Archives available at journals.mriindia.com

**International Journal on Advanced Computer Engineering and Communication Technology**

ISSN: 2278-5140
Volume 14 Issue 01, 2025

# Signature Verification System Using Machine Learning and Image Processing

Gourav G. Jadhav [1], Prathamesh D. Patil [2], Vijaykumar S. Kumbhar[3]
*Research Student [1, 2], Associate Professor [3]*
*[1,2,3] Department Of Computer Science , Shivaji University Kolhapur*
*gouravjadhav2982001@gmail.com[1], ppratham1922@gmail.com[2], vsk_csd@unishivaji.ac.in[3]*

**Peer Review Information**

*Submission: 16 Jan 2025*
*Revision: 13 Feb 2025*
*Acceptance: 12 March 2025*

**Keywords**

*Classification*
*HOG*
*Feature Extraction*
*Genuine Signature*
*Machine Learning*

**Abstract**

This paper presents a robust and automated signature verification system that integrates advanced machine learning and image processing techniques to enhance security and accuracy in signature authentication. The system undergoes a series of preprocessing steps, including grayscale conversion, noise reduction using Gaussian blur, adaptive thresholding for binarization, and feature extraction using the HOG technique. The extracted features are then utilized to train multiple classification models, including SVM, Random Forest, and XGBoost classifiers, to evaluate their effectiveness in signature verification. A GUI has been developed to facilitate seamless user interaction, allowing individuals to upload and verify their signatures in real time. The experimental results demonstrate that our proposed method achieves an accuracy exceeding 80%, making it a viable solution for secure and reliable authentication in applications such as financial transactions, document verification, and identity validation.

## Introduction

Signature verification is a crucial authentication method used in financial transactions, legal documents, and identity verification. Traditional manual verification methods are prone to human error, subjectivity, and inefficiency, making them unreliable for large-scale applications. Automated signature verification systems, leveraging machine learning and image processing techniques, offer a more accurate, efficient, and consistent alternative. These systems analyze signature patterns, extract unique features, and classify them to determine authenticity.

The proposed model in this research enhances signature verification by employing a systematic image processing pipeline. First, the signature images are converted to grayscale to simplify processing, followed by the application of Gaussian blur to remove noise and improve clarity. Adaptive thresholding is then used for binarization, which enhances the contrast between the signature and the background. To ensure uniformity, all images are resized to a fixed dimension of 128x128 pixels. After preprocessing, feature extraction is performed using the HOG, which captures essential structural details of the signature.

For classification, the system employs multiple machine learning algorithms, including SVM, Random Forest, and XGBoost. These models are trained on extracted features to distinguish

© 2025 The Authors. Published by MRI INDIA.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

between genuine and forged signatures. To evaluate the system's effectiveness, the performance of each classifier is compared, and the best model is selected based on accuracy and reliability. Additionally, a GUI is developed to allow users to upload signatures for real- time verification. The experimental results demonstrate that the proposed approach achieves an accuracy of over 80%, making it a viable and robust solution for secure authentication. The integration of image processing and machine learning techniques significantly enhances the accuracy, efficiency, and security of the signature verification process, providing a reliable alternative to manual verification methods.

**Methodology**

**1. Dataset Preparation**

A comprehensive dataset was collected, consisting of signature samples from multiple individuals to facilitate the training and evaluation of the verification model. To maintain organization, each individual's signatures were stored in separate folders, with image filenames labeled numerically for easy identification. This structured approach ensures efficient data retrieval and management. Before feeding the data into the machine learning models, preprocessing techniques were applied to enhance consistency and quality. This included converting images to grayscale to reduce complexity, applying noise reduction techniques such as Gaussian blur, and performing adaptive thresholding for binarization, ensuring clear signature contours. Additionally, all images were resized to a fixed dimension of 128 x 128 pixels to maintain uniformity across the dataset. By implementing these preprocessing steps, the dataset becomes more suitable for feature extraction and classification, ultimately improving the accuracy and robustness of the signature verification system.

**2. Preprocessing**

- **Convert signature images to grayscale**

Signature images are often colored or have unnecessary background details. Converting them to grayscale simplifies the image by removing color information and reducing computational complexity.

- **Apply Gaussian blur for noise reduction**

Gaussian blur smoothies an image by averaging the values of pixels within a small region. This technique helps minimize unwanted noise and accentuates important elements in the signature, such as strokes.

