International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(15s) 2026 pp. 1070-1081
https://doi.org/10.70917/ijcisim-2026-4503
Article

Automatic Classification of Clean and Noisy Offline Handwritten Signatures Using Hybrid Machine Learning

Thejashwini B L[1*], H S Nagendraswamy[1], Rajashekara M[1], Lavanya T S[1], Kavitha S[1]

[1]Department of Studies in Computer Science, University of Mysore, Mysuru, Karnataka, India.
Email: thejudotsignuom@gmail.com

Abstract: The automatic partitioning of handwritten signatures into clean and noisy is crucial to reliable authentication in documents. A new approach is presented here that integrates traditional handcrafted texture features such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and Gray-Level Co-occurrence Matrix (GLCM) with deep learning models. We have performed experiments on an offline signature dataset DocSign V2 comprising 2,400 grayscale signature images (50 users, equal number of clean and noisy signatures) created for this work. Eight conventional classifiers are tested on individual and combinations of features. The performance of a CNN built for the task and also four transfer learning models - EfficientNetB0, ResNet50, MobileNetV2, and DenseNet121 - are compared. The experiments show the best result of 97.5% was achieved with the HOG-LBP+AdaBoost combination. This is better than the best of deep learning models, MobileNetV2, which reached 95%. This demonstrates that, particularly for small datasets, carefully designed handcrafted features can be used to effectively classify clean and noisy signatures automatically. Moreover, the proposed system can also be used as a post-processing module to enhance signatures for other analysis or verification.
Keywords—Authentication; Clean signatures; Noisy signatures; Office signatures; Handcrafted features; Deep learning.

1. Introduction

Offline handwritten signatures are one of the most popular and trusted biometric traits for human identification and verification tasks in practical scenarios. But in real document processing, handwritten signatures are often accompanied by extraneous information that, for instance, includes official seals, stamps or handwritten dates and notes. Such additional information introduces clutter (noise) that affects the effectiveness of automatic handwritten signature recognition [1], [2]. To tackle this issue, this work aims to automatically classify offline handwritten signatures into clean and noisy signatures. Such classification is a crucial preliminary step to improve the performance of subsequent processes like noise removal, signature reconstruction, and signature verification [3]. For our purposes, clean signatures only consist of handwriting on a uniform background, without any external noise. noisy signatures are accompanied by other elements, such as stamps, seals, overlapping text or notations. Although existing methods have been proposed to segment signatures from document images [4], a certain level of noise or artifacts could remain. Consequently, there is a need to have a classification process that will be successful in categorising clean and noisy signatures and offer a more dependable signature analysis system.

To this end, the study discusses application of hand-made textual elements in differentiating clean and noisy signatures. Specifically, it will rely on Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and Gray-Level Co-occurrence Matrix (GLCM) due to their capability to capture the structure and texture of signatures [5], [6]. These are looked at individually and also in combination and their performance is measured by training a number of traditional machine learning classifiers.

In addition, the study explores the deep learning models, including a developed Convolutional Neural Network (CNN) and other transfer learning models, including EfficientNetB0, ResNet50, MobileNetV2, and DenseNet121. Adam optimizer is used to train the models and accuracy, precision, recall, F1-score and confusion matrix are used to test the models. This discussion can provide a clearer insight into the effectiveness of classic machine learning algorithms and deep learning models in terms of classifying clean and noisy signatures.

Although numerous studies have been conducted within the field of handwritten signature verification and classification, most of the studies primarily involve binary classification tasks to detect authentic and forged signatures and exploit clean and well-preprocessed datasets like GPDS, MCYT and BHSig260 [1] [2] [3]. Nevertheless, these data sets are not enough to simulate real-life scenarios when signatures can be overlaid with other artifacts like stamps, seals, handwritten notes and other text. These aspects introduce noise that impairs the performance of automatic systems. This has meant that not many researchers have yet concentrated on the job of applying machine learning to label signatures as either clean or noisy as a preprocessing stage to improve the performance and efficiency of actual systems in verifying signature [4].

The key contributions of this work are as follows:

1. Creation of Data: A new dataset (DocSign V2) is created, which comprises of paired clean and noisy samples of signatures to practically simulate act-ual document conditions.

2. Feature Engineering Framework: Detailed examination of handcrafted texture-based features, such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and Gray-Level Co-occurrence Matrix (GLCM) and combinations thereof is conducted to capture structural and textural differences effectively.

3. Hybrid Machine-Deep Learning Framework: A hybrid model that combines traditional machine learning models with deep learning structures is suggested in order to be able to classify with robustness.

4. Preprocessing Improvement: Addition of a special classification step which enhances the performance of the further process denoising, reconstruction, and signature verification.

