Signature Recognition Using Convolutional Neural Networks and Comparative Analysis with Traditional Feature Extraction Methods

Laiba Batool
Department of Computer Science
FAST NUCES
Ialamabad, Pakistan
i211781@nu.edu.pk

***Abstract*—This paper presents a comparative analysis of signature recognition techniques, evaluating Convolutional Neural Networks (CNN) against traditional feature extraction methods including Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT). Using a dataset of 128 classes containing both genuine and forged signatures, we implement and compare these methods based on accuracy, precision, recall, and F1-score metrics. Our results demonstrate that HOG features combined with SVM classification significantly outperform both CNN and SIFT approaches, achieving 78.18% accuracy compared to 1.52% and 3.64%, respectively. Additionally, we analyze how these methods perform differently when distinguishing between genuine and forged signatures, finding that all methods perform better on genuine signatures. This work contributes to the understanding of effective techniques for signature verification systems and provides insights into their practical applications.**

***Index Terms*—signature recognition, signature verification, convolutional neural networks, histogram of oriented gradients, scale-invariant feature transform, support vector machine, feature extraction**

I. Introduction

Signature verification is a critical aspect of biometric authentication systems, widely used in various domains such as banking, legal documentation, and security applications. The ability to accurately recognize and verify signatures helps prevent fraud and ensures document authenticity. Traditional signature verification was performed manually by human experts, which is time-consuming and subject to human error. The automation of this process through machine learning techniques offers potential improvements in both efficiency and accuracy.

This paper focuses on implementing and comparing different approaches to signature recognition. Specifically, we explore three methods:

1) Convolutional Neural Networks (CNN), a deep learning approach that automatically extracts relevant features from raw signature images
2) Histogram of Oriented Gradients (HOG) feature extraction combined with Support Vector Machine (SVM) classification
3) Scale-Invariant Feature Transform (SIFT) feature extraction, also combined with SVM classification

By conducting this comparative analysis, we aim to identify the most effective method for signature recognition and understand the strengths and limitations of each approach. Additionally, we examine the performance differences between genuine and forged signature recognition, which is crucial for practical applications in fraud detection.

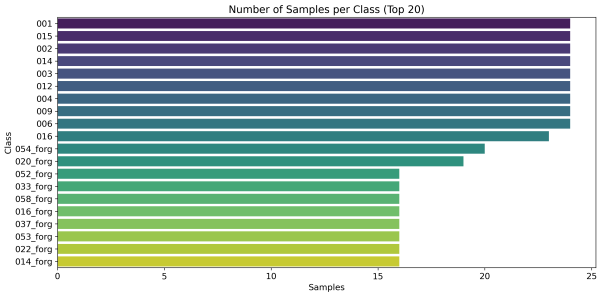

Fig. 1. Number of samples per class (top 20 classes) in the dataset.

II. Methodology

A. Dataset Description

For this study, we utilized the Signature Verification Dataset from Kaggle [**?**]. The dataset contains signatures from multiple individuals, with each person having both genuine signatures and corresponding forgeries.

As shown in Fig. 1, the dataset consists of 128 distinct classes (64 individuals with both genuine and forged signatures). Each genuine class contains between 12-24 signature samples, while forged classes typically contain 8-16 samples. This balanced distribution allows for effective training and testing of our models on both authentic and forged signatures.

Fig. 2 illustrates examples of genuine and forged signature pairs from the dataset. As visible in the samples, forgeries often closely resemble the genuine signatures but contain subtle differences in stroke patterns, pressure points, and overall consistency that our models must learn to detect.

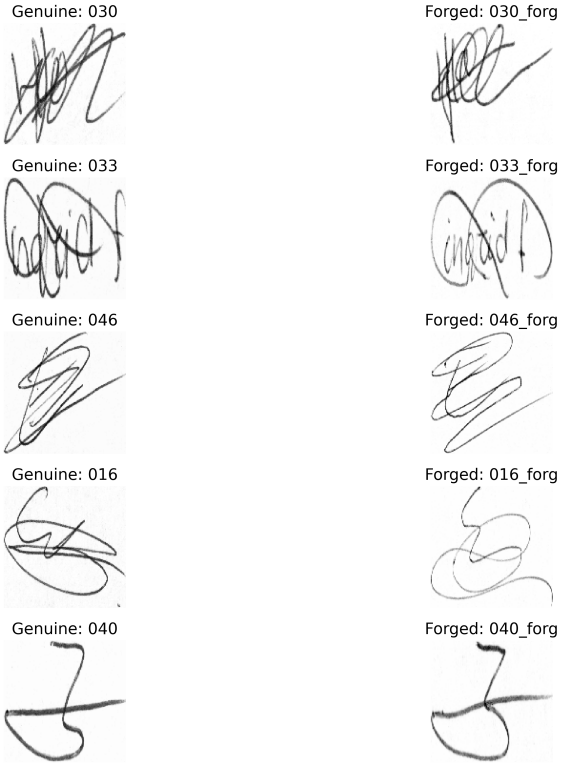

Fig. 2. Examples of genuine and forged signature pairs from the dataset.