- **Perform adaptive thresholding for binarization**

Adaptive thresholding transforms the grayscale image into a black-and-white image. This step is essential for obtaining distinct signature patterns by accentuating dark strokes and eliminating background noise.

- **Resize images to a uniform dimension (128x128 pixels)**

Signatures can vary in size, so resizing them to a standard dimension of 128x128 pixels guarantees consistent input for feature extraction and model training.

- **Extract HOG features**

HOG is a feature extraction technique that captures the direction and intensity of edges in an image. It helps the model distinguish different signatures by analyzing stroke patterns and orientations

**3. Feature Extraction**:

Feature extraction is a crucial step in the signature verification process, as it helps in identifying distinguishing characteristics that differentiate one signature from another. In this study, feature extraction is performed using the HOG descriptor, a widely used technique for capturing essential edge and shape information. The HOG method works by dividing the image into small connected regions called cells, computing the gradient orientation within each cell, and forming a histogram that represents the distribution of gradient directions. This approach ensures that the model can effectively capture signature patterns, strokes, and overall structural details, making it highly suitable for handwritten signature verification.

Since the raw HOG features can be high-dimensional, Principal Component Analysis (PCA) is applied to reduce redundancy and ensure uniform feature dimensions. PCA is a dimensionality reduction technique that transforms the original high-dimensional feature space into a lower-dimensional one while retaining the most significant information. By selecting a fixed number of principal components, PCA eliminates noise and irrelevant variations, thereby improving computational efficiency and reducing the risk of overfitting. The use of PCA

ensures that the extracted feature vectors maintain consistency across all samples, allowing machine learning classifiers to process them more effectively. This combination of HOG for feature extraction and PCA for dimensionality reduction enhances the accuracy and robustness of the signature verification model.

## 4. Classification Models

**Three machine learning classifiers were trained and compared:**

In this study, three different machine learning models— SVM, Random Forest Classifier, and XGBoost Classifier—are used for signature verification. Each model has its unique strengths and is compared to determine the most effective classifier for accurate and reliable signature authentication.

- **SVM:** SVM is a supervised learning algorithm that is widely used for classification tasks. In this study, a linear SVM is used with probability estimation enabled, allowing the model to produce confidence scores for predictions.

**How SVM Works:**

1. SVM finds the optimal hyperplane that separates different classes (genuine and forged signatures) in the feature space.
2. It maximizes the margin (distance) between the closest data points (support vectors) of different classes to improve classification performance.
3. A linear kernel is chosen because signature verification often involves well-structured feature distributions.
4. Probability estimation is enabled to provide a confidence score for classification results, making the decision-making process more interpretable.

SVM is effective in handling high-dimensional data and works well when the dataset is relatively small but has well-separated classes. However, it may struggle with highly complex decision boundaries that require non-linear separations.

- **Random Forest Classifier:** The Random Forest Classifier is an ensemble learning method that constructs multiple decision trees during training and merges their predictions to improve accuracy and robustness.

**How Random Forest Works:**

1. The model generates multiple decision trees, where each tree is trained on a random subset of the data and features.
2. During classification, each tree votes on the class label, and the majority vote determines the final prediction.
3. Random Forest reduces overfitting by averaging multiple decision trees, which makes it more generalizable to new data.
4. It is highly interpretable, as feature importance scores can be extracted to understand which characteristics contribute most to signature verification.

Random Forest performs well when handling noisy or unstructured data, and its ensemble nature makes it resistant to overfitting. However, it requires more computational resources compared to simpler models like SVM.

- **XGBoost Classifier:** XGBoost is an advanced gradient boosting algorithm optimized for speed and accuracy. It is particularly effective for structured classification problems and has gained popularity in machine learning competitions due to its efficiency.

**How XGBoost Works:**

1. XGBoost builds multiple weak decision tree models sequentially, where each new tree corrects the errors of the previous trees.
2. The boosting process assigns higher weights to misclassified samples, forcing the model to focus more on difficult cases.
3. It employs regularization techniques to prevent overfitting, ensuring that the model generalizes well to new signature samples.
4. XGBoost is optimized for performance, using parallel computing and memory-efficient techniques to speed up training.
5.

XGBoost is known for its superior accuracy, especially when dealing with large and complex datasets. It requires careful tuning of hyper parameters to achieve optimal results, but once tuned, it often outperforms other classifiers.

