

Enhanced Offline Signature Verification using GLCM and Correlation-based Feature Selection Technique

S D Bhavania¹ and R K Bharathib²

^{1,2}Department of Computer Applications, JSS Science and Technology University, Mysuru, Karnataka, India
Email: bhavani.sd@jssstuniv.in, rkbharathi@jssstuniv.in

Abstract—In security sensitive applications, signature verification plays vital role in identity confirmation. The feature extraction and feature selection process is crucial for accomplishing the task of authentication. In this paper offline signature verification system is proposed, which mainly concentrates on extracting proper features from signature images. Then irrelevant and redundant features are removed using a feature selection method. Gray-Level Co-occurrence Matrix (GLCM) is employed as a texture feature extractor, followed by correlation-based feature selection (CBS) to refine the feature space and enhance classifier performance. The goal is to find the most discriminative features required for distinguishing authentic and forged signatures. The outcome of the proposed method is the optimized set of features that are fed into an ensemble voting classifier comprising a Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF). The proposed approach is tested on four different publicly available benchmark datasets, including CEDAR, BHSig260-Bengali, BHSig260-Hindi, and MCYT. These datasets comprise diverse signature styles and demographic variations, which are required for developing a more generalized model. Results demonstrate the superiority of the proposed methodology, achieving reduced feature dimensionality resulting in low computational complexity without compromising decision consistency. This paper highlights the potential of combining optimized texture-based feature extraction with ensemble learning for efficient and accurate offline signature verification, facilitating practical applications in resource-constrained devices.

Keywords— Signature Verification, GLCM, Correlation-based feature selection, Voting Classifier, cross validation.

I. INTRODUCTION

Biometric is the more secure way of authenticating persons identity. The physiological or behavioural characteristics like face, finger print, signature can be used for authentication purpose. Amongst these traits signature plays a significant role in our day to day life. Since signature is the simplest of all biometric, it can be easily forged. Signature is widely used in bank checks, legal documents, business contracts. Hence, signature verification is a crucial step in biometric authentication method which widely used for validating the identity and ensuring the security in various fields. This process generally involves three primary major steps: preprocessing, feature extraction, and classification. Preprocessing enhances the quality of the signature image by removing noise and normalizing it, feature extraction identifies unique characteristics of the signature, and classification determines the authenticity of the signature by comparing it against stored references. In signature verification systems, feature extraction is essential because it is the basis for discriminating authentic signatures from fakes.

It entails finding and extracting significant features from the signature, that captures the distinctive qualities of an individual's signature, such as stroke directions and angles, loop and intersection counts, stroke dynamics, connected components and so on. Efficient feature extraction is essential for optimizing the overall performance of the system. Feature selection followed by feature extraction, reduces the feature dimension leading to decreased computational complexity. Feature selection process retains most relevant feature required for verification process, and removes the feature that do not contribute for discriminating genuine from forgery signatures. [5,8,11,12,28]

In this paper, the Gray-Level Co-occurrence Matrix (GLCM) is used to acquire texture-based information that accurately describes the structural patterns of signatures. The correlation coefficient-based feature selection is used to reduce the feature vector dimension and remove redundant or non-contributing features, guaranteeing that only the most discriminating features are kept while reducing duplication. This reduces the computational complexity and improves model generalization. Additionally, to increase robustness and decision reliability, a voting classifier, which integrates the predictions of three classifiers such Support Vector Machine (SVM), Random Forest (RF), and k-Nearest Neighbors (KNN) is used. The proposed method improves the accuracy of verifying authentic and fake signatures by combining GLCM features with correlation-based feature selection and a voting classifier.

This is the arrangement of the remaining portions of the paper: section 2 delineates the literature study, while Section 3 provides a full account of the suggested technique. The datasets used for model evaluation are described in section 4, and section 5 presents the results and their interpretation. Result analysis is given in section 6 and Section 7 concludes.

II. REVIEW OF LITERATURE

The role of feature extraction is crucial in transforming the raw signature image data into a format that is beneficial for comparison and analysis. It involves extracting relevant characteristics or patterns from the image that are distinctive and can be used to differentiate between different classes. It helps in representing the data in a more compact and informative way, making it easier for machine learning algorithms to process. There are several feature extraction methods used in SigVer systems. Some of the prominent ones are Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), Speeded-UP Robust Features (SURF), Local Binary Patterns (LBP)[3,4,6,14,15,17,23,25,27]. In this work, HOG feature descriptor has been exploited to extract features from signature images. The following section explores some of the methods proposed related to feature extraction.

Simplified Histogram of Oriented Gradients (SHOG), statistical characteristics of Gray Level Co-occurrence Matrix (SGLCM), and gradient-based local binary pattern (GLBP) features were combined in work [16]. And then fused feature vector was used to train random forest classifier to develop writer-independent offline SigVer system. In paper [30], authors have utilized a complementary set of features namely HOG, LBP, and SIFT to train two types of SVM classifiers. It was demonstrated that combining classifiers leads to better results, with score level fusion outperforming feature level fusion. In paper [13] authors utilized pyramid histogram of oriented gradient (PHOG) as a feature to train modified K-nearest neighbour classifier. In paper [24], authors used quad tree Histogram of Templates having 20 templates to extract features. For classification, a classic Artificial Immune System was modified by replacing KNN with SVM, which resulted in improved performance of the verification system.

