. The signature images were resized to different sizes, including (224, 224), (112, 112), and (56, 56). Then, they were simultaneously put into the Patch-Embedded module and swin-transformer to extract and combine features. The cross-dataset strategies were used to improve the dataset; considering the generalization ability, CEDAR was utilized as a training dataset and evaluated in Bengali. Three databases were used to assess the model: CEDAR, Bengali, and Hindi. The training and testing datasets extended double, and images were concatenated in combination forms: genuine-genuine signature pairs (GGSP) or genuine-forgery signature pairs. The regularized dropout (R-Drop) strategy and adversarial methods were employed in the training phase to improve the verification performance. The authors used the R-Drop strategy to limit the model's outputs and keep them in identical distributions even when the inputs were run through the model more than once. The accuracy metric significantly increased from 0.955 to 0.973. However, in the experiment on R-Drop, the dropout produced different outputs for the same input images each time.

Despite the tremendous achievements of deep learning in signature identification, one of the significant downsides of deep learning models is that they need a massive amount of labeled data for training to obtain a high level of accuracy. Most signature databases are limited (particularly concerning the number of original signatures per writer). This limitation faced the authors in [26], who used samples from the SVC 2004 and SigComp 2009 datasets to learn a convolutional neural network (CNN) followed by a recurrent neural network (RNN). The proposed model achieved low validation results due to the few samples used; the experiments showed 90.65% accuracy and 15.43% FAR.

In contrast to statistical representation models, structural (i.e., string, tree, and graph-based) techniques express the fundamental topological features of a handwritten signature

in a highly natural and exhaustive form. This model compares the symbolic representation (trees, graphs, and strings) to database-stored models. However, this advantage comes at the expense of increased complexity in basic dissimilarity assessments [27]. The authors in [28] focused on dissimilarity-based graph embedding techniques for signature verification. It generated n-dimensional feature representations for graphs, which were then used to classify signatures. In an experimental assessment of the MCYT-75 and GPDS-960 benchmark datasets, the suggested technique achieved 10.67% EER and 16.53 EER using 10 references.

In addition to being accurate and secure, the signature verification process should be fast. Furthermore, signature verification is complicated since the distinctions used to discriminate are frequently precise. As a result, offline signature recognition is still open research. Table 1 displays an overview of related works and their respective outcomes.

Table 1. Summary of related work.

References	Features Used	Verification Approaches	Accuracy/EER
[29]	Global features and center of gravity features	Threshold technique	87% on the GPDS-960 database
[30]	Global features.	fuzzy-C means + threshold	9.2% EER on MCYT database
[31]	Global feature (entropy) and functional information features	SVM	97.81% n SVC2004 database
[23]	Deep CNN	SVM	12.83% EER on (GPDS) database 4.17% on (BRAZILIAN PUC-PR) database
[32]	Median of Medians (MoM) statistical dispersion measure (Δx)	Fuzzy similarity between test and training signature sample and threshold technique	0.11 ERR on (MCYT-100) database
			0.088 ERR on MCYT-330) database
			0.916 ERR on SVC database
			0.08 ERR on SUSIG database
[33]	Condensed Nearest Neighbors (CNN)	SVM	3.46% EER GPDS-960 dataset
[34]	global and grid features belonging	feature dimension and decision threshold	7.66% ERR on CEDAR database 9.53 on MCYT database
[35]	Meta learning		4.70 ERR on GPDS dataset
			12.77 ERR on MCYT
			8.02 ERR on CEDAR
			6.7 ERR on Brazilian
[36]	Gray Level Co-occurrences Matrix (GLCM) and geometric features	SVM	2.33% MCYT, 9.59% EER on GPDS synthetic
[37]	Structure- and direction-oriented features	Recurrent Neural Network (RNN)	GPDS-300 98.02%
			MCYT-75 99.39%
			BHSig260 Hindi 99.28%
			BHSig260 Bengali 99.37%

Table 1. Cont.

References	Features Used	Verification Approaches	Accuracy/EER
[38]	CNN method transfers learning SIFT + SVM		99.94% on
			SVM, 98.1 on 112 images were from IDRBT bank cheque dataset, used 50 images for testing
[39]	CNN		88% on signatures of 100 people, included 24 genuine signatures and 30 forged signatures

3. Materials and Methods

The proposed model comprises five phases: preprocessing approaches, feature extraction, feature fusion, feature selection, and classification, as shown in Figure 2.

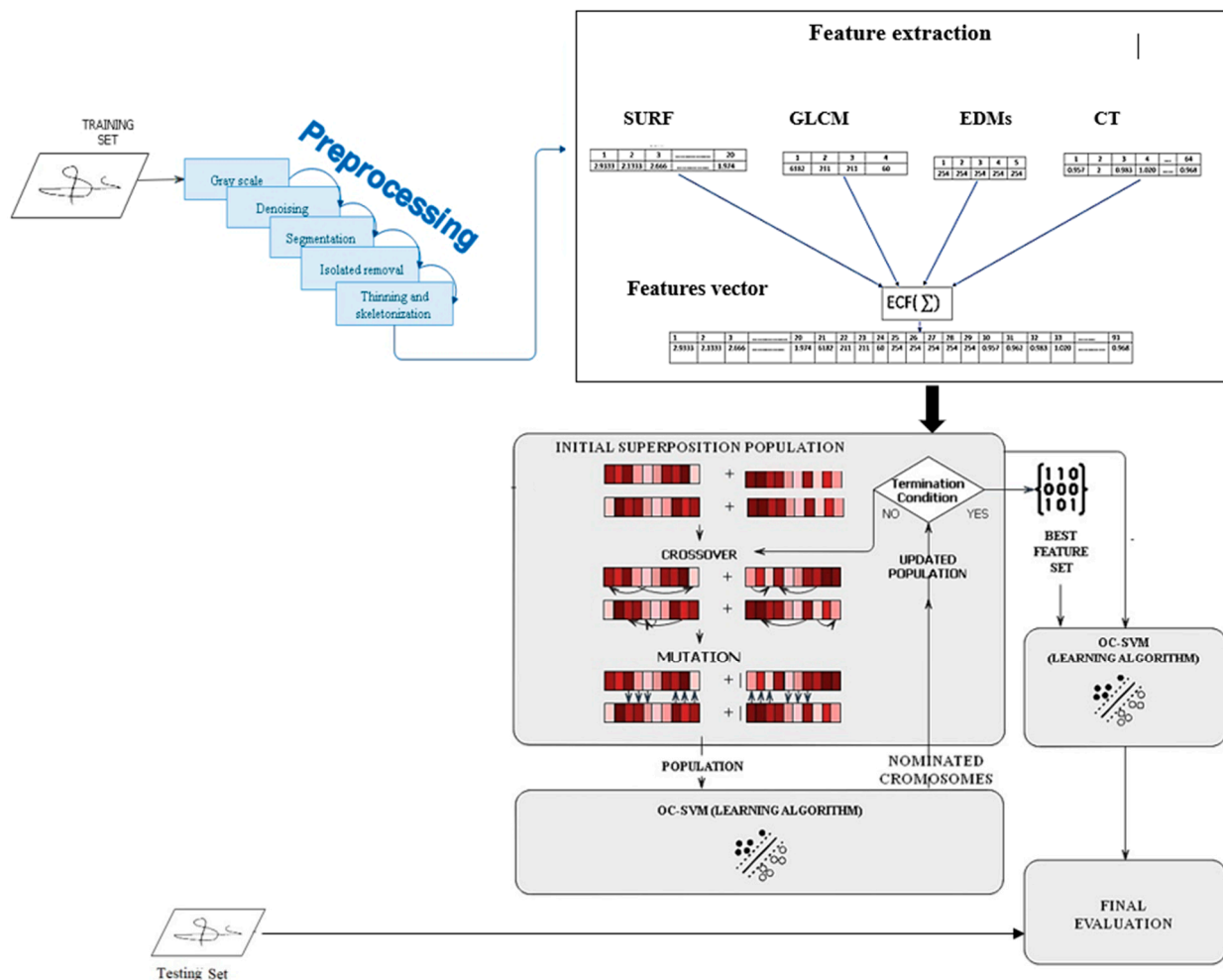


Figure 2. Flowchart of the proposed model.

3.1. Preprocessing Phase

A review of the problems related to offline signatures may include noisy pixels or equipment that may not be in perfect working order. As a result, several preprocessing methods are presented to provide an improved image that can be utilized for subsequent phases without losing data.

3.1.1. Image Conversion

The first step in the proposed approach is to convert an RGB image to a grayscale image, which is required to decrease system complexity and processing time because grayscale images are simpler to modify than RGB images.

3.1.2. Noise Reduction

The scanner or the paper backdrop might produce noise in a scanned image; the image may become fuzzy due to insufficient illumination and stained regions, such as dots and speckles. The image filtering technique improves the image by converting irrelevant brightness information into valuable data, and is easily understandable and concentrated on machine interactions. The median filtering (MF) strategy is utilized in this work to remove noise from the signature image. The MF technique [40] is a statistically based nonlinear method for reducing image noise. Applying a linear low-pass filter is the preferred approach for smoothing, which is the appropriate method in a static signature. The MF has the following two key advantages:

MF retains sharp edges, whereas low-pass linear filtering softens the edges.