5. Performance Optimization: As experimental evidence shows, feature based models that are handcrafted perform better on small data sets compared to deep learning based methods and obtain an accuracy of up to 97.5 percent.

2. RELATED WORKS

In recent studies involving signature recognition, the independent writer and language signature recognition method were introduced (Anamika Jain et al., 2021) [1]. The approach includes the fine-tuned GoogLeNet based on the GPDS transfer learning. The approach trained on BHSig260 and MCYT-75 achieved a precision of 96.5, 95.7, and 93 respectively. The results were enhanced by using deep features and SVM as the classifier. We also performed better with a larger training set. Overall, the strategy demonstrates a good scalability and ability to balance both authentic and fake signatures across languages.

Ibraheem M. Alharbi et al. (2023) [2] This article explains an offline method of signature recognition. The system uses fourier Descriptors (FD) features, Histogram of Oriented Gradients (HOG) features, Local Binary Patterns (LBP) features, and morphological features to improve pattern matching. It was tested with MCYT database and local database, which had 100 signatures to train and 60 signatures to test. Optimal results were obtained with K-Nearest Neighbors (KKN) classification with K=1 with a False Acceptance Rate (FAR) of 0.169. The study shows that it is necessary to have Arabic signature datasets across various regions to have a better recognition rate.

Jonathan X. Wang et al. (2019) [3] The work presents DeepSign, a simple Siamese CNN that is based on SqueezeNet. Its design is efficient in terms of offline and online signature verification with a minimal number of parameters. The authors tested their model on the CEDAR and BHSig260 signatures databases, where the model worked well. It had 0.85 accuracy, 0.76 precision, 0.84 recall, and 0.93 AUROC. DeepSign has performed well compared to SigNet and MobileNetv2. These results indicate the high generalization and effectiveness of DeepSign. It can also be used to verify mobile signatures.

Parashuram Bannigidad et al. (2018) [4] This research introduces a collection of more than 1,200 historical documents containing Kannada handwritten Kannada documents from the Hoysala, Vijayanagara and Mysore Wodeyar dynasties. The subjects took the images at high resolution. They segmented the document images into text blocks and extracted features using the LBP method. And they classified them with LDA, K-NN and SVM classifiers. The results were that SVM had the highest accuracy (99.3%). This was better than LDA at 94.6% and K-NN at 98.3%. This indicates that SVM is highly efficient in identifying documents by dynasties.

Hassiba Nemmour et al. (2016) [5] In this paper, the authors investigate gender classification of handwritten documents using the Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) features. They used the IAM-OnDB data set containing samples of more than 200 writers. The researchers reported good results with Support Vector Machine (SVM) classifiers. The results showed robust gender classification when the data used for training roughly equal amounts of 75 samples for each gender. The results verify the enhanced accuracy from the combination of LBP and HOG features. The approach is suitable for writer authentication.

Luiz G. Hafemann et al. (2016) [6] describes a offline, handwritten signature verification method. The method involves the use of Deep Convolutional Neural Networks (DCNNs) for learning writer-independent features and writer-dependent classification.It used GPDS (GPDS-160, GPDS-300) and Brazilian PUC-PR to test the approach. They achieved good results - an Equal Error Rate (EER) of 15.05% with four signatures in GPDS. They also reported lower Average Error Rates (AER) in PUC-PR. The results show that normalization improved performance. The features they learned were able to distinguish valid signatures from skilled forgeries for both data sets.

Dan Claudiu Cireşan et al. (2011) [7] The authors of this paper propose a system that uses an ensemble of CNNs to identify handwritten characters. It is based on the MNIST and NIST SD 19 datasets. The authors have used an ensemble of classifiers to boost the performance. This approach has led to an error rate of 0.27% with a system of seven networks, with a 30 to 350% increase in accuracy over earlier studies. The report found that the error rate for digits was less than that of letters. This was an indication of efficacy and success.

K. M. bin Abdl et al. (2009) [8] This is a review of handwriting identification where they discuss the importance of collecting data, extracting features and classification. The importance of features for distinguishing between different handwriting samples and consequently enhancing recognition rates is stressed. The findings identify current issues, appreciate the shortcomings of current approaches to identification, and arguments for the need for improved feature design and approaches in the future.

Sargur N. Srihari et al. (2004) [9] This is a study of learning approaches for offline signature verification that consider the information about the writer. The paper uses classifiers like distance-based, Naïve Bayes and SVM. It turns out that distance statistics are good in situations when writer information is not necessary and one class learning. However, SVM is preferred in two-class cases. Generally, the results show that distance methods are relatively good. when only normal samples are available. However, SVM works better with forgery data.