In [1], authors proposed a method to extract HOG features and the extracted were passed to Decision Tree to select relevant features. The selected features are fed to CNN for verification. In [33], authors attempted to extract HOG features by setting the block size to 22x22, and the cell size to 128 with 9 bins histogram per cell. From each signature sample, 216 features were extracted and passed to SVM for classification. In paper [2], authors attempted to extract the HOG feature having a block size of [4x4] pixels.

Hence, 34,596 feature vector length is utilized to describe each signature image, and these features are passed to LSTM for classification.

In [15], authors attempted to develop a reliable verification system using both online and offline signatures acquired simultaneously. Static features including geometric features, GLCM, and HOG were recovered from offline signatures, whereas dynamic features like radius of curvature, velocity, acceleration, and angle were retrieved from online signatures. The static features were passed to SVM and dynamic features were passed to DTW for classification.

Choosing a subset of pertinent features from the initial collection of features is known as feature selection. By concentrating primarily on the most informative features, it aids in enhancing model performance, decreasing

overfitting, and accelerating training. Commonly utilized approaches for feature selection include filter methods (e.g., correlation-based feature selection), wrapper methods (e.g., recursive feature removal), and embedding methods (e.g., LASSO regression) [18,19,26,29,31].

By projecting the data onto a lower-dimensional space, dimensionality reduction entails lowering the number of features while maintaining the most crucial information. In addition to lowering computing complexity and addressing the curse of dimensionality, it can aid in the visualization of high-dimensional data. The commonly used for dimensionality reduction techniques are Principal Component Analysis (PCA)[4,21], Singular Value Decomposition (SVD) and linear discriminant analysis (LDA). These methods ensure us to look upon the need to eliminate the non-performing or ill-performing features.

It is observed that large feature dimension leads to high computation complexity. In real-time applications, it is preferred to have a system that involves less computation complexity. Regarding this, an attempt is made to develop a model whose computation complexity is as minimal as possible by selecting relevant features using a filter-based feature selection method.

III. METHODOLOGY

The proposed methodology for signature verification encompasses several key stages: preprocessing, feature extraction, and classification. The proposed model is illustrated in in Figure 1. During preprocessing, the signature images are denoised and normalized to ensure uniformity. Feature extraction is performed using the Gray-Level Co-occurrence Matrix (GLCM), which captures texture-based features essential for distinguishing signatures. To reduce dimensionality and retain the most relevant features, correlation-based feature selection is employed. Finally, classification is carried out using a voting classifier that combines SVM, Random Forest, and KNN, with cross-validation ensuring robust performance evaluation. Each of these phases are detailed in the following sections.

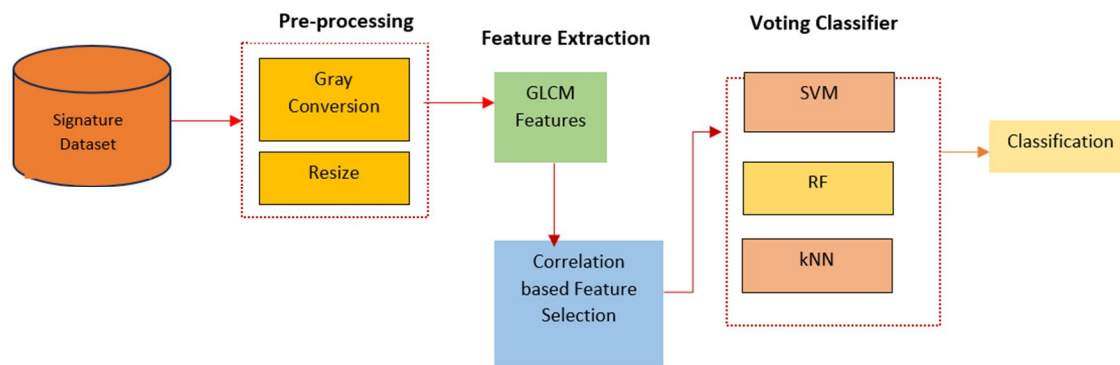


Fig. 1 Proposed Methodology

A. Pre-processing

At first signature images are converted to grayscale, in order to reduce computational cost, enhance focus on structural information, and simplify the task.

B. Feature extraction using GLCM

The GLCM is a texture analysis technique used in image processing and computer vision. GLCM is a square matrix that measures the frequency with which pairs of pixel intensity values (grey levels) occur at a given spatial relationship inside an image. The GLCM is created by evaluating pairs of pixels that have a specified spatial relationship. GLCM is computed based on the co-occurrence of pixel intensity values at a specified distance and direction in an image. Given an image, the GLCM is constructed by counting the number of times pairs of pixel values occur at a certain relative position. To compute GLCM two parameters are required: (1) Distance and Direction - The co-occurrence matrix is often computed for multiple distances and angles, capturing different aspects of texture. (2) Grey Levels - The number of grey levels used in the image affects the size of the GLCM. Once the grey-level co-occurrence matrices are computed various texture features can be derived from the GLCM [3].