MF is quite effective in smoothing down a noise spike.

MF retains the pertinent information of the image and changes the original grey value of each pixel to the median gray value of the area of the neighborhood. This filter reduces visual noise without causing edge blurring. The median is calculated by sorting the pixel values of the neighborhood window and substituting the considered pixel with the middle (*median*) value. The formula for the MF image $D(x, y)$ of image $I(m, n)$ is represented as (1).

$$D(x, y) = \text{median}_{(m, n) \in R_{xy}} \{I(m, n)\} \quad (1)$$

where $m, n \in$ center around the processed pixel (x, y) .

3.1.3. Binarization

This process transforms a grayscale image into a binary image. Image binarization is the earliest step of image processing and analysis. Pixels in an image are separated into two different areas, black and white. The main goal of image binarization is to be able to describe the difference between text in the foreground and text in the background. The thresholding technique is the simplest type of binarization. In thresholding, pixels are identified as foreground or background by comparing them to the maximum threshold value. However, determining the optimal threshold value for such signature text is challenging. Inaccurate estimation of the threshold value leads to the erroneous classification of pixels as foreground or background, which affects binarization results and the accuracy of signature authentication.

In this research, the backdrop of an image is estimated using the grayscale morphological method [41]. The contrast of the image text regions is boosted using the approximate background data. A recognition threshold value for image sections is determined by analyzing the histogram of the contrast image. In image processing, morphology can be applied to two types of pixel sets: objects and structural elements (SEs). Objects are described as collections of foreground pixels. SEs are created using both foreground and background pixels.

The size of the SEs is first determined and calculated using the histogram of the distance between consecutive edges. The morphological processing techniques include dilation and erosion. SEs generate both dilation and erosion by interacting with a collection of exciting pixels in an image. The SEs have a morphology and an origin. $A \oplus B$ denotes dilation, which is the collection of all shifts satisfying the condition in Equation (2):

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \quad (2)$$

where A is a set of foreground pixels, B is SEs, (\hat{B}) is the reflection of the structuring B about its origin, followed by a shift by z , and z 's are foreground values (one's). The erosion represented by the symbol $A \ominus B$ is defined as Equation (3):

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (3)$$

3.1.4. Image Segmentation

Segmentation is used to extract the signature region from an image. This procedure decreases the processing time by deleting the excess pixels of the image. In this work, an automated segmentation technique is computed using the histogram; the region is automatically segmented based on pixel values. The histogram depicts the total amount of black-and-white pixels [42]. The distribution is horizontally and vertically separated. The white pixel's highest point is utilized as a trimming reference. The image is divided in half to simplify the recovery of the starting and ending points.

Consequently, two points denote the beginning and end of cutting originating from the highest point. The horizontal histogram determines the starting and ending positions of the horizontal trim. Additionally, vertical cutting uses the vertical histogram to calculate the beginning and ending locations. Equations (4)–(6) generate histograms:

$$H_X = \sum_{y=1}^n bim(x, y) \quad (4)$$

$$H_Y = \sum_{x=1}^m bim(x, y) \quad (5)$$

$$imcrop = \begin{cases} origin_{image} & X_{max} < X < X_{min} \text{ and } Y_{max} < Y < Y_{min} \\ 0 & otherwise \end{cases} \quad (6)$$

bim denotes the binary image, while m and n indicate the matrix bim 's rows and columns, respectively.

3.1.5. Stray Isolated Pixel Elimination

In some signatures, extra points caused by ink flowing unrelated to the signature may affect the original signature area. Consequently, the MATLAB function eliminates any connected components (objects) with less than 50 pixels from the binary image $B(x, y)$. This procedure is known as an area opening, as shown in Equation (7).

$$B_{new}(x, y) = bwareaopen(B(x, y)) \quad (7)$$

3.1.6. Skeletonization and Thinning

Thinning is an iterative process that results in skeleton production. This procedure minimizes the number of character characteristics to aid feature extraction and classification by erasing the width fluctuations of the pen. Applying a specific morphological operation to the binary image B , a fast parallel thinning method (FPT) removes inside pixels to leave an outline of the signature [43]. The FPT approach extracts the skeleton from an image by removing all contour points except those relevant to the skeleton. As illustrated in Figure 3, each point $p(i, j)$ has eight neighbors.

$P9(i-1, j-1)$	$P2(i-1, j)$	$P3(i-1, j+1)$
$P8(i, j-1)$	$P1(i, j)$	$P4(i, j+1)$
$P7(i+1, j-1)$	$P6(i+1, j)$	$P5(i+1, j+1)$

Figure 3. Designations of the pixels in a 3×3 window.

Each iteration is separated into two subiterations to preserve the structure of the skeleton. See Algorithm 1 and Figure 4.

Algorithm 1: FPT

- 1: $A(P_1)$ is the number of (01) patterns in the ordered set $P_2, P_3, P_4, \dots, P_8, P_9$ that are the eight neighbors of P_1
 - 2: $B(P_1)$ is the number of nonzero neighbors of P_1
 - 3: $B(P_1) = \sum_{i=2}^9 P_i$
 - 4: Iteration 1: $P_1 = 0$
 - If $2 \leq B(P_1) \leq 6$
 - If $A(P_1) = 1$
 - If $P_2 \times P_4 \times P_6 = 0$
 - If $P_4 \times P_6 \times P_8 = 0$
 - Else
 - $P_1 = 1$
 - $A(P_1) = 2$
 - 5: Iteration 2:
 - $P_2 \times P_6 \times P_8 = 0, P_2 \times P_4 \times P_8 = 0$
 - Keep the rest points
 - 6: End
-

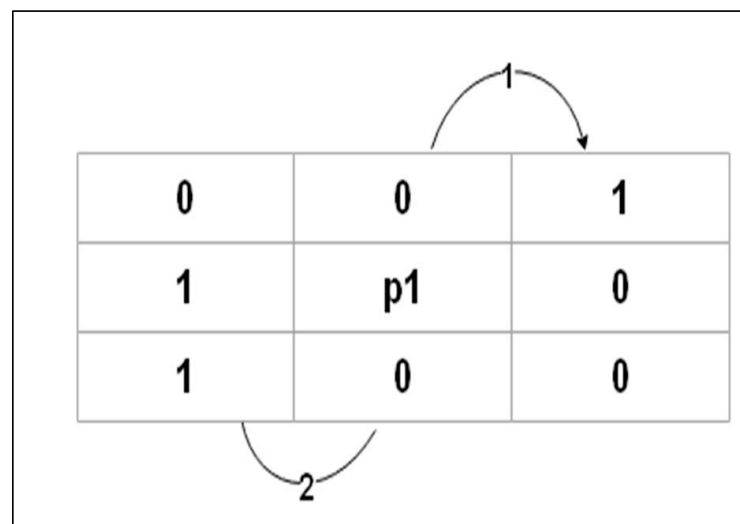


Figure 4. Counting the 01 patterns in the ordered set P_2, \dots, P_9 .

Skeletonization is achieved by removing specific foreground pixels from a binary image through image thinning. Consequently, a collection of tiny arcs and curves portrays the signature pattern. The complement of the image is adjusted to make the signature bright and the background dark to skeletonize the original image. See Figures 5 and 6.

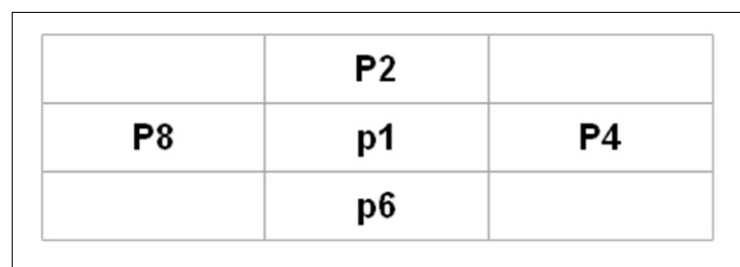


Figure 5. Locations of points that satisfy the conditions.

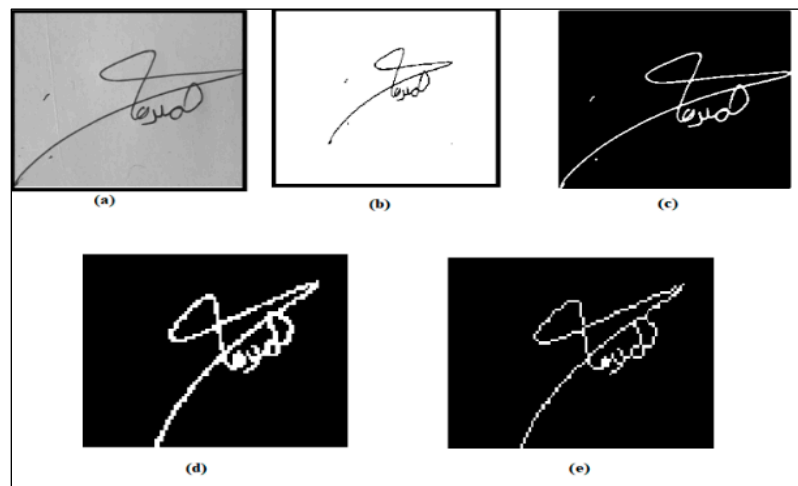


Figure 6. Preprocessing steps (a) grayscale image, (b) image denoising, (c) segmentation, (d) isolated removal, and (e) thinning and skeletonization.