Chen-Ming Hung et al. (2007) [10] explore a technique for text classification that uses the Fuzzy Partition Clustering Method (FPCM) combined with Naive Bayesian Augment Learning. The method performs well in using non-labeled data and avoids the limitation of a large labeled body. Using 4,800 web pages from six topics, this combined method has been found to be quicker and more accurate than other clustering techniques (including fuzzy C-means and ISODATA). The technique increases accuracy and negates the need for extensive labeling. In the future, attention needs to be paid to issues such as removal of outlier points and the influence of the initial choice of cluster points to further improve the accuracy.

H. H. Malik et al. (2011) [11] use ATDC, a technique for automatically purifying the training dataset. It combines the use of high-quality class labels, unsupervised clustering and judgments from multiple sources to detect and clean noisy labels automatically. It was tested on large news data sets and boosted the Micro F1 score from 0.678 to around 0.74 and the Macro F1 score from 0.66 to around 0.75. The study seeks to enhance classification. It aims to provide more evidence through experimentation, and k-means and other clustering algorithms.

Harith Al-Sahaf et al. (2013) [12] proposed a genetic programming (GP) approach for binary image classification that uses a single example for each of the two classes.This approach involves local binary pattern (LBP) features, a support vector machine (SVM) classifier and a multiple objective fitness function to define regions of interest in images. They validating it on six texture and object datasets which have different lighting, scale and poses. They demonstrated that the hybrid GP method is significantly more accurate than five other approaches on four datasets. It was comparable with the best methods in the remaining two datasets. In conclusion, it is good for image classification with small data sets and can be applied in the future for feature extraction.

Bojan Despodov et al. (2024) [13] The research proposes enhancing the accuracy of handwritten character recognition by using Hu Moments for feature extraction and the k-nearest neighbors (KNN) algorithm. This approach overcomes the problem of variability in handwriting. The method was tested with 52 characters of the English alphabet with an average recognition rate of 82.69%. The average recognition rate demonstrates the effectiveness of shape descriptors. Data processing steps were involved. This involved converting to grayscale, applying adaptive thresholding and using Gaussian blurring to enhance features. Overall, the study highlights the need for good techniques in handwriting recognition. It also offers lessons on document digitization and language research.

Amrita Tripathi et al. (2024) [14] worked on diagnosing lung diseases in their early stage by generating the two new deep learning models. The first model (MobileNetV2L2) had good results. It had a training accuracy of 99.53%, validation accuracy of 99.8% and testing accuracy of 95.51%. It was the best model evaluated. The other model, a modified CNN2 model, also performed well. It had an accuracy of 96.79% for training, 91.56% for validation and 99.26% for testing. This was very successful and suggests it can be used successfully.

Khaddouj Taifi et al. (2025) [15] This research enhances breast cancer prediction by optimising deep learning models and using GELU instead of ReLU. Modifications to DenseNet-121, DenseNet-201 and MobileNetV2 boosted accuracy and F1-scores on the MIAS, INbreast and DDSM datasets. DenseNet-201 was the best-performing model at 98% accuracy with perfect precision and recall. Gains are considerable, and slightly increase computational costs.

It is evident from the literature that numerous studies have focused on signature verification and recognition. This primarily involves trying to identify authentic and counterfeit signatures by using different feature extraction techniques and deep learning algorithms. Researchers such as Hafemann et al. [6], Wang et al. [3], and Jain et al. [1] have achieved a high success rate in verification by using CNNs and transfer learning-based approaches. However, these works only focus on the case of verification and do not consider the additional noise present in signatures on documents. The available datasets, such as GPDS, BHSig260 and MCYT, consist of clean and cropped signatures. They do not provide information on the challenges of noisy signatures, which may contain seals, text or dates, or other obtrusions.

The present work proposes a new dataset for clean/noisy signature classification. The dataset is made up of signatures extracted from documents that may contain remnant noise after extraction, like it could be the case in an office environment. This dataset underpins the development of a classification system that can automatically identify clean signatures from those that are noisy, before scanning for noise, reconstructing or recognizing the hand movement. So, this dataset is a big advancement, and it opens a new field on noise detection and classification for offline signature analysis that has not had enough attention in the past.

3. PROPOSED METHODOLOGY

The proposed system aims to classify offline handwritten signatures into two categories: clean and noisy. The process involves four main stages: dataset creation, preprocessing**,** feature extraction, and classification. The overall workflow is shown in Fig. 1.