**Comparison and Selection**

Each model has its own advantages and is evaluated based on accuracy, precision, recall, and F1- score.

1. SVM works well for structured data with clear decision boundaries.
2. Random Forest is robust against noise and

generalizes well.

3. XGBoost provides high accuracy but requires careful tuning.

## 5. Model Training and Testing

The dataset is divided into two subsets: 80% for training and 20% for testing to ensure a balanced evaluation of the model's performance. The training set is used to learn patterns from the extracted features, while the test set evaluates the generalization ability of the trained model. Feature extraction is performed using the HOG descriptor, which captures essential shape and edge information from signature images. These features serve as the input for machine learning models such as SVM, Random Forest, and XGBoost, which are trained to classify signatures as genuine or forged. After training, the models are tested on the 20% test set to measure their accuracy, precision, recall, and other performance metrics. This evaluation helps determine the model's effectiveness in distinguishing between authentic and fraudulent signatures, ensuring its reliability in real-world applications.

## 6. Results and Discussion

The performance of the trained models is evaluated using key classification metrics, including accuracy, precision, recall, and F1-score, to ensure a comprehensive analysisof their effectiveness in signature verification. Accuracy measures the overall correctness of predictions, while precision and recall assess the model's ability to correctly identify genuine signatures and minimize false positives. The F1-score, which is the harmonic mean of precision and recall, provides a balanced performance metric, especially when dealing with imbalanced datasets. Additionally, a feature matrix visualization is generated to represent the extracted patterns from the signature images, providing insights into the distinct characteristics used for classification. The comparative analysis of the models indicates that XGBoost outperforms other classifiers, achieving the highest accuracy due to its optimized gradient boosting mechanism. Random Forest follows with slightly lower accuracy, benefiting from its ensemble learning approach. SVM, while effective, performs the least among the three, possibly due to its sensitivity to high-dimensional feature spaces. These results suggest that XGBoost is the most suitable model for signature verification, offering superior accuracy and reliability in distinguishing between authentic and forged signatures.

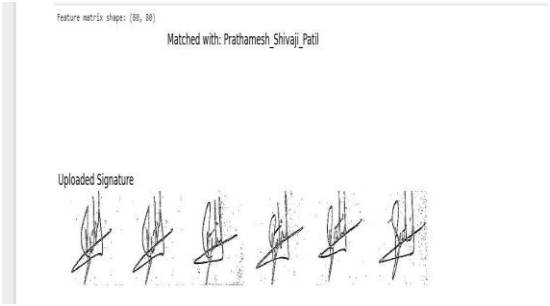


*Figure 1. Result*

A user-friendly GUI is developed using Tkinter. The system allows users to upload a signature image, which is then processed and compared against stored samples. If a match is found, the corresponding individual's name and signature are displayed; otherwise, a "Signature Not Matched" message is shown.

## 7. User Interface Implementation

A user-friendly GUI is designed to streamline the signature verification process. The interface provides an intuitive and efficient way for

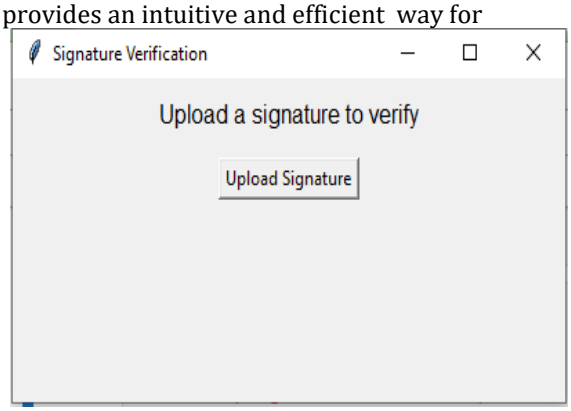


*Figure 2. User Interface*

users to interact with the system. It includes functionalities such as uploading a signature image, which allows users to input a scanned or digital signature for verification. Once the signature is uploaded, the system processes it by converting it to grayscale, applying preprocessing techniques, and extracting unique features. The extracted features are then compared against a database of stored signatures using trained machine learning models.

If a match is found, the system displays the corresponding person's name and the matched signature image. If no match is identified, the system informs the user that the signature does not match any stored samples. This interactive and automated approach enhances the efficiency and accuracy of the verification process, making it a

viable solution for secure authentication in various applications, such as banking, legal documents, and identity verification. Displaying results with a confidence score.