In this study, the grey level co-occurrence matrices are calculated from the pre-processed signature images along four different angles viz. $\theta=0^\circ, 45^\circ, 90^\circ$, and 135° by keeping the distance equal to 1. With the help of grey-level

co-occurrence matrices, we have computed five statistical features: Contrast, Dissimilarity, Homogeneity, Energy, and Correlation.

Contrast: Contrast is a measure of local intensity variations within an image. It describes how varied the pixel brightness is for nearby pixels at a certain spatial relationship.

The formula for computing contrast (C) from the GLCM is often given as:

$$\text{Contrast} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i, j) \cdot (i - j)^2 \quad (1)$$

Where, $P(i, j)$ denotes the value in the GLCM matrix at row i and column j , N is the number of grey levels.

A high contrast value implies significant intensity variations between surrounding pixels, whereas a low contrast value shows that neighboring pixels have identical intensity values

Energy: is a metric of similarity or angular second momentum (ASM). Energy assesses the consistency or smoothness of the image's texture. It is calculated as the sum of the squared components in the GLCM and is frequently normalized. The formula for calculating energy (E) is provided by:

$$\text{Energy} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i, j)^2 \quad (2)$$

High energy levels demonstrate a more consistent or smoother texture in the image whereas low energy levels imply a more complex and varied texture. Energy is frequently normalized to ensure that its values fall inside a certain range (for example, between 0 and 1). Normalization is usually accomplished by dividing the energy by the total number of components in the GLCM.

Homogeneity: is also called inverse difference moment (IDM). It describes the proximity of the distribution of elements in the GLCM to the diagonal, demonstrating how evenly the intensity values in the picture are spread. The formula for computing homogeneity (H) is often given by:

$$\text{Homogeneity} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i, j) / 1 + |i - j| \quad (3)$$

High homogeneity values imply that the intensity levels in the image are close to each other, resulting in a more evenly distributed texture. Low homogeneity scores indicate a more diverse and heterogeneous texture. Homogeneity is frequently normalized to ensure that its values fall inside a predetermined range (for example, between 0 and 1).

Correlation: It characterizes the linear dependency between the pixel intensities at different locations in an image. The correlation feature is calculated using the formula:

$$\text{Correlation} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - \mu_x)(j - \mu_y) P(i, j) / \sigma_x \sigma_y \quad (4)$$

μ_x and μ_y are the means of the pixel intensities in the x and y directions, respectively

σ_x and σ_y are the standard deviations of the pixel intensities in the x and y directions, respectively.

Positive correlation values exhibit a positive linear connection between pixel intensities at various places, implying that the intensities tend to fluctuate concurrently. Negative correlation values exhibit a negative linear connection, implying that the intensities fluctuate inversely. The correlation value ranges between -1 and 1, where -1 indicates perfect negative correlation, 1 indicates perfect positive correlation, and 0 indicates no correlation.

Dissimilarity: It measures the average absolute difference in intensity between pairs of pixels at different locations in an image. The dissimilarity feature is calculated using the formula:

$$\text{Dissimilarity} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |i - j| \cdot P(i, j) \quad (5)$$

Dissimilarity is related to the contrast feature, but it focuses on absolute differences in intensity rather than the squared differences utilized in contrast calculations. High dissimilarity values indicate a larger average variation in intensity between adjacent pixel pairs, implying a more varied texture. Low dissimilarity values indicate a smaller average difference in intensity, implying a more uniform texture.

C. Correlation-based Feature selection (CFS)

Feature extraction process extracts all possible features from images, but all these feature may not contribute for the verification process. The extracted feature vector may contain redundant and irrelevant features. Hence, before proceeding towards verification process, it is necessary to take care of redundant and irrelevant features. Feature selection methods eliminate irrelevant or redundant features, thereby enhancing accuracy of verification process, also reduces overfitting, and decreases computational cost. Feature selection is an important step in datasets with a large number of features, where many of the features do not contribute for the prediction

capability of the model. In this paper, Correlation coefficient feature selection (CFS) method used to evaluate the relationship among the features, which helps to retain the features that contribute more for model prediction. The principle of CFS is that good feature subsets should contain features that: 1) have strong correlation with the target variable meaning they offer meaningful information for predicting the label. 2) Have little inter-correlation with each other meaning they provide distinct information. This confirms that the features that are selected are both informative and non-redundant. While selecting features, it is crucial to inspect for multicollinearity meaning high correlation between features, because highly correlated features could hinder the performance of several models. Hence, multicollinearity has to be prevented by eliminating one feature from pairs of features that have a strong correlation with one another. Since signature images are non-linear data, we chose to use Spearman's Rank correlation coefficient for selecting the most relevant features. Spearman's correlation coefficient (ρ) is calculated as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (6)$$

where d_i is the difference between the ranks of corresponding variables and n is the number of data points. The value of ρ can range between -1 and 1.