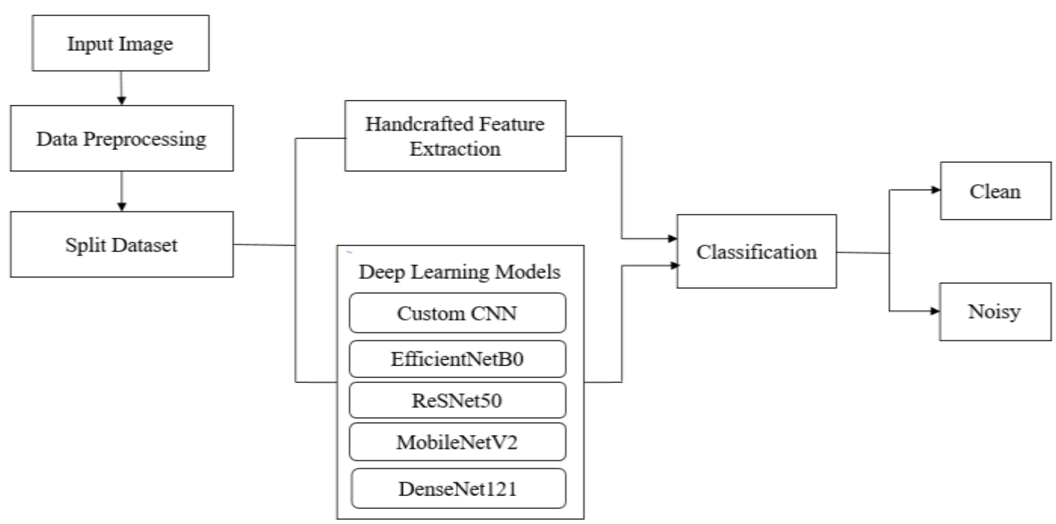

Fig 1. Architecture of proposed work

1. Dataset Creation

In this work, we collected an offline dataset - DocSign v2 - to study classification of signature noise Fig 2 and Fig 3. It contains 2,400 grayscale images of 50 users, each contributing 48 signatures. There are 24 clean and 24 noisy signatures per user. We collected the clean signatures on plain white sheets in a controlled environment to eliminate possible background noise. We then cropped these samples and saved them in user-specific folders. We then generated the noisy signatures by including stamp and seals, date, handwritten comments and additional marks on the same sheets. This mimics real-world situations. Noisy signatures were cropped in the same way as the clean ones, enabling a paired structure. The images were cropped into 256 × 256 pixels and saved in black and white.

This pairing is beneficial for binary classification and enables the potential use of these data in noise removal, reconstruction and recognition..

2. Preprocessing

Prior to training, images were first preprocessed in OpenCV. The images were scaled to 128 x 128 pixels and normalized to [0, 1].We trained and tested the system using 20 samples for training and 4 for testing form each user to achieve a balanced and consistent evaluation. This made the data consistent, and training time shorter.

3. Feature Extraction

Two different approaches were used to extract features - handcrafted features and deep features.

a. Handcrafted Features

Three texture-based descriptors were used to capture the structure and texture of signatures:

- **Histogram of Oriented Gradients (HOG): Stores information about the directions and shapes of strokes.**
- **Local Binary Patterns (LBP): Describes local texture in terms of comparisons between pixel intensities.**
- **Gray-Level Co-occurrence Matrix (GLCM): Evaluates pixel co-occurrences to describe texture features like contrast, energy and correlation.**

To improve representation, combinations such as HOG+LBP, HOG+GLCM, and LBP+GLCM were also tested.

Fig 2.Noisy and Clean signature samples from a single user.