In total, the dataset contains 1,649 signature images, which we split into training (80%) and testing (20%) sets, ensuring stratified sampling to maintain the same class distribution in both sets. This resulted in 1,319 training samples and 330 testing samples.

B. Preprocessing Steps

All signature images underwent several preprocessing steps to ensure consistency and improve model performance:

1) **Grayscale Conversion**: All images were converted to grayscale to reduce dimensionality and focus on structural features rather than color information.
2) **Resizing**: Images were resized to 128×128 pixels to maintain a uniform input size for all models.
3) **Normalization**: Pixel values were normalized to the range [0,1] by dividing by 255, which helps with model training stability and convergence.
4) **Channel Expansion**: For the CNN model, a channel dimension was added to the grayscale images to fit the input requirements of convolutional layers.

C. CNN Architecture

The CNN model architecture consists of multiple convolutional, pooling, and fully connected layers designed to extract hierarchical features from signature images:

The model was compiled using:

- Optimizer: Adam with default learning rate
- Loss function: Sparse categorical cross-entropy
- Metrics: Accuracy

TABLE I
CNN ARCHITECTURE

Layer	Parameters	Output Shape
Conv2D	32 filters, 3x3, ReLU, same padding	128x128x32
Conv2D	32 filters, 3x3, ReLU, same padding	128x128x32
MaxPooling2D	2x2 pool size	64x64x32
Dropout	rate=0.25	64x64x32
Conv2D	64 filters, 3x3, ReLU, same padding	64x64x64
Conv2D	64 filters, 3x3, ReLU, same padding	64x64x64
MaxPooling2D	2x2 pool size	32x32x64
Dropout	rate=0.25	32x32x64
Conv2D	128 filters, 3x3, ReLU, same padding	32x32x128
Conv2D	128 filters, 3x3, ReLU, same padding	32x32x128
MaxPooling2D	2x2 pool size	16x16x128
Dropout	rate=0.25	16x16x128
Flatten	-	32,768
Dense	512 units, ReLU	512
Dropout	rate=0.5	512
Dense	128 units, softmax	128

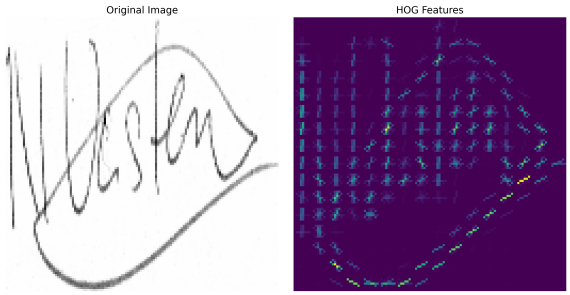

Fig. 3. Original signature image (left) and its corresponding HOG feature visualization (right).

During training, we employed several techniques to improve performance:

- Data augmentation (rotation, width/height shifts, shear, zoom)
- Early stopping based on validation loss with patience of 10 epochs
- Model checkpointing to save the best model
- Learning rate reduction when validation loss plateaued

D. HOG Feature Extraction

Histogram of Oriented Gradients (HOG) is a feature descriptor that captures local gradient orientation statistics [2]. For our implementation, we used the following parameters:

- Orientations: 9 bins
- Pixels per cell: 8×8
- Cells per block: 2×2
- Block normalization: L2-Hys

Fig. 3 shows an original signature image and its corresponding HOG feature visualization. The HOG features effectively capture the gradient structures that represent the distinctive aspects of the signature, such as stroke directions and patterns.

E. SIFT Feature Extraction

Scale-Invariant Feature Transform (SIFT) identifies keypoints in an image that are invariant to scaling, rotation, and

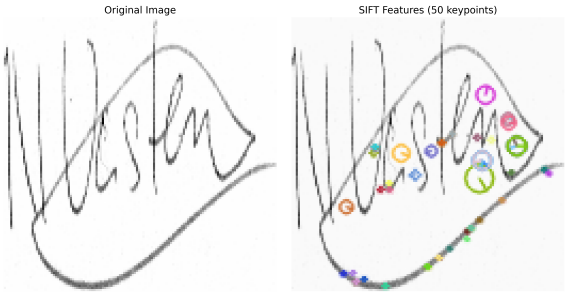

Fig. 4. Original signature image (left) and the detected SIFT keypoints (right).

translation [1]. We implemented the following steps:

1) Extract SIFT keypoints and descriptors from all training images
2) Create a visual vocabulary using K-means clustering with 100 clusters
3) Generate bag-of-visual-words histograms for each image
4) Use these histograms as feature vectors for classification

Fig. 4 displays an original signature image alongside the detected SIFT keypoints. The colored circles represent keypoints of different scales and orientations, which are used to create distinctive feature vectors for each signature.