D. Voting Classifier

Once the informative and non-redundant features are selected, the next step is to classify the signature sample into genuine or forged. In this paper, voting classifier (VC) an ensemble of machine learning technique that aggregates predictions from several classifiers to produce a final prediction is used for verification. The VC usually performs better than any one model in the ensemble, when several models are combined. The models such as k-NN or decision trees are prone to overfitting, in such cases majority vote reduces the impact of error of single model. It has two modes of operations: Hard Voting: Selects the class label that has the majority vote from all models. Soft Voting: Computes the average of predicted probabilities and chooses the class with the highest probability. Here we used Soft Voting, as it performed well compared to Hard Voting. In soft voting, the final prediction is determined by computing the average probability for each class and selecting the one with the highest value. The proposed method uses three popular classifiers: Support Vector Machine (SVM), Random Forest (RF) and K Nearest Neighbour (KNN).

E. Cross Validation

To evaluate the performance of machine learning models, fixed size train-test split is a commonly used technique but it has certain drawbacks and constraints such as chances of bias in the split, overfitting or underfitting, high variance in outcomes, and so on. The limitations of a fixed train-test split, are solved by the cross-validation (CV) technique, which works by dividing the dataset into several subsets. Then some of these subsets are used for training the model and remaining subsets are used for testing. In this investigation, Stratified k-fold Cross-Validation is used, which is an improved version of k-fold cross-validation that is especially useful when working with imbalanced datasets, where samples in one class are much more frequent than another. In this method, the entire dataset is split into k equally sized subsets (folds). In each iteration, one-fold is used as the test set, and the remaining k-1 folds are used as the training set. Stratification guarantees that each fold has approximately the same percentage of samples from each class as the original dataset.

IV. DATASET DESCRIPTION

The proposed model is evaluated on four different datasets: CEDAR, MCYT-75, BHSig260-Bengali and BHSig260- Hindi.

CEDAR: This is developed and made accessible to everyone for research purposes, generated by the University of Buffalo, State University of New York's CEDAR research centre. In an 8-bit greyscale area measuring 2 by 2 inches, writers were instructed to sign. The signed documents were then scanned at 300 dpi and saved in Portable Network Graphic (PNG) format. All 2640 signatures from various signers are included in this collection. Every signer signs a total of twenty-four authentic and skilled forgery signatures.

MCYT-75: The dataset comprises scanned images of signatures, from 75 different individuals. Each individual typically provides 15 genuine signature samples as well as a set of 15 skilled forged signatures. The signatures are scanned at 600 dpi and saved in Bit Map (BMP) format.

BHSig260-Bengali: This dataset comprises signatures from 100 different individuals written in Bengali language. Each individual typically provides 24 genuine samples as well as a set of 30 skilled forged signatures. The signatures are scanned at 300 dpi and saved in Tagged Image File (TIF) format.

BHSig260-Hindi: This dataset comprises signatures from 160 different individuals written in Hindi language. Each individual typically provides 24 genuine samples as well as a set of 30 skilled forged signatures. The signatures are scanned at 600 dpi and saved in Bit Map (BMP) format.

V. EXPERIMENTATION AND RESULTS

This section presents the verification results of the proposed model. This section includes two parts, first part explains how features are extracted and relevant features are selected. While the second section summarizes the classifier's accuracy, and performance, and compares it with other existing state-of-the-art models.

A. Experimental setup

The study utilized a dataset of signature images, preprocessed to ensure uniform size and grayscale conversion for consistent texture analysis. Then Gray-Level Co-occurrence Matrices (GLCM) were computed for a fixed pixel distance of 1 and four angular orientations: 0°, 45°, 90°, and 135°. The five texture features contrast, correlation, energy, dissimilarity, and homogeneity, were extracted from each GLCM, yielding 6 features per angle. For all four angular orientations, a total of 20 features (5 features $\times$ 4 angles) were computed for each image.

The extracted GLCM features are passed into a correlation coefficient feature selection algorithm in order to lower the feature dimension, which in turn reduces the system complexity. Features are ranked based on the absolute values of their Spearman correlation coefficients. We calculated the Spearman correlation between the features themselves to identify multicollinearity. The Spearman correlation coefficient matrix for 5 features out of 20 features is shown in Table 1 for demonstration purposes.

TABLE I: SPEARMAN CORRELATION COEFFICIENT MATRIX FOR BHSig260-BENGALI DATASET

	f_0	f_1	f_2	f_3	f_4
f_0	1.00	0.76	0.85	0.72	0.93
f_1	0.76	1.00	0.73	0.92	0.71
f_2	0.85	0.73	1.00	0.72	0.84
f_3	0.72	0.93	0.72	1.00	0.74
f_4	0.93	0.71	0.84	0.74	1.00

In the correlation coefficient matrix, the value 0 represents no correlation and 1 represents high correlation. From the correlation coefficient matrix, the feature pairs having high inter-feature correlations are considered for removing redundant features. One of the two features from the high inter-feature correlation pair is removed to avoid redundancy. When two features are highly correlated, one can be removed to reduce multicollinearity and simplify the model without losing much information. In table 1 the feature pairs like (f_0 , f_4), (f_1 , f_3) are highly correlated because the coefficient value is more than 0.9. To reduce multicollinearity, f_0 , f_1 can be retained and f_3 , f_4 can be removed.