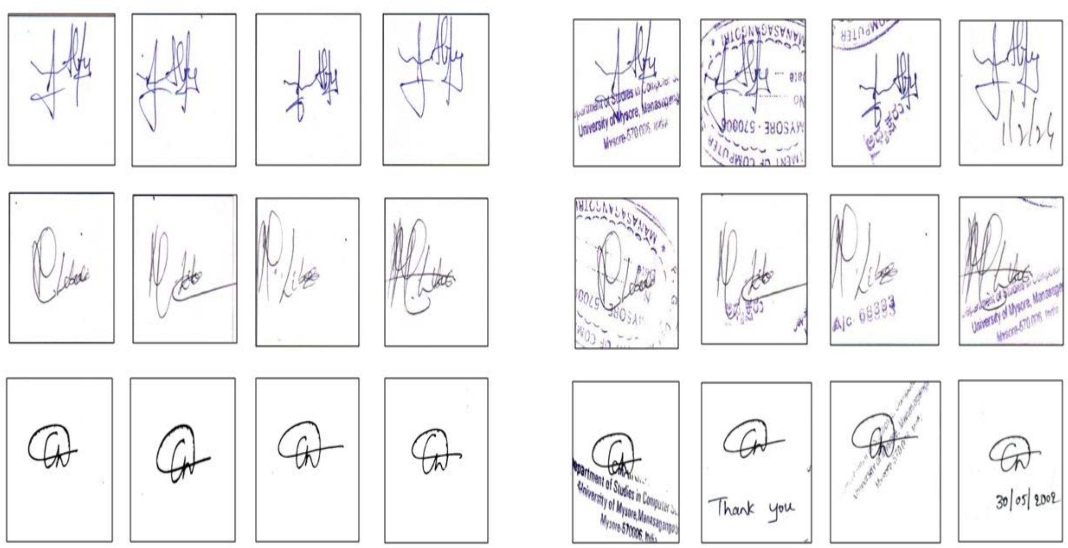

Fig 3. Samples of signature pairs from the dataset show each user's clean signature next to its noisy version. These samples include different handwriting styles and types of noise.

Mathematical Representation of Features

$$\mathrm{LBP}(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c)2^p \qquad (1)$$

Where i_c is the center pixel intensity and i_p represents neighboring pixel values.

$$\mathrm{GLCM}(i,j) = \sum_{x,y} \begin{cases} 1, & \text{if } I(x,y) = i \text{ and } I(x+\Delta x, y+\Delta y) = j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Such formulations are useful to capture local texture patterns and spatial relationship important in differentiating clean and noisy signatures.

b. Deep Learning Features

These formulations assist in capturing local texture patterns and spatial relationship which is essential we automatically learned deep features using a Convolutional Neural Network (CNN) we implemented, and four transfer models: EfficientNetB0, ResNet50, MobileNetV2, and DenseNet121.

All models were trained with the Adam optimizer during 10 epochs. The CNN learnt strokes and edges whereas the transfer models learnt higher-level features.

Machine Learning Classifiers with Evaluation

Examples of traditional machine learning classifiers that we availed ourselves of in this work to classify the clean or dirty handwritten variations in terms of texture-based handcrafted features include SVM, RF, LR, KNN, NB, DT, GB and AB. Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and Gray Level Co-occurrence Matrix (GLCM) were the primary descriptors in the feature extraction step. In addition to these, The various combinations of features that included HOG+LBP, HOG+GLCM and LBP+GLCM were also employed to extract the structural as well as the textural information in the signatures.We chose classifiers on the basis of their learning capacity. Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel was able to deal with non-linearly separable data by projecting the data onto a higher-dimensional space. Logistic Regression (LR) provided a strong linear model and probabilities of results. This feature space nearest neighbour classifier KNN was best suited to small datasets. Multi-level boundaries were formed by the use of decision tree (DT) and random forest (RF) models. RF increased stability through average predictions of a number of trees. Naïve Bayes (NB) provided a simple and fast probabilistic method, which presupposes that all features are independent.Gradient Boosting (GB) and AdaBoost (AB) are based on the same principles of ensemble learning, whereby, a combination of several weak learners is used but with more weight given to the training samples which were misclassified by the weak learner to enhance the performance of the prediction (especially hard cases).

We scaled vectors of features as well as used 20 training and 4 testing samples per user. We evaluated the model on accuracy, precision, recall, macro-averaged F1 score and utilized the confusion matrix to obtain a more insightful understanding of the models. In addition, the majority voting was employed to aid in the consistency of the different sets of features.

As we analyzed the experiments, we found out that the performance was best with the combination of hand-crafted features and the HOG+LBP combination performed the best. AdaBoost performed relatively well in all the models with 97.5% accuracy. Gradient Boosting was 97.25% and SVM 97%. Based on the findings, it is evident that with fewer samples as well as without the adoption of deep learning, reasonably good models to distinguish between good and bad signatures can be developed by designing relevant features. The high performance in the HOG + LBP feature set could be attributed to the fact that it is able to capture the coarse structure of the texture as well as the fine detail. This provides a more detailed and more challenging signatures representation.

Deep Learning Classifiers with Evaluation

In the case of deep learning-based classification, we used a self-trained Convolutional Neural Network (CNN) and four models Applied with transfer learning - EfficientNetB0, ResNet50, MobileNetV2 and DenseNet121. These models do not need the extraction of features to learn the distinctive features of offline signature images. We included a training subset and testing subset in training and verification respectively consisting of 20 and 4 samples respectively. The images were scaled to 128x128 pixels, grayscaled and then scaled[0,1] to be consistent.

The new CNN was comprised of multiple convolutional layers and max-pooling layers. It also had fully connected layers that were using ReLU activation and a softmax activation at the end so as to classify the binary. This CNN produced direction, curvature and edge features that exist in a signature. In the case of the transfer learning models, ResNet50 retained deep feature representations by skip connections and solved the vanishing gradient issue. MobileNetV2 provided effective and lightweight feature extraction, which suits resource-constrained environments. DenseNet121 improves the learning process by linking the layers together to re-use the features and learn fine-grained details. An effective balance between efficiency and performance is achieved in EfficientNetB0 through a balance between depth, width and resolution.