## Literature Review

### 1. Self-Supervised and Attention-Based Learning for Signature Verification

Self-supervised learning (SSL) has emerged as a powerful approach to enhance writer-independent signature verification. Chattopadhyay et al. (2022) [1] proposed SURDS, a self- supervised model that integrates attention-guided reconstruction and dual triplet loss to improve signature discrimination. This method enhances verification accuracy by reducing dependency on labeled data.

Similarly, Ji et al. (2022) [2] introduced Top-Rank Pair Learning, which ranks genuine and forged signature pairs to improve classification performance. Their findings suggest that top-rank learning prioritizes reliable signature features, improving robustness against forgery.

### 2. Feature Engineering and Distance-Based Classification

Feature engineering is fundamental in traditional machine learning-based verification. Parziale et al. (2024) [3] proposed Stability Modulated Dynamic Time Warping (SM-DTW), a feature alignment technique that adapts to signature variability, making verification more robust.

Additionally, Jain et al. (2021) [4] explored geometrical feature-based models combined with Artificial Neural Networks (ANN), emphasizing the importance of stroke curvature, aspect ratios, and signature boundaries.

### 3. Deep Learning-Based Signature Verification

Deep learning models, especially CNNs, have revolutionized signature verification. Salama (2023) [5] demonstrated that deep CNNs extract complex spatial features, outperforming classical models like SVM and Random Forest.

Similarly, Nam et al. (2020) [7] compared multiple CNN architectures, concluding that ResNet and MobileNet perform better due to their ability to capture intricate texture details.

### 4. Transfer Learning and Siamese Networks

Transfer learning has improved verification accuracy by leveraging pre-trained CNNs. Khalifa et al. (2020) [11] implemented triplet loss in a Siamese network, enhancing inter-class separability and improving writer-independent verification.

Another study by Borges et al. (2020) [12] optimized signature verification through genetic algorithm-based feature selection, improving classification efficiency.

### 5. Machine Learning Model Comparison for Signature Verification

Several studies have compared different machine learning classifiers for signature verification. Ferrer et al. (2012) [6] analyzed gray-level features and found that ensemble methods (XGBoost, Random Forest) outperform linear classifiers.

Additionally, Bertolini et al. (2010) [10] demonstrated that ensemble classifiers reduce forgery acceptance rates by combining predictions from multiple models.

### 6. Applications Beyond Signature Verification

Deep learning-based image recognition techniques have influenced signature verification. Studies like Sladojevic et al. (2016) [15] and Fuentes et al. (2017) [17] highlight CNN-based plant disease classification, which shares similarities with signature verification in feature extraction and pattern recognition.

## Conclusion

This research highlights the effectiveness of machine learning techniques in automating signature verification, reducing human error, and improving authentication accuracy. Among the tested models, XGBoost emerges as the most reliable classifier, achieving the highest accuracy due to its optimized gradient boosting framework. The results confirm that feature extraction using HOG, combined with machine learning classifiers, can effectively distinguish between genuine and forged signatures. However, there is scope for further improvements. Future research will focus on expanding the dataset to include more diverse handwriting styles and forgeries, enhancing generalization. Additionally, the integration of deep learning models, such as Convolutional Neural Networks (CNNs), Siamese Networks, or Triplet Loss-based architectures, will be explored to extract more complex signature features and further improve verification accuracy. Another important direction is improving robustness against skilled forgeries, ensuring that the system remains effective even against sophisticated fraudulent attempts. By refining preprocessing

techniques and employing advanced deep learning-based feature embedding's, the performance and security of automated signature verification systems can be significantly enhanced.
Model trained (svm) with accuracy: 66.67% and saved to signature_model.pkl Model trained (random_forest) with accuracy: 48.48% and saved to signature_model.pkl Model trained (xgboost) with accuracy: 45.45% and saved to signature_model.pkl

**References**

S. Chattopadhyay, S. Manna, S. Bhattacharya, and U. Pal, "SURDS: Self-Supervised Attention-guided Reconstruction and Dual Triplet Loss for Writer Independent Offline Signature Verification," *arXiv preprint arXiv:2201.10138*, 2022. [Online]. Available: https://arxiv.org/abs/2201.10138