The efficacy of the model is determined by the accuracy obtained when the model is run on four different datasets viz CEDAR, BHSig260-B, BHSig260-H, and MCYT-75. Both genuine and skilled forgeries present in the dataset are used for training and testing the model. The response of the model for various thresholds (viz 0.7, 0.8, and 0.9) are summarized in the Table 3 for all four datasets. The Table 2 shows the number of features selected per threshold values after removing one of the features from feature-pair having high inter-feature correlation, and also, the accuracies of the voting classifier.

TABLE II: RESULTS OF THE MODEL ACROSS DIFFERENT THRESHOLDS

Dataset	CEDAR		BHSig260-B		BHSig260-H		MCYT_75	
Threshold	FVD	VCA	FVD	VCA	FVD	VCA	FVD	VCA
0.7	5	98.03	5	86.90	3	68.59	5	70.16
0.8	8	98.07	8	87.51	5	73.79	8	72.46
0.9	10	98.18	11	90.16	6	76.70	10	75.23

*FVD: Feature Vector Dimension; VCA : Voting Classifier Accuracy

The experimental results indicate that feature dimensionality plays a crucial role in the model's performance. When a lower correlation threshold is applied, it leads to a smaller feature dimension by eliminating more redundant features. Conversely, increasing the threshold retains more features, resulting in a higher feature dimension. This larger feature set allows the model to capture more nuanced information, leading to improved performance. However, the trade-off is an increase in computational complexity as the feature dimension grows. Fig. 2 illustrates the variations of accuracies and feature dimensions for different thresholds. Overall, the results

suggest that higher feature dimensions, obtained through a higher correlation threshold, enhance the model's ability to discriminate between genuine and forged signatures.

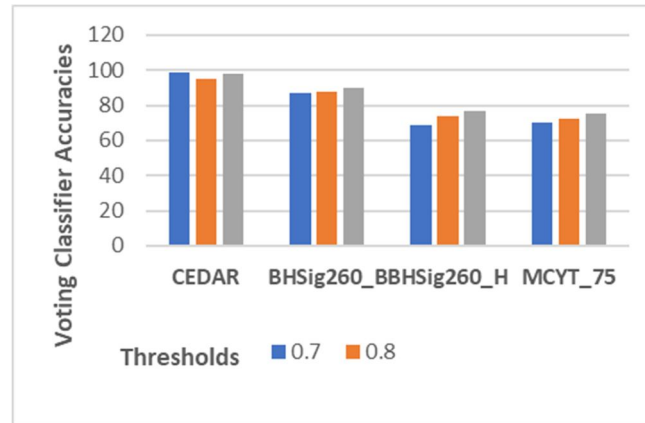


Fig. 2 Shows voting classifier accuracies for different thresholds across various datasets

Among the various experiments conducted with different correlation thresholds, the optimal results were obtained for a threshold value of 0.9 combined with 5-fold cross-validation. This configuration produced the most favourable performance across several key metrics, including Accuracy, False Acceptance Rate (FAR), False Rejection Rate (FRR), and Average Error Rate (AER). These results are summarized in Table 4, which highlights the model's ability to effectively balance the trade-offs between accuracy and error rates, indicating that a higher threshold value for feature selection leads to enhanced discriminative capability and overall model performance. This combination of high threshold and cross-validation effectively captures the model's robustness and generalization ability across unseen data, ensuring reliable signature verification.

The graph illustrating the relationship between feature dimension and model performance is presented in Fig. 3. It demonstrates how varying the number of selected features impacts the accuracy across different datasets. The model achieved its highest accuracy of 98.18% on the CEDAR dataset with 10 features, while for the BHSig260-B dataset, it reached 90.16% accuracy with 11 features. On the BHSig260-H dataset, the model attained 76.7% accuracy using 6 features, for the MCYT-75 dataset, the accuracy was 75.23% with 10 features. These results highlight the correlation between feature dimensionality and model performance, indicating that different datasets benefit from varying feature dimensions, and the optimal number of features varies based on the dataset characteristics.

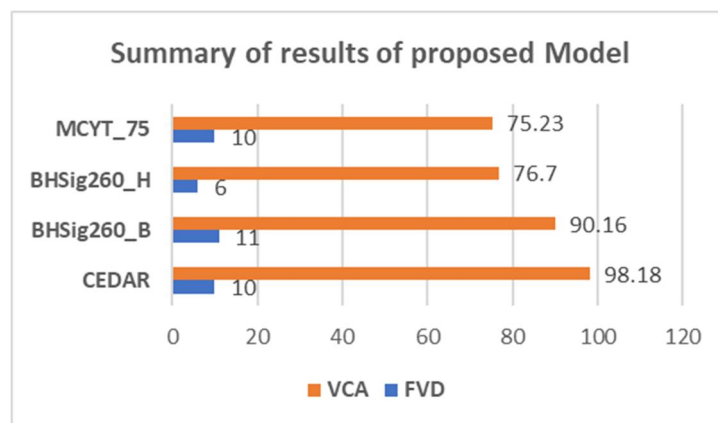


Fig. 3 Illustrates feature vector dimension and the accuracies of all four datasets.