All models were trained with the Adam optimizer. Our learning rate was 0.0001 and batch size was 16. We will use the default number of epochs of 10 to reduce the categorical cross-entropy loss. We measured the model performance in terms of conventional measures like accuracy, precision, recall and F1-score. To analyze in greater detail, we looked at the confusion matrix.

The highest accuracy of 95 percent was with MobileNetV2. This was trailed by the custom CNN at 93.25%. DenseNet121 reached 90.75%. ResNet50 performed fairly well at 74.75%. EfficientNetB0 was the least accurate with 50%. These deep learning models were extremely good at feature learning and yet they were not accurate because of the limited amount of data. Alternatively, classifiers based on machine learning and designed handcrafted features, including AdaBoost with HOG and LBP, were much more successful than the deep learning models with 97.5% accuracy. This shows the utility of the handcrafted features based on texture to smaller datasets and their high discriminative ability in clean and noisy signatures.

4. RESULTS AND DISCUSSION

The results demonstrate that both handcrafted features and deep learning models can accurately classify signatures as clear or noisy (Table 1, 2, 3, 4, 5, 6) along with confusion matrix in Fig 4.The top-performing CNN was MobileNetV2 with an accuracy of 95%. DenseNet121 also performed well with 90.75%, while the CNN was the second best with 93.25%. EfficientNetB0 only achieved 50%, and ResNet50 74.75%. However, the handcrafted features performed best. The major feature was Logistic Regression on HOG, HOG+GLCM up to 97%. The overall best performance of 97.5% was achieved using HOG+LBP features with AdaBoost, showing that handcrafted features were the best performing in the study.

The higher accuracy using hand-inserted features is due to the dataset size. Deep learning models tend to require large amounts of data to learn. With a smaller number of data points, they may underfit or overfit. Descriptors such as HOG, LBP and GLCM encode the stroke edges, local textures and statistical features respectively and are suitable for smaller amounts of data like in our study. The combination of the features improved the information representations and let the classifiers such as Logistic Regression, Gradient Boosting and AdaBoost work. These results suggest that deep learning is powerful but handcrafted feature-based methods are still very robust and in this instance were able to better discriminate between good and bad signatures when compared to deep learning.

Table 1. Result Table for Classification

Feature	Classifiers	Accuracy	Precision	Recall	F1 Score
HOG	**SVM**	0.97	0.9708	0.97	0.97
	Random Forest	0.915	0.9154	0.915	0.915
	Logistic Regression	0.965	0.9654	0.965	0.965
	KNN	0.93	0.9304	0.93	0.93
	Naïve Bayes	0.8975	0.8992	0.8975	0.8974
	Decision Tree	0.8275	0.8277	0.8275	0.8275
	Gradient Boosting	0.9425	0.9428	0.9425	0.9425
	Ada Boost	0.92	0.92	0.92	0.92
LBP	**SVM**	0.8975	0.9028	0.8975	0.8972
	Random Forest	0.9625	0.9628	0.9625	0.9625
	Logistic Regression	0.7175	0.7222	0.7175	0.716

	KNN	0.9375	0.9388	0.9375	0.9375
	Naïve Bayes	0.8975	0.8992	0.8975	0.8974
	Decision Tree	0.93	0.9304	0.93	0.93
	Gradient Boosting	0.9725	0.9728	0.9725	0.9725
	Ada Boost	0.955	0.9557	0.955	0.955
GLCM	**SVM**	0.8125	0.8138	0.8125	0.8123
	Random Forest	0.87	0.8703	0.87	0.87
	Logistic Regression	0.875	0.8764	0.875	0.8749
	KNN	0.8025	0.8047	0.8025	0.8021
	Naïve Bayes	0.86	0.8629	0.86	0.8597
	Decision Tree	0.8125	0.8127	0.8125	0.8125
	Gradient Boosting	0.88	08803	0.88	0.88
	Ada Boost	0.865	0.8668	0.865	0.8648

Table 2. Performance of Classifiers on HOG+LBP Feature Results

Feature	**Classifiers**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
HOG+LBP	**SVM**	0.97	0.9708	0.97	0.97
	Random Forest	0.935	0.9354	0.935	0.935
	Logistic Regression	**0.965**	**0.9654**	**0.965**	**0.965**
	KNN	0.93	0.9304	0.93	0.93
	Naïve Bayes	0.9025	0.9037	0.9025	0.9024
	Decision Tree	0.895	0.8952	0.895	0.895
	Gradient Boosting	**0.9725**	**0.9726**	**0.9725**	**0.9725**
	Ada Boost	**0.975**	**0.9754**	**0.975**	**0.975**