3.2. Hybrid Feature Extraction

Feature extraction is a crucial step in the verification of a signature. The proposed Hybrid Statistical Feature Extraction (HSFE) technique extracts highly informative features by combining multiple types of features using three statistical approaches: interest point features, global and local texture features, and curvelet transformation features.

3.2.1. Texture Feature

In image processing, the texture is described as a function of the spatial variation of the brightness intensity of the pixel. Image processing is the primary term to define objects or concepts in a given image. Texture analysis is critical in computer vision applications such as object recognition, surface defect detection, pattern recognition, and medical image analysis. This paper combines two statistical methods of edge direction matrices (EDMs) [44] and local binary pattern (LBP) [45] to extract texture attributes.

LBP features are also known as the texture operator for a grayscale image, which helps to characterize the spatial structure of the input image texture. Once the central pixel value is obtained, the pattern code can be computed by comparing these values to its neighborhoods. It can be expressed as Equation (1).

$$LBP_{N,R} = \sum_{n=0}^{N-1} s(I_n - I_g)^{2N} \quad (8)$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

where I_g denotes the gray value of the center pixel, I_n represents the gray values of the circularly symmetrical neighborhood, and N denotes the total number of spaced pixels on a circle of radius R . The final texture feature employed in texture analysis is the histogram of the operator outputs (i.e., pattern labels) accumulated over a texture sample. The operator for grayscale and rotation-invariance texture description is shown in Equations (9) and (10).

$$LBP_{N,R} = \begin{cases} \frac{\sum_{n=0}^{N-1} s(I_n - I_g)}{N+1} & \text{if } ULBP_{N,R} \leq 2 \\ \text{otherwise} & \end{cases} \quad (9)$$

where

$$ULBP_{N,R} = |s(I_{N-1} - I_g) - s(I_0 - I_g)| + \sum_{n=1}^{N-1} |s(I_n - I_g) - s(I_{n-1} - I_g)| \quad (10)$$

However, LBP cannot provide information about shape; that is, the spatial relationships of pixels in an image. As a result, LBP is combined with EDMS. The global features

are the features that result from the shape of a signature contour [45]. EDMS is a feature extraction approach that detects the texture of a binary image $I(x, y)$ based on edge-to-neighbor pixel relationships. Eight adjoining kernel matrices were applied, and each pixel was linked to two neighboring pixels. A connection was established between the edge pixel $E(x, y)$ and its neighboring pixels, as illustrated in Figure 7a. The eight pixels were used to change the surrounding values into the position values, as shown in Figure 7b.

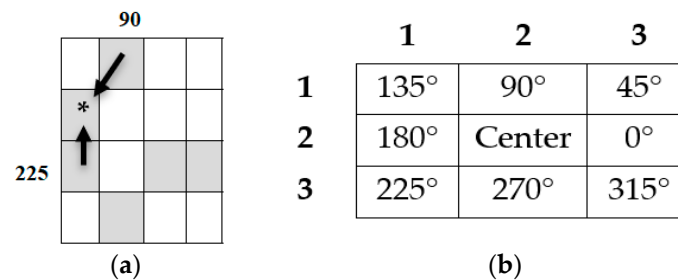


Figure 7. (a) Two neighboring edge pixels, (b) EDMs principal.

This approach is presented from two perspectives: first-order relationship (FOR) identification and second-order relationship (SOR) identification. Each cell in the FOR matrix has a location between 0 and 315 degrees, depending on the pixel neighborhood association. The relationship between the pixel values can be determined by computing the occurrence of the FOR values while considering the edge image of each pixel concerning two other pixels.

The relationships are sorted according to their priority by ordering the values in FOR in descending order. Subsequently, the highest-order relationships are selected, and the others are disregarded. The acquired relationships are computed and saved in the SOR cell. Algorithms 2 and 3 provide critical statistical features, including data attributes and distribution descriptions.

Algorithm 2: FOR

```

1: for each pixel in  $(E(x, y))$ 
2: If  $p(x, y) = 0$  {Black pixel at center} Then
  Increase the frequency of occurrences at FOR (2,2) by 1
3: If  $p(x + 1, y) = 0$  {Black pixel at 0°} Then,
  Increase the frequency of occurrences at FOR (2,3) by 1
4: If  $p(x + 1, y - 1) = 0$  {Black pixel at 45°} Then
  Increase the frequency of occurrences at FOR (3,1) by 1
5: If  $p(x, y - 1) = 0$  {Black pixel at 90°} Then
  Increase the frequency of occurrences at FOR (2,1) by 1
6: If  $p(x - 1, y - 1) = 0$  {Black pixel at 135°} Then
  Increase the frequency of occurrences at FOR (1,1) by 1
7: If  $p(x, y - 1) = 0$  {Black pixel at 180°} Then
  Increase the frequency of occurrences at FOR (2,3) by 1
8: If  $p(x - 1, y + 1) = 0$  {Black pixel at 225°} Then
  Increase the frequency of occurrences at FOR (3,1) by 1
9: If  $p(x, y + 1) = 0$  {Black pixel at 270°} Then
  Increase the frequency of occurrences at FOR (2,1) by 1
10: If  $p(x + 1, y + 1) = 0$  {Black pixel at 315°} Then
  Increase the frequency of occurrences at FOR (1,1) by 1
End

```

Algorithm 3: SOR

```

1: Sort  $R_1 = \text{FOR}(x, y) \downarrow$ 
2: For each pixel in  $(E(x, y))$ 
3: If  $E(x, y) = \text{Black}$  Then
     $R_2 = \text{Relationships of neighborhood two pixels in } E(x, y)$ 
4: Compare  $(R_1, R_2)$ 
5: Connected cell in  $\text{SOR} = \text{SOR} + 1$ ,
End

```

3.2.2. Interest Point Features

This work uses the speeded up robust feature (SURF) to identify an image's interesting points. SURF is a resilient representation approach invariant to translation, rotation, and scaling. This descriptor is used to find the similarity between different interesting points. The entry of an integral image $\int(x, y)$ at a location $(x, y)^T$ is used to represent the sum of all pixels in the input image $I(x, y)$ within a rectangular region formed by the origin and $(x, y)^T$. See Equation (11).

$$\int(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (11)$$

Additionally, the Hessian matrix is used to identify blob-like formations at regions where the determinant is optimal. The Hessian matrix $H(p, \sigma)$ at point $p = (x, y)^T$ and scale $\sigma = 3$ is shown as Equation (12):

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix} \quad (12)$$

where $L_{xx}(p, \sigma)$ is the convolution of the second-order derivative of the Gaussian $\frac{\partial^2 g(x, \sigma)}{\partial x^2}$ with image $\int(x, y)$ at point p and is similar to $L_{xy}(p, \sigma)$ and $L_{yy}(p, \sigma)$.

3.2.3. Curvelet Transformation (CT)

CT is a multiscale pyramid with several orientations and placements at each length and is needle-shaped at a small scale. CT was produced in recent years to address the inherent limits of conventional multiscale representations, which describe curve-like edges with a limited number of coefficients compared to wavelets far from optimal [46]. The CT technique captures the curved edge of characters in an Arabic script. The CT is mathematically described as Equations (13) and (14):

$$\text{CT} = \int_R^0 f(x) W_{a,b,\theta}^-(x) dx \quad (13)$$

$$W_{a,b,\theta}(x) = W_a(R_\theta(x - b)) \quad (14)$$

where $W_{a,b,\theta}(x)$ are the wavelet coefficients; a is the number of levels in the wavelet pyramid ($a = 4$); $b = [3 \ 4 \ 4 \ 5]$ represents location scalar parameters; θ is an orientation parameter $\theta \in [0, 2\pi]$; $R_\theta = \begin{pmatrix} \cos \theta & \sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$ and is the rotation matrix with angle θ .

3.3. Feature Fusion

The precision of signature classification can be improved by extracting appropriate features. A method for fusing hybrid features is proposed to solve the restriction of a single feature extraction technique, as shown in Figure 8. Feature fusion combines several feature vectors to generate the final feature vector, which involves complementing each other's advantages to obtain more robust and accurate outcomes [47]. The ESCF technique converts the feature matrix into a feature vector that describes the signature and can reduce

error rates. ESCF is simple to implement, does not cause the loss of information, and has no impact on computational efficiency.