B. Performance comparison with the state-of-the-art methods

We present a comprehensive comparison between our proposed method and several state-of-the-art techniques across four publicly available datasets. The results of this comparison are summarized in Table 3. Since the metrics reported in previous studies vary, to ensure a fair and consistent evaluation, we have focused on two key metrics: Accuracy (Acc.) and Average Error Rate (AER). These metrics provide a clear and direct basis for

assessing the overall performance and error rates of the models, allowing for meaningful comparisons between our approach and existing methods. By standardizing the evaluation criteria, we can more effectively highlight the strengths of our method in terms of both precision and error minimization across diverse datasets.

TABLE III. SHOWS THE COMPARISON PROCESS OF OUR MODEL WITH SOME OF THE STATE-OF-THE-ART OFFLINE SIGVER MODELS.

Reference	METHOD	Results
[1]	CNN-HOG-LSTM	CEDAR: Acc:93.7; AER :13.4, UTSig : Acc:95.4
[2]	HOG-LSTM	CEDAR: Acc:87.7, UTSig:92.4
[20]	CNN	BHSig260-B: Acc:95 BHSig260-H: Acc:90
[17]	Geometrical features -ANN	BHSig260-B: Acc:76.03 BHSig260-H: Acc:83.5 MCYT: Acc:97.33; AER:0.31
[23]	GLCM - GNN	MCYT-75: AER:0.814 UTSig : AER :0.49
[10]	RNN-Structural-Directional features	MCYT: Acc : 91.82
[32]	MA-SCN	BHSig260-B: Acc:92.68; AER:7.90 BHSig260-H: Acc:94.99; AER:5.29
[7]	SURDS	BHSig260-B: Acc:87.34; AER:12.65 BHSig260-H: Acc:89.50; AER:10.49
[9]	Circlet transform-statistical features	MCYT: AER:8.20
[27]	GLCM-ILBP-Geom.feats-SVM	CEDAR: Acc:97.00; AER:3.49 MCYT: Acc:97.85; AER:5.64
[25]	Global-Local feat-GA+SVM	CEDAR: AER: 4.67 MCYT: AER: 5.0
[22]	Graph edit distance model utilizing inkball models	MCYT-75 : AER :9.24 UTSig : AER: 15.80
GLCM_CBS		CEDAR: Acc:98.59; AER:3.25 BHSig260-B: Acc:90.16; AER:7.38 BHSig260-H: Acc:88.5; AER:11.83 MCYT: Acc:76.70; AER:13.31

The articles listed in Table 5 have proposed both neural network based models and conventional machine learning models for offline SigVer. The CNNs, extract a large number of features form signature images , leading to high-dimensional feature space. Processing high dimensional features becomes challenging when the resources are limited. Although CNNs are powerful, they require high computational resources, especially when more number of hidden layers with many filters are used. In paper [1] authors have extracted both HOG, and CNN features , then relevant features were selected using Decision tree algorithm from both HOG and CNN feature vector. After selecting features individually, the selected HOG and CNN features are fused resulting in 23,318 feature dimension. Finally, the fused features were fed to three different classifiers namely SVM, KNN and LSTM. Here, we need to focus on resulting in feature dimension (i.e 23,318) that makes the model computationally heavy. The GLCM features are typically low-dimensional, as they capture specific statistical patterns. GLCM is computationally light, but combining it with GNN introduces high complexity [23] because GNNs involve graph convolutions and dependencies between graph nodes, making the overall model computationally intensive. GNNs also require iterative neighborhood aggregation, which increases the training time. The Geometric features extracted in [17] are low-dimensional and they are effective for shape-based tasks, but they lack the depth or complexity required for more sophisticated tasks. This limits the model's representational power. In paper [10], authors have used RNNs, which usually operate on sequential data, produce time-dependent features. In the case of offline signature data, where temporal data is not the key aspect of the problem, RNNs may generate redundant or irrelevant features, limiting their effectiveness. RNNs require recurrent processing, making them computationally heavy, especially for long sequences. Each step in the sequence depends on the previous step, so training can be slow, and inference times may also increase. The other conventional machine learning models have used handcrafted features like GLCM, LBP, geometric features, circlet transform features used in [9,27], are effective for some tasks but are generally task-specific and do not generalize well across varied datasets.

The primary challenge in machine learning models is to balance the trade-off between computation cost and model performance, especially for high-dimensional data is involved. A larger feature set often leads to increased computational costs because of longer training times and more memory usage. On the other hand, if too many features are removed to reduce feature dimension, the model might lose valuable information, leading

to a decline in performance. This paper makes a sincere effort to explore possible sets of features that can lead to low computational costs without much compromise in performance. This paper includes exploration of feature extraction techniques like GLCM and CBS to find the optimal feature set, by conducting various experiments on different thresholds. This study makes a significant contribution by not only experimenting with GLCM and CBS for feature selection but also systematically exploring thresholds to find the optimal feature set. None of the previous works have focused on this specific threshold-based exploration across different datasets, making it a novel and valuable addition to the field.