Table 3.Performance of Classifiers on GLCM+LBP Feature Results

Feature	**Classifiers**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
GLCM+LBP	**SVM**	0.81	0.8115	0.81	0.8098
	Random Forest	0.9575	0.9581	0.9575	0.9575
	Logistic Regression	0.9	0.9049	0.9	0.8997
	KNN	0.8075	0.8092	0.8075	0.8072
	Naïve Bayes	0.85	0.8535	0.85	0.8496
	Decision Tree	0.9425	0.943	0.9425	0.9425
	Gradient Boosting	0.9775	0.9778	0.9775	0.9775
	Ada Boost	0.955	0.9566	0.955	0.955

Table 4. Performance of Classifiers on HOG+GLCM Feature Results

Feature	Classifiers	Accuracy	Precision	Recall	F1 Score
HOG+GLCM	**SVM**	0.8125	0.8138	0.8125	0.8123
	Random Forest	0.93	0.9304	0.93	0.93
	Logistic Regression	0.97	0.97	0.97	0.97
	KNN	0.8575	0.8608	0.8575	0.8572
	Naïve Bayes	0.9075	0.9075	0.9075	0.9075
	Decision Tree	0.8825	0.8825	0.8825	0.8825
	Gradient Boosting	0.9675	0.9678	0.9675	0.9675
	Ada Boost	0.9475	0.9476	0.9475	0.9475

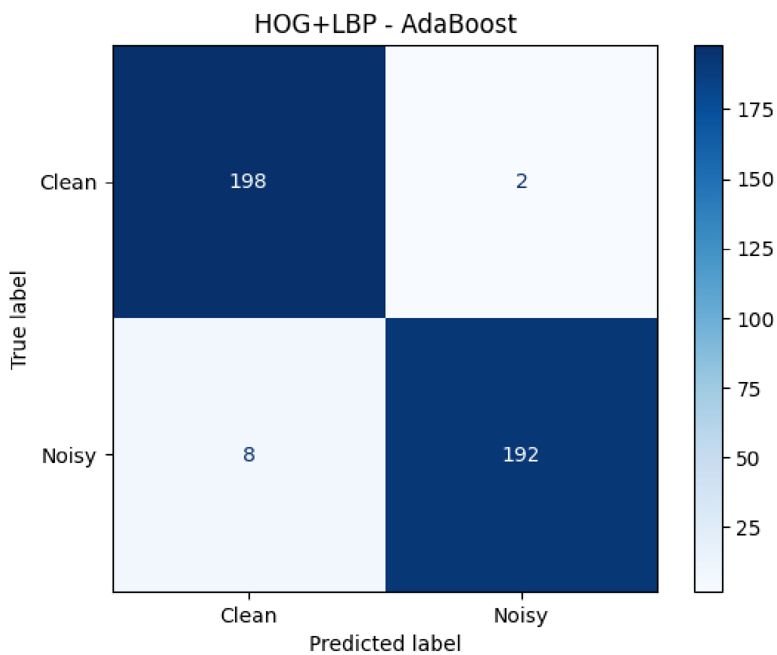

Fig 4. Confusion matrix of HOG+LBP for Ada Boost classifier.

Table 5. Performance of Deep learning models.

Model	Accuracy	Precision	Recall	F1 Score
Custom CNN	0.93250	0.9326	0.9325	0.9325
EfficientNetB0	0.5	0.25	0.5	0.3333
ResNet50	0.7475	0.7584	0.7475	0.7448
MobileNetV2	0.95	0.9511	0.95	0.95
DenseNet121	0.9075	0.912	0.9075	0.9072

Table 6. Comparison Table

Methods	Feature Extraction	Classifiers	Evaluation Matrices			
			Accuracy	Precision	Recall	F1 Score

Machine Learning	HOG+LBP	Ada Boost	**0.975**	**0.9754**	**0.975**	**0.975**
Deep learning		MobileNetV2	0.95	0.9511	0.95	0.95

5. CONCULSION

This research presented an automatic classifier which categorises clean and noisy signatures. It is crucial for signature document workflow management. Noisy handwritten signatures DocSig V2 can contain other elements, such as date, tampon, stamp, and other handwritten words. These add noise and obfuscate the process.

Our experiments showed that the feature combinations of HOG and LBP and AdaBoost performed best, achieving an accuracy of 97.5%. This accuracy was better than other deep learning methods, including MobileNetV2 (95%) and CNN (93.25% on a small dataset). This suggests that texture descriptors can be used to capture detail very well, and are suitable for small datasets.