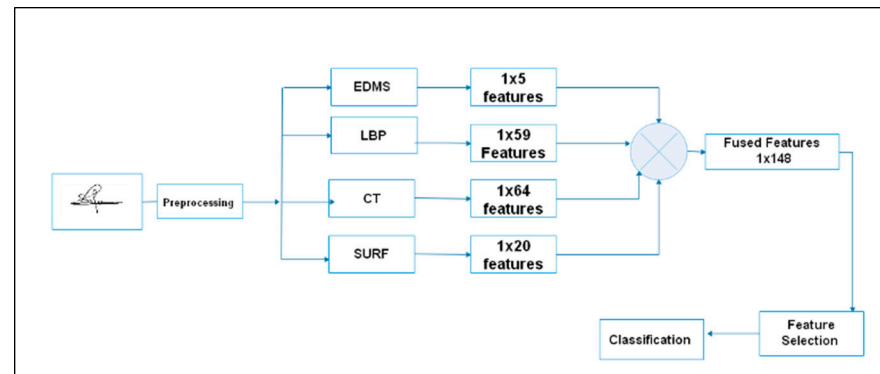


Figure 8. A strategy of feature fusion.

Let A and B be two feature spaces specified on pattern space Ω . For an arbitrary sample $\xi \in \Omega$, the associated feature vectors are $\alpha \in A$ with n -dimensional features and $\beta \in B$ with m -dimensional features; the Serial Fused feature of ξ is defined as $\gamma = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ with dimension $(m + n)$. The mathematical description of the fusing process is based on Equation (15).

$$F(v) = \begin{bmatrix} CT_{1 \times N} \\ EDM_{1 \times N} \\ SURF_{1 \times N} \\ LBP_{1 \times N} \end{bmatrix} \quad (15)$$

where $F(v)$ is the final fused vector of $1 \times \text{sum}(N)$ dimensions for all samples.

3.4. Feature Selection

Feature selection has been a productive area of research in intelligent algorithms and machine learning, which is undoubtedly essential. Feature selection eliminates attributes that may negatively affect the performance of classifiers, such as irrelevant, redundant, or less informative features. As indicated in the preceding section, a simple concatenated fusion technique combines various statistical features to generate an additional dimension that can identify skilled forgeries and genuine signatures with high accuracy.

The problem with combining characteristics without considering correlation and discrimination is that the resulting feature vector cannot detect a skilled forgery. Furthermore, fused features from multiple approaches may provide high-dimension features that could influence the verification process. As a result, a feature selection approach is necessary to minimize the number of features and remove data correlations.

GA has achieved success in many applications. GA can handle more complicated problems than neural networks and specializes in identifying an appropriate feature for a given class. However, automating the design of such fitness functions is still an open challenge. Adopting simple and effective fitness functions is a critical issue for GA.

In this work, GA is used with one class support vector machine (OC-SVM) classifier to discover the genes with the highest predictive performance. Meanwhile, OC-SVM is employed for the classification. This proposal is one of the valuable contributions to reducing the issue of complexity and extending search spaces. Figure 9 shows the flowchart of the GA-OCSVM.

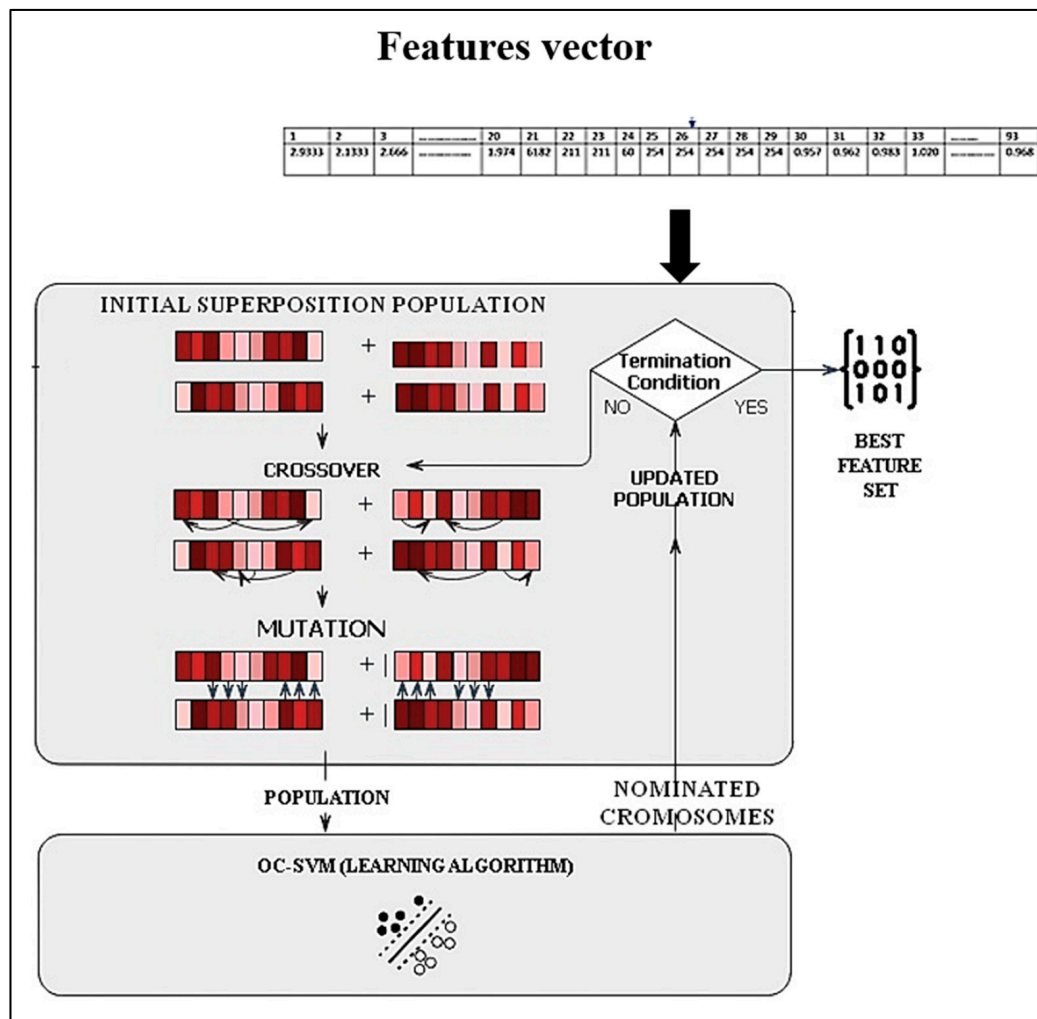


Figure 9. Flowchart of feature selection.

The procedure starts by randomly generating an initial population.

- The initial population size is created and set to 10.
- Calculate and assign a score of the fitness value to each member of the current population. These values are regarded as the raw fitness scores. The fitness function of each individual is determined by evaluating the OC-SVM using a training set. As a result, the fitness function containing classification precision is utilized in this study, as described in Equation (16).

$$fitness(f) = Max(accuracy(f)) \quad (16)$$

where $accuracy(f)$ is the accuracy of the classifier for the subset selection of features expressed by f .

- Select members, known as parents, according to their expectations. Some individuals in the present population with maximum fitness levels are selected as elite (the subset with the best classification precision). These elite members are transmitted to the following population.
- Generates offspring from the selected parents. Offspring are produced by combining the vector entries of two parents (crossover). A uniform crossover with a crossover rate of 0.8 is employed.
- Low-frequency offspring introduce variety into a single-parent population (mutation). A uniform mutation method is selected with a mutation rate of 0.2.

The roulette wheel method is applied to randomly cross and mutate the Chromosome, which keeps the selective pressure in the center rather than at the extremes. In roulette wheel selection, the circular wheel is divided into n pies, where n equals the number of individuals. Based on their fitness level, each individual receives an appropriate circular piece. The wheel is rotated at a defined circumferential point. The region of the wheel just forward of the setpoint is referred to as the parent. The same method is followed for the next generations. The probability P_i of the individual is defined as Equation (17).

$$P_i = \frac{f_i}{S} \quad (17)$$

where $S = \sum_{i=1}^n f_i$, n is the size of the population, and f_i is the fitness of individual i .

- Individuals with higher fitness are more likely to be selected for reproduction.

The 110 features are selected as optimal features, a collection of discriminant characteristics. These distinguishing characteristics are fed into the classifier for verification.

3.5. One-Class Classification

One-class classification (OCC) is used to solve the issue of an imbalanced database of signatures in the real world, where the authentic signature is only generated, and the forged signature is absent, which indicates that the original signer is incapable of forging the signature. This research employs OC-SVM to address these issues. The OC-SVM can successfully deal with the positive samples in the training set. The appropriate distance used by the radial basis function kernel (RBF kernel) must be specified to train the OC-SVM. OC-SVM is developed in two phases. First, one-class information (normal class) trains a classifier to distinguish genuine instances. The classifier rejects the samples belonging to unknown classes and classifies them as forgeries, as shown in Figure 10. A hypersphere with the shortest radius is constructed around the positive class data, which encloses approximately every point in the dataset. According to the parameter RBF γ , the hypersphere is defined by Equation (18).

$$K(x_i, x) = e^{-\gamma(d(x, x_i))} \quad (18)$$

where $d(x, x_i)$ is the distance between the original images x and the target samples x_i (or positive), and $\gamma = 0.07$ is the deviation parameter of the kernel function. The OC-SVM decision function is shown in Equation (19).

$$f(x) = \text{sign}(\sum_{i=1}^N \alpha_i K(x_i, x) - \rho), \quad (19)$$

$$0 \leq \alpha_i \leq \frac{1}{v}$$

where N is the number of training instances, ρ is the distance of the hypersphere from the origin, α_i denotes the Lagrange multiplier for each distance, and $v = 0.01$ represents the trade-off between maximizing the data points (encompassed by the hypersphere) and reducing the hypersphere's distance from the origin. If the decision value of the sample is more significant than zero, we conclude that the target is a positive class; otherwise, it is a negative class.