VI. CONCLUSION

An offline SigVer system is proposed in this paper, which focuses on exploring an optimised feature set required to discriminate authentic and forged signatures. The impact of feature vector dimension on computational complexity is studied in this paper. The features are extracted using the GLCM method, and the most relevant features are selected using a correlation-based feature selection method. To increase the generalizing capability, the voting classifier is trained using Stratified k-fold cross-validation. The model performs well on the CEDAR and BHSig260-B datasets, but performance declines on BHSig260-H and MCYT-75.

This paper mainly focuses on reducing the feature vector dimension without compromising the performance of the proposed model. The relevant features are wisely chosen by applying the correlation coefficient feature selection method. Where in features are selected by studying the effect of feature vector dimension on the performance of the model. The outcome of the experiments has revealed that feature reduction has not had much impact on CEDAR and BHSig260-B datasets, but performance declines on BHSig260-H and MCYT-75. This motivates us to explore more efficient feature extraction and feature selection methods in our future work.

REFERENCES

- [1] Alsuhimat, F. M., & Mohamad, F. S. (2023a). A Hybrid Method of Feature Extraction for Signatures Verification Using CNN and HOG a Multi-Classification Approach. *IEEE Access*, 11, 21873–21882. <https://doi.org/10.1109/ACCESS.2023.3252022>
- [2] Alsuhimat, F. M., & Mohamad, F. S. (2023b). Offline signature verification using long short-term memory and histogram orientation gradient. *Bulletin of Electrical Engineering and Informatics*, 12(1), 283–292. <https://doi.org/10.11591/eei.v12i1.4024>
- [3] Bakheet, S., & Al-Hamadi, A. (2021). Automatic detection of COVID-19 using pruned GLCM-Based texture features and LDCRF classification. *Computers in Biology and Medicine*, 137, 104781. <https://doi.org/10.1016/j.compbiomed.2021.104781>
- [4] Batool, F. E., Attique, M., Sharif, M., Javed, K., Nazir, M., Abbasi, A. A., Iqbal, Z., & Riaz, N. (2020). Offline signature verification system: A novel technique of fusion of GLCM and geometric features using SVM. *Multimedia Tools and Applications*, 83(5), 14959–14978. <https://doi.org/10.1007/s11042-020-08851-4>
- [5] Bhavani, S. D., & Bharathi, R. K. (2024). A multi-dimensional review on handwritten signature verification: Strengths and gaps. *Multimedia Tools and Applications*, 83(1), 2853–2894. <https://doi.org/10.1007/s11042-023-15357-2>
- [6] Chandra, S., & Maheshkar, S. (2017). Verification of static signature pattern based on random subspace, REP tree and bagging. *Multimedia Tools and Applications*, 76(18), 19139–19171. <https://doi.org/10.1007/s11042-017-4531-2>
- [7] Chattopadhyay, S., Manna, S., Bhattacharya, S., & Pal, U. (2022). SURDS: Self-Supervised Attention-guided Reconstruction and Dual Triplet Loss for Writer Independent Offline Signature Verification. 2022 26th International Conference on Pattern Recognition (ICPR), 1600–1606. <https://doi.org/10.1109/ICPR56361.2022.9956442>
- [8] Diaz, M., Ferrer, M. A., Impedovo, D., Malik, M. I., Pirlo, G., & Plamondon, R. (2019). A Perspective Analysis of Handwritten Signature Technology. *ACM Computing Surveys*, 51(6), 1–39. <https://doi.org/10.1145/3274658>
- [9] Foroozandeh, A., Hemmat, A. A., & Rabbani, H. (2020). Offline Handwritten Signature Verification Based on Circlet Transform and Statistical Features. 2020 International Conference on Machine Vision and Image Processing (MVIP), 1–5. <https://doi.org/10.1109/MVIP49855.2020.9116909>
- [10] Ghosh, R. (2021). A Recurrent Neural Network based deep learning model for offline signature verification and recognition system. *Expert Systems with Applications*, 168, 114249. <https://doi.org/10.1016/j.eswa.2020.114249>
- [11] Hafemann, L. G., Sabourin, R., & Oliveira, L. S. (2017). Offline handwritten signature verification—Literature review. 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), 1–8. <https://doi.org/10.1109/IPTA.2017.8310112>
- [12] Hameed, M. M., Ahmad, R., Kiah, M. L. M., & Murtaza, G. (2021). Machine learning-based offline signature verification systems: A systematic review. *Signal Processing: Image Communication*, 93, 116139. <https://doi.org/10.1016/j.image.2021.116139>
- [13] Harfiya, L. N., Widodo, A. W., & Wihandika, R. C. (2017). Offline signature verification based on pyramid histogram of oriented gradient features. 2017 1st International Conference on Informatics and Computational Sciences (ICICoS), 23–28. <https://doi.org/10.1109/ICICoS.2017.8276332>