Finally, classifying clean and noisy signatures is a post-processing step. It helps to identify and isolate bad and overlapping signature samples. It does this prior to applying noise reduction and reconstruction techniques. This makes the subsequent steps, such as denoising and reconstruction, work better. This ensures more accurate results. This also enhances the overall stability and accuracy of the system while preparing it for use in practical settings in offline noise-aware signature analysis.

Our research provides a basis for distinguishing among handwritten signatures as clean or noisy, but there are a number of possibilities for future work. Future research should focus on building bigger and more varied databases and including multilingual and multi-script signatures for wider usage. Models like Vision

Vision Transformers or a CNN-Transformer hybrid can be studied to model stroke variability. Generative adversarial networks (GANs) can be used to generate noisy training data. Addressing multiple tasks such as noise classification, noise removal and verification would enhance its usefulness. To make them practical, programmers should develop efficient models for mobile devices and scanners, particularly in multi-script environments. Finally, testing on various real documents across different domains, as well as providing explainable AI and forgery detection, can help build an end-to-end signature authentication system for real-world and forensic applications.

References

1. D. Engin, A. Kantarcı, S. Arslan, and H. K. Ekenel, “Offline signature verification on real-world documents,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2020. [Online]. Available: https://openaccess.thecvf.com/content_CVPRW_2020/html/w48/Engin_Offline_Signature_Verification_on_Real-World_Documents_CVPRW_2020_paper.html
2. H.-H. Kao and C.-Y. Wen, “An offline signature verification and forgery detection method based on explainable deep learning,” Appl. Sci., vol. 10, no. 11, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/11/3716
3. L. Liu, X. Yang, and Y. Wang, “Offline signature verification using deep metric learning,” Pattern Recognit., vol. 121, 2021, Art. no. 108192. [Online]. Available: https://doi.org/10.1016/j.patcog.2021.108192
4. A. Jain, S. K. Singh, and K. P. Singh, “Multi-task learning using GNet features and SVM classifier for signature identification,” IET Biometrics, vol. 10, no. 2, pp. 117–126, 2021. [Online]. Available: https://doi.org/10.1049/bme2.12007
5. S. Manna, S. Chattopadhyay, S. Bhattacharya, and U. Pal, “SWIS: Self-supervised representation learning for writer-independent offline signature verification,” in Proc. IEEE Int. Conf. Image Process. (ICIP), 2022. [Online]. Available: https://arxiv.org/abs/2202.13078
6. H. Li, Z. Zhang, and J. Chen, “TransOSV: Offline signature verification with transformers,” Pattern Recognit., vol. 144, 2023, Art. no. 109895. [Online]. Available: https://doi.org/10.1016/j.patcog.2023.109895
7. I. M. Alharbi, “Efficient handwritten signature identification using machine learning,” Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 3, 2023. [Online]. Available: https://doi.org/10.14569/IJACSA.2023.0140316
8. Y. Liu, M. Chen, and L. Zhang, “Deep learning for document image analysis: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 46, no. 5, pp. 2345–2365, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10159345
9. B. Despodov, D. Stojanov, and C. M. Bande, “Evaluating handwritten character recognition with Hu moments and K-nearest neighbors algorithm,” TEM J., vol. 13, no. 3, pp. 1813–1824, 2024. [Online]. Available: https://doi.org/10.18421/TEM133-10
10. A. Tripathi, T. Singh, R. R. Nair, and P. Duraisamy, “Improving early detection and classification of lung diseases with innovative MobileNetV2 framework,” IEEE Access, vol. 12, pp. 116202–116217, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3440577
11. H. Zhang, Y. Li, and X. Zhao, “Offline signature verification based on variational autoencoder,” arXiv preprint arXiv:2409.19754, 2024. [Online]. Available: https://arxiv.org/abs/2409.19754

12. K. G. de Moura, R. Oliveira, and P. Cavalcanti, "Offline handwritten signature verification using a stream-based approach," arXiv preprint arXiv:2411.06510, 2024. [Online]. Available: https://arxiv.org/abs/2411.06510
13. A. Singla and A. Mittal, "Exploring offline signature verification techniques: A survey," Multimedia Tools Appl., 2025. [Online]. Available: https://link.springer.com/article/10.1007/s11042-024-20454-x
14. K. Taifi, Y. Sabbar, H. Ardah, and A.-H. Abdel-Aty, "Enhanced deep learning-based breast cancer diagnosis and classification using modified DenseNet architectures," IEEE Access, vol. 13, pp. 137851–137868, 2025. [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3594827
15. N. C. Rathore, S. Sharma, and R. Gupta, "A hybrid machine learning framework for offline signature verification using gray wolf optimization," Sci. Rep., vol. 16, 2026. [Online]. Available: https://www.nature.com/articles/s41598-026-36163-4