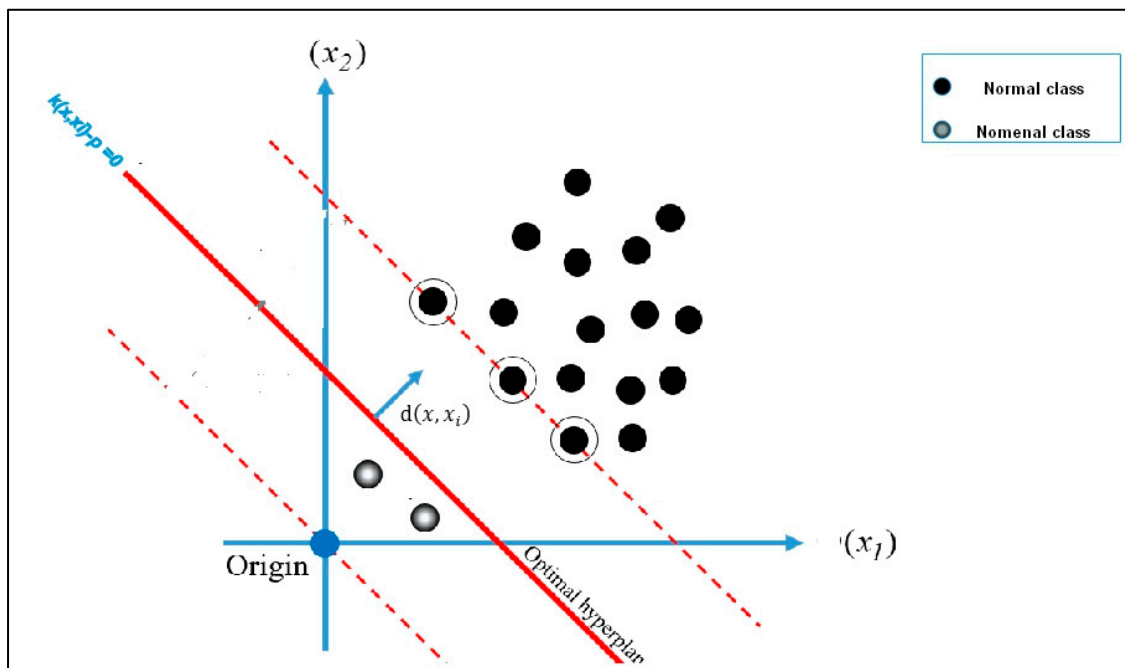


Figure 10. One-class support vector machine.

4. Experimental Results and Analysis

The suggested model was evaluated on three databases: SID Arabic signatures, CEDAR, and UTSig [48–50]. The model was constructed in MATLAB®2021a on an Intel® Core™ i5-8300H CPU @2.30 GHz.

4.1. Experiments and Evaluation of Preprocessing

The proposed preprocessing strategies' performance is appraised using image quality performance metrics such as mean square error (MSE) and peak signal-to-noise ratio (PSNR) in Equations (20) and (21). The assessment aims to illustrate the efficiency of image enhancement procedures on captured signature images.

The PSNR ratio is a quality measurement between the original image I and the enhanced image J . The high value of PSNR demonstrates the significant quality of the processed image. However, as mentioned above, more than this ratio is needed in the offline system because critical data may be lost during preprocessing. As a result, MSE is used to verify the image quality without losing key features. If the value of MSE is close to zero, the image quality is accepted. Conversely, the image loses its main attributes.

$$\text{PSNR}(I, J) = 10 \log_{10} R^2 / \text{MSE}(I, J) \quad (20)$$

$$\text{MSE}(I, J) = \frac{1}{N * M} \sum_{a=1}^N \sum_{b=1}^m (I(a, b) - J(a, b))^2 \quad (21)$$

$M * N$ signifies the image size, and $I(a, b)$ and $J(a, b)$ denote the original and processed image pixel intensities, respectively. R is the maximum allowable pixel value. Six experiments were performed to show the impact and significance of each preprocessing method on the image, as shown in Tables 2–4.

Table 2. Results of preprocessing on the SID database.

Binarization	Filtering	Segmentation	Isolation	Thinning	Skeletonization	PSNR	MSE
x	✓	✓	✓	✓	✓	0.66473	55,796.55
✓	x	✓	✓	✓	✓	48.22255	0.9791
✓	✓	x	✓	✓	✓	59.33641	0.07576
✓	✓	✓	x	✓	✓	48.22479	0.97859
✓	✓	✓	✓	x	✓	48.21457	0.9809
✓	✓	✓	✓	✓	x	48.21593	0.98059
✓	✓	✓	✓	✓	✓	64.85921	0.02124

Table 3. Results of preprocessing on the UTSIG database.

Binarization	Filtering	Segmentation	Isolation	Thinning	Skeletonization	PSNR	MSE
x	✓	✓	✓	✓	✓	0.29984	60,687.15
✓	x	✓	✓	✓	✓	48.29015	0.96397
✓	✓	x	✓	✓	✓	58.16179	0.10013
✓	✓	✓	x	✓	✓	48.29242	0.96347
✓	✓	✓	✓	x	✓	48.27211	0.96799
✓	✓	✓	✓	✓	x	48.27471	0.96741
✓	✓	✓	✓	✓	✓	62.94906	0.03297

Table 4. Results of preprocessing on the CEDAR database.

Binarization	Filtering	Segmentation	Isolation	Thinning	Skeletonization	PSNR	MSE
x	✓	✓	✓	✓	✓	0.12439	63,189.01
✓	x	✓	✓	✓	✓	48.14727	0.99622
✓	✓	x	✓	✓	✓	60.87482	0.05316
✓	✓	✓	x	✓	✓	48.16807	0.99146
✓	✓	✓	✓	x	✓	48.16467	0.99223
✓	✓	✓	✓	✓	x	48.1664	0.99184
✓	✓	✓	✓	✓	✓	72.40394	0.00374

Based on the MSE and PSNR results. The findings demonstrate that after applying the proposed preprocessing, the output images maintain the exact representation of the original images but with fewer pixels. It can be evidenced that the proposed preprocessing methods achieve an equilibrium between noise reduction and image data preservation.

4.2. Experiments and Evaluation of Verification

In this step, the performance of the proposed model is comprehensively evaluated. The verification is evaluated in two experiments: (a) features selection and extracted features without combining preprocessing; and (b) integration preprocessing with feature extraction and selection. FAR, FRR, and EER measurements were used to assess the model. FAR and FRR are two types of error measurements used to evaluate the performance of biometric systems. FRR is the percentage of authorized users whose request is incorrectly rejected. FAR is the percentage of unauthorized users who are mistakenly accepted, and EER specifies the point where the FRR and FAR are equal and stated as flowing equations.

$$FAR = \frac{\text{no. of forged signatures classified as genuine}}{\text{Total no : forged signatures}} \times 100 \quad (22)$$

$$FRR = \frac{\text{no. of genuine signatures classified as forged}}{\text{Total no : genuin signatures}} \times 100 \quad (23)$$

$$EER = \frac{FRR + FAR}{2} \quad (24)$$

The proposed model was only trained on genuine samples with no forgeries to simulate the verification system in the real world, as shown in Table 5. The ratio validation of the model is 0.2. with random selected.

1. In the first experiment, the model performance was examined without preprocessing steps. This experiment aims to show the impact of preprocessing on verification accuracy. As shown in Table 6, the unsatisfactory verification results confirm that each preprocessing step significantly enhances image quality; this is what the second experiment proved.
2. The second experiment included all stages of the proposed model, including preprocessing, hybrid feature extraction, feature fusion, feature selection, and verification. The training phase was separately performed using three sets of genuine (G) samples. Table 7 displays the verification results on the SID Arabic database.

Table 5. Division of database samples for modeling assessment.

Database	Phase	Genuine	Skilled Forgery	Simple Forgery
UTSIG	Training	Set1 575 (5×115)	0	0
		Set2 1150 (7×115)		
		Set3 1380 (10×100)		
	Testing	2530 (22×115)	690 (6×115)	4140 (20×115)
		2300 (20×115)		
		1955 (17×115)		
SID	Training	Set1 700 (7×100)	0	0
		Set2 1000 (10×100)		
		Set3 1200 (12×100)		
	Testing	3300 (33×100)	2000 (20×100)	2000 (20×100)
		3000 (30×100)		
		2800 (28×100)		
CEDAR	Training	Set1 275 (5×55)	0	
		Set2 385 (7×55)		
		Set3 550 (10×55)		
	Testing	1045 (19×55)	1320 (24×55)	
		935 (17×55)		
		770 (14×55)		

Table 6. Results of verification without preprocessing.