F. SVM Classification

For both HOG and SIFT features, we used Support Vector Machine (SVM) classification [**?**] with the following parameters:

- Kernel: Radial Basis Function (RBF)
- C: 10 (for HOG+SVM)
- Gamma: 'scale'
- Probability: True (to enable probability estimates)

G. Evaluation Metrics

To comprehensively evaluate and compare the three methods, we used the following metrics:

- Accuracy: The proportion of correctly classified signatures
- Precision: The proportion of positive identifications that were actually correct
- Recall: The proportion of actual positives that were correctly identified
- F1-score: The harmonic mean of precision and recall

Additionally, we analyzed the performance separately for genuine and forged signatures to understand each method's capabilities in these distinct scenarios.

III. Results

A. CNN Performance

The CNN model showed limited performance on the signature recognition task, achieving only 1.52% accuracy on the test set.

Fig. 5 displays the training and validation accuracy/loss curves for the CNN model. Despite training for multiple epochs, the model failed to learn effectively, with validation

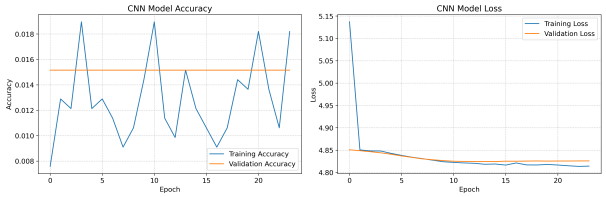

Fig. 5. Training and validation accuracy/loss curves for the CNN model.

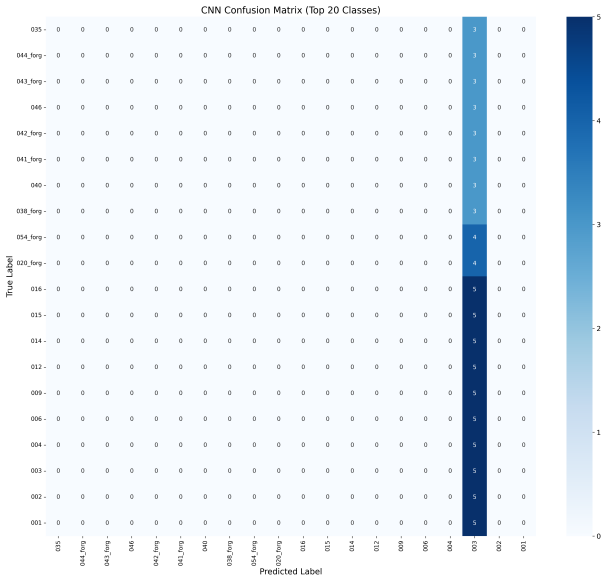

Fig. 6. Confusion matrix for the CNN model (top 20 classes).

accuracy plateauing at around 1.5% and showing signs of underfitting rather than overfitting.

The confusion matrix in Fig. 6 reveals that the CNN model primarily predicted a single class for most inputs, indicating that it failed to learn discriminative features for different signatures.

B. HOG+SVM Performance

The HOG+SVM approach substantially outperformed the CNN model, achieving 78.18% accuracy on the test set.

Fig. 7 shows the confusion matrix for the HOG+SVM model. Unlike the CNN, this approach successfully classified most signature classes, as evidenced by the strong diagonal pattern in the matrix.

Fig. 8 illustrates the performance difference between genuine and forged signature recognition. The HOG+SVM model achieved 91.38% accuracy on genuine signatures but only 63.46% on forged signatures, indicating that forgery detection remains challenging even for the best-performing method.

C. SIFT+SVM Performance

The SIFT+SVM approach performed poorly, achieving only 3.64% accuracy on the test set.

Fig. 9 presents the confusion matrix for the SIFT+SVM model, showing scattered predictions with limited accuracy across all classes.