X. Ji, Y. Zheng, D. Suehiro, and S. Uchida, "Revealing Reliable Signatures by Learning Top-Rank Pairs," *arXiv preprint arXiv:2203.09927*, 2022. [Online]. Available: https://arxiv.org/abs/2203.09927

A. Parziale, M. Diaz, M. A. Ferrer, and A. Marcelli, "SM-DTW: Stability Modulated Dynamic Time Warping for Signature Verification," *arXiv preprint arXiv:2405.11978*, 2024. [Online]. Available: https://arxiv.org/abs/2405.11978

A. Jain, S. K. Singh, and K. P. Singh, "Signature Verification using Geometrical Features and Artificial Neural Network Classifier," *arXiv preprint arXiv:2108.02029*, 2021. [Online]. Available: https://arxiv.org/abs/2108.02029

W. Salama, "Signature Verification Based on Deep Learning," *Alexandria Journal of Science and Technology*, 2023. [Online].

M. A. Ferrer, J. F. Vargas, A. Morales, and A. M. Alonso, "Robustness of Offline Signature Verification Based on Gray Level Features," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 966–977, 2012, doi: 10.1109/TIFS.2012.2189183.

K. D. Nam, S. R. Kang, and S. M. Koo, "A Study on Signature Verification Based on Deep Learning," *Journal of Information and Communication Convergence Engineering*, vol. 18, no. 1, pp. 19–24, 2020, doi: 10.6109/jicce.2020.18.1.019.

Y. M. Lui, D. Bolme, and B. Draper, "Signature Authentication Using Advanced Feature Extraction and Classification Methods," *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1307–1315, 2012, doi: 10.1016/j.patrec.2012.02.021.

H. Suh, J. Cho, and S. Lee, "A Survey on Deep Learning-Based Signature Verification," *Journal of Visual Communication and Image Representation*, vol. 67, 2020, doi: 10.1016/j.jvcir.2019.102747.

D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Reducing Forgeries in Writer-Independent Offline Signature Verification Through Ensemble of Classifiers," *Pattern Recognition*, vol. 43, no. 1, pp. 387–396, 2010, doi: 10.1016/j.patcog.2009.05.018.

A. B. Khalifa, T. H. Chalidabhongse, and P. Chamnongthai, "A Deep Learning Approach to Offline Signature Verification Using Triplet Loss Function," *International Journal of Biometrics*, vol. 12, no. 4, pp. 287–308, 2020, doi: 10.1504/IJBM.2020.109867.

J. O. Borges, R. S. de Araujo, and T. Oliveira-Santos, "Feature Selection for Offline Signature Verification Using Genetic Algorithms," *Neural Computing and Applications*, vol. 32, no. 7, pp. 3087–3097, 2020, doi: 10.1007/s00521-019-04125-5.

D. Kumar, M. Srivastava, and R. A. G. Raj, "Offline Signature Verification Using Multi-Scale Deep Learning," *Multimedia Tools and Applications*, vol. 79, no. 21, pp. 15127–15142, 2020, doi: 10.1007/s11042-019-7366-1.

S. Ishak, M. H. F. Rahiman, S. N. A. M. Kanafiah, and H. Saad, "Leaf Disease Classification Using Artificial Neural Network," *Jurnal Teknologi*, vol. 77, no. 17, 2015, doi: 10.11113/jt.v77.6463.

S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Computational Intelligence and Neuroscience*, vol. 2016, 2016, doi: 10.1155/2016/3289801.

S. A. Wallelign, M. Polceanu, and C. Buche, "Soybean Plant Disease Identification Using Convolutional Neural Network," in *FLAIRS Conference*, pp. 146-151, 2018. [Online]. Available: https://www.enib.fr/~buche/article/FLAIRS_18.pdf

A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests' Recognition," *Sensors*, vol. 17, no. 9, 2017, doi: 10.3390/s17092022.

A. Dhakal and S. Shakya, "Image-Based Plant Disease Detection with Deep Learning," *International Journal of Computer Trends and Technology*, vol. 61, no. 1, pp. 26-29, 2018, doi: 10.14445/22312803/IJCTT-V61P105.

J. Boulent, S. Foucher, J. Théau, and P.-L. St-Charles, "Convolutional Neural Networks for the Automatic Identification of Plant Diseases," *Frontiers in Plant Science*, vol. 10, pp. 941, 2019, doi:10.3389/fpls.2019.00941.