Database	FAR_Simple	FAR_Skilled	FRR	ERR
SID	0.077	0.065	0.190	0.130
UTSig	0.042	0.229	0.235	0.185
CEDAR		0.048	0.309	0.309

Table 7. Results of the proposed model on the SID database.

Training Signature Sample	FRR	FAR_Skilled	FAR_Simple	EER	STD *	Acc. (%)	Time (Sec.)
7G	0.032	0.061	0.072	0.049	0.017	95.099	86.8799
10G	0.026	0.041	0.070	0.041	0.014	95.929	82.2922
12G	0.037	0.039	0.063	0.044	0.007	95.583	85.0556

* STD represents the standard division.

The suggested approach was also evaluated on CEDAR (English signature) and UTSIG (Farsi signature) to demonstrate its comprehensive performance. The model attained superb results on these databases, as shown in Tables 8 and 9.

Table 8. Results of the proposed model on the CEDAR database.

Training Signature Sample	FRR	FAR	ERR	STD	Acc. (%)	Time (Sec.)
5G	0.065	0.058	0.061	0.002	93.859	46.647
7G	0.067	0.034	0.051	0.008	94.926	48.5451
10G	0.056	0.041	0.048	0.004	95.179	52.3336

Table 9. Results of the proposed model on the UTSIG database.

Training Signature Sample	FRR	FAR_Skilled	FAR_Simple	ERR	STD	Acc. (%)	Time (Sec.)
5G	0.045	0.186	0.028	0.076	0.03	92%	97.6121
7G	0.055	0.178	0.024	0.078	0.02	92%	100.738
10G	0.052	0.162	0.030	0.074	0.02	93%	127.86

To compare the proposed algorithm results with existing signature verification techniques, Tables 10–12 show the results of the proposed model with state-of-the-art methods. From the results, it is clear that the performance of the presented model is good in terms of FAR and FRR using one-class training. The average error rate has been reduced to 10% in the SID database, 2–10% in the UTSIG database, and 3–7% in the CEDAR database.

Table 10. Comparison of the results of the proposed approach and the state-of-the-art methods using the SID database.

References	Feature Type	Classifier	FAR-Simple	FAR-Skilled	FRR	EER
[48]	Geometric features + wavelet Transformation	(MLP)	0.0379	0.0895	0.1495	0.0984
[51]	Graphometric + geometric	NN	0.0745	0.1870	-	-
		HMM	0.295	0.3705		
		SVM	0.1385	0.1875		
[52]	Global feature Proposed	PMCM-BP/SVM	0.0835	0.0910	-	-
			0.063	0.039		

Table 11. Comparison of the results of the proposed approach and the state-of-the-art methods using the UTSIG database.

References	Features Type	Classifier	FAR-Skilled	FRR	EER
[53]	ResNet CNN pretrained on Handwriting classification tasks	SVM	-	-	0.0980
[49]	Geometric features (fixed-point arithmetic)	SVM	0.1841	0.4170	0.2933
[54]	DWT + Gabor filter	CNN	0.1365	0.3267	0.223
[55]	HOG+ DRT	Discriminative Deep Metric Learning (DDML)	0.1615	0.1896	0.1745
[56]	Gaussian Weighting Based Tangent Angle (GWBTA) + Cylindrical Shape Context	SVM	0.2495	0.0741	0.1618
[57]	Statistical + shape based + Similarity based + Frequency based	Binary Red Deer Algorithm (BRDA) feature selection + Naïve Bayes classifier	-	-	0.100
	Proposed		0.162	0.052	0.074

Table 12. Comparison of the results of the proposed approach and the state-of-the-art methods using the CEDAR database.

References	Features Type	Classifier	FAR	FRR	EER
[58]	Histogram of oriented gradients (HOG)	SVM	-	-	0.2092
	LPB				0.0890
	LDF		0.0654	0.0581	0.0618
[59]	Chain code histogram	SVM	0.0784	0.0939	0.086
[60]	DWT + local quantized patterns (LQP)	SVM	0.0746	0.0786	0.0766
[61]	Local + Global	SVM	0.0743	0.0446	0.0595
[34]	Geometric feature	Threshold	0.0654	-	0.0766
[62]	DWT + multi-resolution box-counting (MRBC) pretrained	Gaussian process (GP)	0.0757	0.0643	0.07
[5]	DCNN(GoogLeNet) + NCA features selection	SVM	-	-	0.200
[63]	SNN	Threshold	0.0734	0.0694	0.0714
[64]	Interval type-2 fuzzy set (IT2FS)	ELM (extreme learning machine) + SRC (sparse representation classifier)	0.1054	0.1236	0.111
[65]	AlexNet	Decision Tree (DT)	-	-	0.079
	Proposed		0.041	0.056	0.048

4.3. Discussion

The proposed approach obtained superior FAR, FRR, and ERR on the three databases, particularly for skilled forgeries, which is the essential contribution of this study. Each stage of the model contributed to the increased precision. The preprocessing steps enhanced the verification results because all uninformative data and noise were removed. Moreover, the verification system's supremacy is due to fused hybrid features and discrimination feature selection. The proposed feature extraction is advantageous because it combines multiple features to address the low intraclass difference between skilled forgery and genuine signatures, and the high intraclass difference between original signatures to the same writer. This combination maximizes the merit of each approach by complementing the advantages of other techniques, hence improving verification capabilities.

EDMs is a global textural descriptor used to analyze the entire image. Although EDMs were adequate for simple forgeries, they could not achieve high precision for skilled forgeries. For skilled forgers, LBP was more effective and accurate than EDMs. LBP is a local texture descriptor that describes a small part of the image and extracts more information. Some images may have local details that LBP could not figure out. In order to increase the detection percentage of skilled forgery, the SURF descriptor was used to add more distinct local features. SURF can detect and describe the interesting feature of the image. The key points in the picture include characteristics such as corners, edges, spots, and so on. The consistency of the key points can be helpful for performance. SURF outperforms SIFT in terms of performance and computational complexity.

Furthermore, the curvy lines in the Arabic characters were captured using a CT; CT accurately represents curved discontinuities. In addition, the feature selection strategy plays a crucial role in improving accuracy by removing insignificant characteristics. It also tackled the problem of correlation that may result from the feature fusion process, as illustrated in Figure 11.

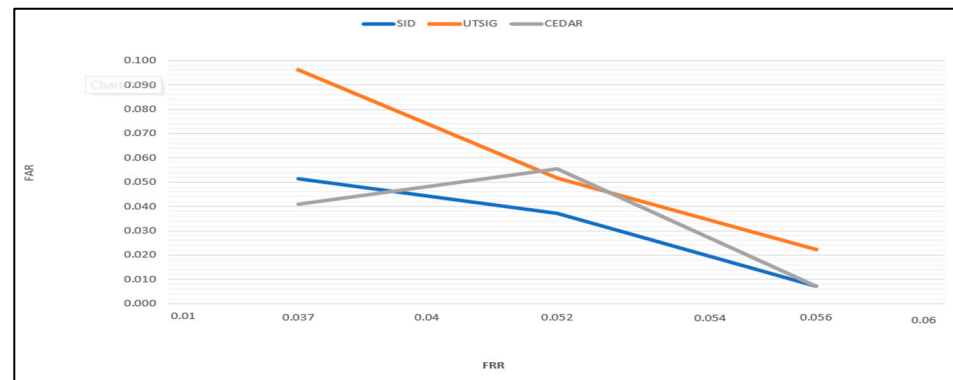


Figure 11. The accuracy in the plot curve.

Overall, the findings of the proposed methodology prove that the proposed method performs considerably better than recent signature verification approaches.

The complexity of the proposed model was assessed in this work using computing time. Each signature was processed in 0.01063 s, 0.01982 s, and 0.01544 s for the SID, CEDAR, and UTSIG databases.

5. Summary of the Scientific Work

The Arabic OSV system was presented using six stages: preprocessing, hybrid feature extraction and fusion, GA-OCSVM-based optimum feature selection, and OCC. This research suggests a multi-step process for preprocessing images, starting with image binarization, and moving on to denoising, segmenting, isolating, thinning the signature, and skeletonizing. Experiments yielded efficient results with high PSNR and low MSE. The suggested approaches significantly impacted verification processing time and accuracy.

Though the proposed method elicits and fuses four different statistical techniques, the ESCF fusion strategy has remained feasible regarding complexity. The best features were then selected using GA.

Additionally, OCC was employed to address the need for forgery signature samples in practical applications in the real world. The proposed model was implemented using the three databases mentioned earlier.

6. Conclusions

This paper proposed a signature verification model with four primary phases: preprocessing and hybrid feature extraction, followed by feature fusion. Finally, features selection and verification. The algorithm's output was constructed with genuine and forged sig-

nature samples from three standard databases: SID-Signatures, CEDAR, and UTSIG. The suggested approaches significantly impacted the verification processing time and accuracy.