- [14] Houtinezhad, M., & Ghaffary, H. R. (2020a). Writer-independent signature verification based on feature extraction fusion. *Multimedia Tools and Applications*, 79(9–10), 6759–6779. <https://doi.org/10.1007/s11042-019-08447-7>
- [15] Zhou, Y., Zheng, J., Hu, H., & Wang, Y. (2021). Handwritten Signature Verification Method Based on Improved Combined Features. *Applied Sciences*, 11(13), 5867. <https://doi.org/10.3390/app11135867>
- [16] Hu, J., Guo, Z., Fan, Z., & Chen, Y. (2017). Offline Signature Verification Using Local Features and Decision Trees. *International Journal of Pattern Recognition and Artificial Intelligence*, 31(03), 1753001. <https://doi.org/10.1142/S0218001417530019>
- [17] Jain, A., Singh, S. K., & Singh, K. P. (2021). Signature verification using geometrical features and artificial neural network classifier. *Neural Computing and Applications*, 33(12), 6999–7010. <https://doi.org/10.1007/s00521-020-05473-7>
- [18] Khaire, U. M., & Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. *Journal of King Saud University - Computer and Information Sciences*, 34(4), 1060–1073. <https://doi.org/10.1016/j.jksuci.2019.06.012>
- [19] Kou, G., Yang, P., Peng, Y., Xiao, F., Chen, Y., & Alsaadi, F. E. (2020). Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. *Applied Soft Computing*, 86, 105836. <https://doi.org/10.1016/j.asoc.2019.105836>
- [20] Longjam, T., Kisku, D. R., & Gupta, P. (2023). Multi-scripted Writer Independent Off-line Signature Verification using Convolutional Neural Network. *Multimedia Tools and Applications*, 82(4), 5839–5856. <https://doi.org/10.1007/s11042-022-13392-z>
- [21] Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). *Computers & Geosciences*, 19(3), 303–342. [https://doi.org/10.1016/0098-3004\(93\)90090-R](https://doi.org/10.1016/0098-3004(93)90090-R)
- [22] Maergner, P., Howe, N. R., Riesen, K., Ingold, R., & Fischer, A. (2019). Graph-Based Offline Signature Verification (Version 1). *arXiv*. <https://doi.org/10.48550/ARXIV.1906.10401>
- [23] Roy, S., Sarkar, D., Malakar, S., & Sarkar, R. (2023). Offline signature verification system: A graph neural network based approach. *Journal of Ambient Intelligence and Humanized Computing*, 14(7), 8219–8229. <https://doi.org/10.1007/s12652-021-03592-0>
- [24] Serdouk, Y., Nemmour, H., & Chibani, Y. (2016). New off-line Handwritten Signature Verification method based on Artificial Immune Recognition System. *Expert Systems with Applications*, 51, 186–194. <https://doi.org/10.1016/j.eswa.2016.01.001>
- [25] Sharif, M., Khan, M. A., Faisal, M., Yasmin, M., & Fernandes, S. L. (2020). A framework for offline signature verification system: Best features selection approach. *Pattern Recognition Letters*, 139, 50–59. <https://doi.org/10.1016/j.patrec.2018.01.021>
- [26] Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. Fco. (2020). A review of unsupervised feature selection methods. *Artificial Intelligence Review*, 53(2), 907–948. <https://doi.org/10.1007/s10462-019-09682-y>
- [27] Wang, Y., Zheng, J., & Zhou, Y. (2022). An Efficient Offline Signature Verification Method Based on Improved Feature Extraction. *2022 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, 609–612. <https://doi.org/10.1109/CEI57409.2022.9950165>
- [28] Wang, Z., Muhammad, M., Yadikar, N., Aysa, A., & Ubul, K. (2023). Advances in Offline Handwritten Signature Recognition Research: A Review. *IEEE Access*, 11, 120222–120236. <https://doi.org/10.1109/ACCESS.2023.3326471>
- [29] Wu, Z., Dong, Y., Wei, H.-L., & Tian, W. (2020). Consistency measure based simultaneous feature selection and instance purification for multimedia traffic classification. *Computer Networks*, 173, 107190. <https://doi.org/10.1016/j.comnet.2020.107190>
- [30] Yilmaz, M. B., & Yanıkoğlu, B. (2016). Score level fusion of classifiers in off-line signature verification. *Information Fusion*, 32, 109–119. <https://doi.org/10.1016/j.inffus.2016.02.003>
- [31] Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *Journal of Applied Science and Technology Trends*, 1(1), 56–70. <https://doi.org/10.38094/jastt1224>
- [32] Zhang, X., Wu, Z., Xie, L., Li, Y., Li, F., & Zhang, J. (2022). Multi-Path Siamese Convolution Network for Offline Handwritten Signature Verification. *2022 The 8th International Conference on Computing and Data Engineering*, 51–58. <https://doi.org/10.1145/3512850.3512854>
- [33] ABBAS, N. H., YASEN, K. N., ALI FARAJ, K. H., RAZAK, L. F., & ALLAH, F. L. (2018). Offline handwritten signature recognition using histogram orientation gradient and support vector machine. *Journal of Theoretical and Applied Information Technology*, 96(8), 2075–2084.