The proposed method combined four different statistical techniques. The best GA-based features were then identified for classification. Additionally, the proposed model employed OC-SVM to address the restriction of the current Arabic OVS regarding skilled forgery. The results revealed that the proposed system outperformed existing techniques. It improved the FAR by 10% on the SID-Arabic signature database without increasing the computation time. The experiment yielded 0.037 FRR, 0.039 FAR_skilled, 0.063 FAR simple, and 0.044 EER.

Moreover, the model was superior in enhancing the EER values of UTSig and CEDAR databases, which achieved 0.074 and 0.048, respectively. The FRR value could be enhanced by adding structural features in the future. Additionally, the accuracy of feature selection can be strengthened by improving crossover and mutation.

Author Contributions: A.A.A., M.F.N. and S.M.D.; methodology, A.A.A.; software, A.A.A.; validation, A.A.A., M.F.N. and Z.A.A.A.; formal analysis, A.A.A., M.F.N., S.M.D. and Z.A.A.A.; investigation, A.A.A. and M.F.N.; resources, A.A.A.; writing—original draft preparation, A.A.A., M.F.N. and S.M.D.; writing—review and editing, M.F.N., S.M.D. and Z.A.A.A.; visualization, Z.A.A.A., M.F.N. and S.M.D.; supervision, A.A.A., M.F.N., S.M.D. and Z.A.A.A.; project administration, M.F.N.; funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Universiti Kebangsaan Malaysia (UKM), grant number GUP-2020-065.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the CAIT research group at the Faculty of Information Science and Technology, University Kebangsaan Malaysia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Taherzadeh, G.; Karimi, R.; Ghobadi, A.; Beh, H.M. Evaluation of Online Signature Verification Features. In Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT2011), Gangwon-Do, Republic of Korea, 13–16 February 2011; pp. 772–777.
2. Ahmed, K.; El-Henawy, I.M.; Rashad, M.Z.; Nomir, O. On-Line Signature Verification Based on PCA Feature Reduction and Statistical Analysis. In Proceedings of the 2010 International Conference on Computer Engineering & Systems, Cairo, Egypt, 30 November–2 December 2010; pp. 3–8.
3. Pal, S.; Pal, U.; Blumenstein, M. Off-Line English and Chinese Signature Identification Using Foreground and Background Features. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, 10–15 June 2012; pp. 1–7.
4. Diaz, M.; Ferrer, M.A.; Impedovo, D.; Malik, M.I.; Pirlo, G.; Plamondon, R. A Perspective Analysis of Handwritten Signature Technology. *ACM Comput. Surv.* **2019**, *51*, 1–39. [\[CrossRef\]](#)
5. Poddar, J.; Parikh, V.; Bharti, S.K. Offline Signature Recognition and Forgery Detection Using Deep Learning. *Procedia Comput. Sci.* **2020**, *170*, 610–617. [\[CrossRef\]](#)
6. Al-Omari, Y.M.; Abdullah, S.N.H.S.; Omar, K. State-of-the-Art in Offline Signature Verification System. In Proceedings of the 2011 International Conference on Pattern Analysis and Intelligence Robotics, Kuala Lumpur, Malaysia, 28–29 June 2011; Volume 1, pp. 59–64.
7. Haddad, A.; Riadi, L. Online Automatic Arabic Handwritten Signature and Manuscript Recognition. In Proceedings of the 2017 International Conference on Engineering & MIS (ICEMIS), Monastir, Tunisia, 8–10 May 2017; pp. 1–7.
8. Wilson-Nunn, D.; Lyons, T.; Papavasiliou, A.; Ni, H. A Path Signature Approach to Online Arabic Handwriting Recognition. In Proceedings of the 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), London, UK, 12–14 March 2018; pp. 135–139.
9. Majid, A.; Khan, M.A.; Yasmin, M.; Rehman, A.; Yousafzai, A.; Tariq, U. Classification of Stomach Infections: A Paradigm of Convolutional Neural Network along with Classical Features Fusion and Selection. *Microsc. Res. Tech.* **2020**, *83*, 562–576. [\[CrossRef\]](#) [\[PubMed\]](#)

10. Smejkal, V.; Kodl, J. Dynamic Biometric Signature—An Effective Alternative for Electronic Authentication. *Adv. Technol. Innov.* **2018**, *3*, 166–178.
11. Levy, A.; Nassi, B.; Elovici, Y.; Shmueli, E. Handwritten Signature Verification Using Wrist-Worn Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 1–26. [\[CrossRef\]](#)
12. Rehman, A.; Naz, S.; Razzak, M.I. Writer Identification Using Machine Learning Approaches: A Comprehensive Review. *Multimed. Tools Appl.* **2019**, *78*, 10889–10931. [\[CrossRef\]](#)
13. Morocho, D.; Morales, A.; Fierrez, J.; Vera-Rodriguez, R. Towards Human-Assisted Signature Recognition: Improving Biometric Systems through Attribute-Based Recognition. In Proceedings of the 2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA), Sendai, Japan, 29 February–2 March 2016; pp. 1–6.
14. Stauffer, M.; Maergner, P.; Fischer, A.; Ingold, R.; Riesen, K. Offline Signature Verification Using Structural Dynamic Time Warping. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 1117–1124. [\[CrossRef\]](#)
15. Basu, A.; Shioya, H.; Park, C. *Statistical Inference: The Minimum Distance Approach*; CRC: Boca Raton, FL, USA, 2019; ISBN 1420099663.
16. Kaur, H.; Kansa, E.R. Distance Based Online Signature Verification with Enhanced Security. *Int. J. Eng. Dev. Res.* **2017**, *5*, 1703–1710.
17. Al-juboori, S.S. Signature Verification Based on Moments Technique. *Ibn AL-Haitham J. Pure Appl. Sci.* **2013**, *26*, 385–395.
18. Suthaharan, S. Support Vector Machine. In *Machine Learning Models and Algorithms for Big Data Classification*; Springer: Boston, MA, USA, 2016; pp. 207–235.
19. Kumar, A.; Bhatia, K. *Offline Handwritten Signature Verification Using Decision Tree*; Springer: Singapore, 2023; pp. 305–313.
20. Chu, J.; Zhang, W.; Zheng, Y. Signature Verification by Multi-Size Assembled-Attention with the Backbone of Swin-Transformer. *Neural Process. Lett.* **2023**, preprint. [\[CrossRef\]](#)
21. Kumari, K.; Rana, S. Towards Improving Offline Signature Verification Based Authentication Using Machine Learning Classifiers. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 3393–3401. [\[CrossRef\]](#)
22. Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Analyzing Features Learned for Offline Signature Verification Using Deep CNNs. *arXiv* **2016**, arXiv:1607.04573.
23. Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Writer-Independent Feature Learning for Offline Signature Verification Using Deep Convolutional Neural Networks. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN 2016), Vancouver, BC, Canada, 24–29 July 2016; pp. 2576–2583.
24. Chugh, A.; Jain, C. Kohonen Networks for Offline Signature Verification. *Comput. Sci.* **2017**, *4*, 18–23.
25. Akhundjanov, U.Y.; Starovoitov, V.V. *Static Signature Verification Based on Machine Learning*; Big Data and Advanced Analytics: Minsk, Belarus, 2022; pp. 11–12.
26. Singh, A.; Viriri, S. Online Signature Verification Using Deep Descriptors. In Proceedings of the 2020 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, 11–12 March 2020; pp. 1–6.
27. Stauffer, M.; Maergner, P.; Fischer, A.; Riesen, K. A Survey of State of the Art Methods Employed in the Offline Signature Verification Process. In *New Trends in Business Information Systems and Technology*; Dornberger, R., Ed.; Springer: Berlin/Heidelberg, Germany, 2021; Volume 294, pp. 17–30. ISBN 9783030483319.
28. Stauffer, M.; Maergner, P.; Fischer, A.; Riesen, K. Graph Embedding for Offline Handwritten Signature Verification. In Proceedings of the 2019 3rd International Conference on Biometric Engineering and Applications, Stockholm, Sweden, 29–31 May 2019; pp. 69–76.
29. Zulkarnain, Z.; Mohd Rahim, M.S.; Othman, N.Z.S. Feature Selection Method for Offline Signature Verification. *J. Technol.* **2015**, *75*, 79–84. [\[CrossRef\]](#)
30. Guru, D.S.; Manjunatha, K.S.; Manjunath, S.; Somashekara, M.T. Interval Valued Symbolic Representation of Writer Dependent Features for Online Signature Verification. *Expert Syst. Appl.* **2017**, *80*, 232–243. [\[CrossRef\]](#)
31. Kar, B.; Mukherjee, A.; Dutta, P.K. Stroke Point Warping-Based Reference Selection and Verification of Online Signature. *IEEE Trans. Instrum. Meas.* **2017**, *67*, 2–11. [\[CrossRef\]](#)
32. Sekhar, V.C.; Mukherjee, P.; Guru, D.S.; Pulabaigari, V. Online Signature Verification Based on Writer Specific Feature Selection and Fuzzy Similarity Measure. *arXiv* **2019**, arXiv:1905.08574.
33. Souza, V.L.F.; Oliveira, A.L.I.; Cruz, R.M.O.; Sabourin, R. Improving BPSO-Based Feature Selection Applied to Offline WI Handwritten Signature Verification through Overfitting Control. In Proceedings of the GECCO'2020: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, Cancun, Mexico, 8–12 July 2020; pp. 69–70. [\[CrossRef\]](#)
34. Manjunatha, K.S.; Guru, D.S.; Annapurna, H. Interval-Valued Writer-Dependent Global Features for Off-Line Signature Verification. In *Lecture Notes in Computer Science, Proceedings of the Mining Intelligence and 7 Exploration: 5th International Conference, MIKE 2017, Hyderabad, India, 13–15 December 2017*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10682, pp. 133–143; ISBN 9783319719276.
35. Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Meta-Learning for Fast Classifier Adaptation to New Users of Signature Verification Systems. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1735–1745. [\[CrossRef\]](#)
36. Batool, F.E.; Attique, M.; Sharif, M.; Javed, K.; Nazir, M.; Abbasi, A.A.; Iqbal, Z.; Riaz, N. Offline Signature Verification System: A Novel Technique of Fusion of GLCM and Geometric Features Using SVM. *Multimed. Tools Appl.* **2020**, *v*, 1–20. [\[CrossRef\]](#)

37. Ghosh, R. A Recurrent Neural Network Based Deep Learning Model for Offline Signature Verification and Recognition System. *Expert Syst. Appl.* **2021**, *168*, 114249. [[CrossRef](#)]
38. Agrawal, P.; Chaudhary, D.; Madaan, V.; Zabrovskiy, A.; Prodan, R.; Kimovski, D.; Timmerer, C. Automated Bank Cheque Verification Using Image Processing and Deep Learning Methods. *Multimed. Tools Appl.* **2021**, *80*, 5319–5350. [[CrossRef](#)]
39. Sharma, N.; Gupta, S.; Mehta, P.; Cheng, X.; Shankar, A.; Singh, P.; Nayak, S.R. Offline Signature Verification Using Deep Neural Network with Application to Computer Vision. *J. Electron. Imaging* **2022**, *31*, 041210. [[CrossRef](#)]
40. Gandhi, T.; Bhattacharyya, S.; De, S.; Konar, D.; Dey, S. Advanced Machine Vision Paradigms for Medical Image Analysis. In *Hybrid Computational Intelligence for Pattern Analysis and Understanding*; Academic Press: Amsterdam, The Netherlands, 2020.
41. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 4th ed.; Pearson: New York, NY, USA, 2018; ISBN 9780133356724.
42. Martiana, K.E.; Barakbah, A.R.; Akmilis, S.S.; Hermawan, A.A. Auto Cropping on Iris Image for Iridology Using Histogram Analysis. In Proceedings of the 2016 International Conference on Knowledge Creation and Intelligent Computing (KCIC), Manado, Indonesia, 15–17 November 2016; pp. 42–46. [[CrossRef](#)]
43. Sathesh, A.; Babikir Adam, E.E. Hybrid Parallel Image Processing Algorithm for Binary Images with Image Thinning Technique. *J. Artif. Intell. Capsul. Netw.* **2021**, *3*, 243–258. [[CrossRef](#)]
44. Talab, M.A.; Abdullah, S.N.H.S.; Razalan, M.H.A. Edge Direction Matrixes-Based Local Binary Patterns Descriptor for Invariant Pattern Recognition. In Proceedings of the 2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR), Hanoi, Vietnam, 15–18 December 2013; pp. 13–18.
45. Cajote, R.D.; Guevara, R.C.L. Combining Local and Global Features for Offline Handwriting Recognition. *Cajote* **2006**, *26*, 21–32.
46. Starck, J.L.; Candès, E.J.; Donoho, D.L. The Curvelet Transform for Image Denoising. *IEEE Trans. Image Process.* **2002**, *11*, 670–684. [[CrossRef](#)] [[PubMed](#)]
47. Sudha, D.; Ramakrishna, M. Comparative Study of Features Fusion Techniques. In Proceedings of the 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT), Bangalore, India, 16–17 March 2017; pp. 235–239. [[CrossRef](#)]
48. Abdelghani, I.A.B.; Amara, N.E.B. SID Signature Database: A Tunisian Off-Line Handwritten Signature Database. In *Lecture Notes in Computer Science, Proceedings of the New Trends in Image Analysis and Processing, ICIAP 2013 Workshops, Naples, Italy, 11 September 2013*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8158, pp. 131–139; ISBN 9783642411892.
49. Soleimani, A.; Fouladi, K.; Araabi, B.N. UTSig: A Persian Offline Signature Dataset. *IET Biom.* **2017**, *6*, 1–8. [[CrossRef](#)]
50. Singh, S.; Hewitt, M. Cursive Digit and Character Recognition in CEDAR Database. In Proceedings of the 15th International Conference on Pattern Recognition (ICPR-2000), Barcelona, Spain, 3–7 September 2000; Volume 2, pp. 569–572.
51. Abdelghani, I.A.B.; Amara, N.E. Ben Planar Multi-Classifer Modelling-NN/SVM: Application to Off-Line Handwritten Signature Verification. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 369, pp. 87–97; ISBN 9783319197128.
52. Abroug, I.; Amara, N.E. Ben Off-Line Signature Verification Systems: Recent Advances. In Proceedings of the International Image Processing, Applications and Systems Conference, Kuala Lumpur, Malaysia, 19–21 October 2015; pp. 1–6.
53. Mersa, O.; Etaati, F.; Masoudnia, S.; Araabi, B.N. Learning Representations from Persian Handwriting for Offline Signature Verification, a Deep Transfer Learning Approach. In Proceedings of the 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), Tehran, Iran, 6–7 March 2019; pp. 268–273.
54. Jagtap, A.B.; Hegadi, R.S.; Santosh, K.C. Feature Learning for Offline Handwritten Signature Verification Using Convolutional Neural Network. *Int. J. Technol. Hum. Interact.* **2019**, *15*, 54–62. [[CrossRef](#)]
55. Soleimani, A.; Araabi, B.N.; Fouladi, K. Deep Multitask Metric Learning for Offline Signature Verification. *Pattern Recognit. Lett.* **2016**, *80*, 84–90. [[CrossRef](#)]
56. Narwade, P.N.; Sawant, R.R.; Bonde, S.V. Offline Handwritten Signature Verification Using Cylindrical Shape Context. *3D Res.* **2018**, *9*, 48. [[CrossRef](#)]
57. Banerjee, D.; Chatterjee, B.; Bhowal, P.; Bhattacharyya, T.; Malakar, S.; Sarkar, R. A New Wrapper Feature Selection Method for Language-Invariant Offline Signature Verification. *Expert Syst. Appl.* **2021**, *186*, 115756. [[CrossRef](#)]
58. Arab, N.; Nemmour, H.; Chibani, Y. New Local Difference Feature for Off-Line Handwritten Signature Verification. In Proceedings of the 2019 International Conference on Advanced Electrical Engineering (ICAEE), Algiers, Algeria, 19–21 November 2019; pp. 1–5.
59. Bharathi, R.K.; Shekar, B.H. Off-Line Signature Verification Based on Chain Code Histogram and Support Vector Machine. In Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, India, 22–25 August 2013; pp. 2063–2068. [[CrossRef](#)]
60. Bhunia, A.K.; Alaei, A.; Roy, P.P. Signature Verification Approach Using Fusion of Hybrid Texture Features. *Neural Comput. Appl.* **2019**, *31*, 8737–8748. [[CrossRef](#)]
61. Sharif, M.; Khan, M.A.; Faisal, M.; Yasmin, M.; Fernandes, S.L. A Framework for Offline Signature Verification System: Best Features Selection Approach. *Pattern Recognit. Lett.* **2020**, *139*, 50–59. [[CrossRef](#)]
62. Shariatmadari, S.; Emadi, S.; Akbari, Y. Nonlinear Dynamics Tools for Offline Signature Verification Using One-Class Gaussian Process. *Int. J. Pattern Recognit. Artif. Intell.* **2020**, *34*, 2053001. [[CrossRef](#)]
63. Jagtap, A.B.; Sawat, D.D.; Hegadi, R.S.; Hegadi, R.S. Verification of Genuine and Forged Offline Signatures Using Siamese Neural Network (SNN). *Multimed. Tools Appl.* **2020**, *79*, 35109–35123. [[CrossRef](#)]

-
64. Qiu, S.; Fei, F.; Cui, Y. Offline Signature Authentication Algorithm Based on the Fuzzy Set. *Math. Probl. Eng.* **2021**, 2021, 5554341. [[CrossRef](#)]
 65. Wencheng, C.; Xiaopeng, G.; Hong, S.; Limin, Z. Offline Chinese Signature Verification Based on AlexNet. In Proceedings of the Advanced Hybrid Information Processing, First International Conference, ADHIP 2017, Harbin, China, 17–18 July 2017; Volume 219, pp. 33–37.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.