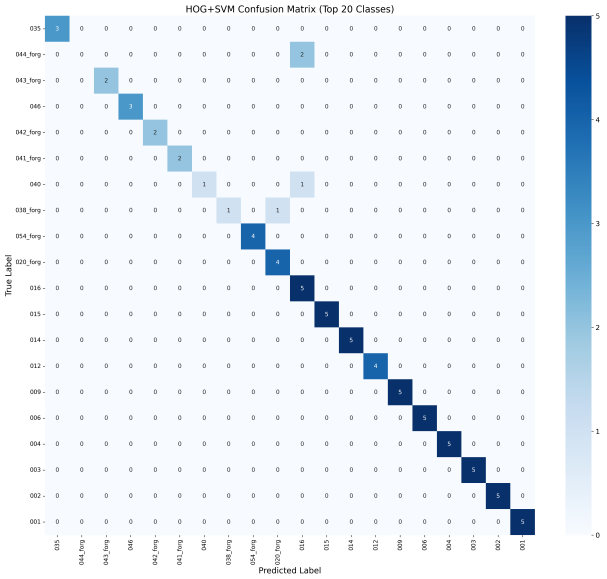

Fig. 7. Confusion matrix for the HOG+SVM model (top 20 classes).

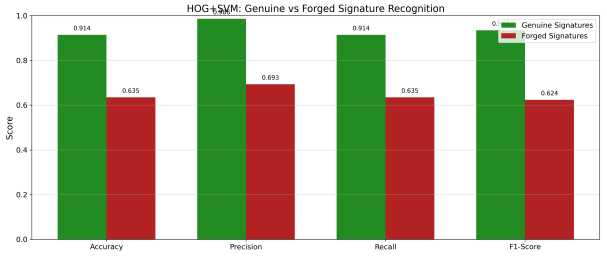

Fig. 8. Performance comparison between genuine and forged signature recognition using HOG+SVM.

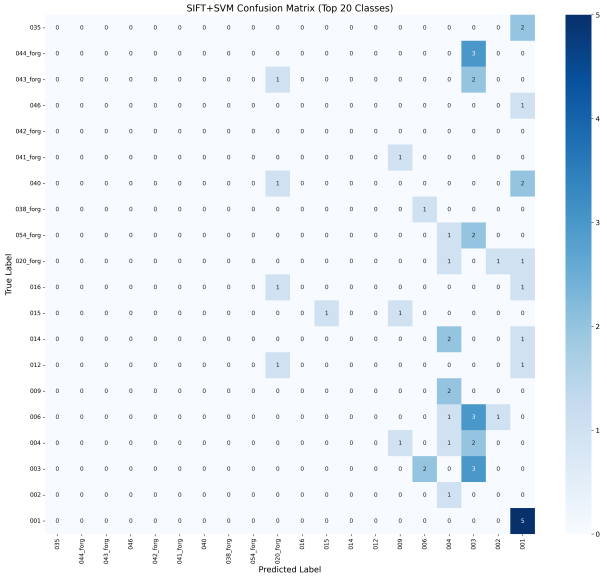

Fig. 9. Confusion matrix for the SIFT+SVM model (top 20 classes).

D. Comparative Analysis

Fig. 10 provides a side-by-side comparison of all three methods across the evaluation metrics. The HOG+SVM approach clearly outperformed both CNN and SIFT+SVM across all metrics, as summarized in Table II.

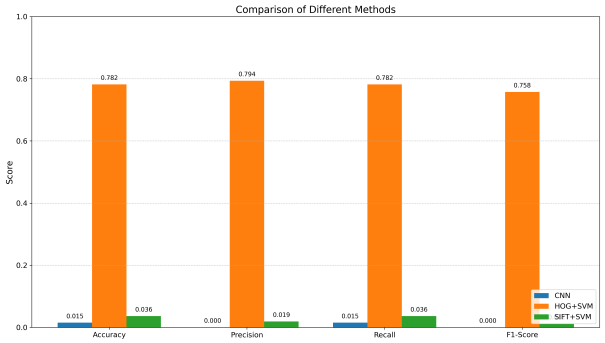

Fig. 10. Performance comparison of CNN, HOG+SVM, and SIFT+SVM approaches.

TABLE II
COMPARISON OF PERFORMANCE METRICS ACROSS METHODS

Method	Accuracy	Precision	Recall	F1-Score
CNN	1.52%	0.02%	1.52%	0.05%
HOG+SVM	78.18%	79.37%	78.18%	75.76%
SIFT+SVM	3.64%	1.91%	3.64%	1.16%

IV. Discussion

A. Model Performance Analysis

The significant performance difference between the three approaches warrants careful analysis. Several factors may explain these results:

1) **CNN Limitations**: Despite the theoretical advantages of CNNs in image recognition tasks, our CNN model performed poorly on signature recognition. This may be attributed to:
   - Insufficient training data for the complexity of the task and number of classes (128)
   - Difficulty in learning discriminative features from signatures that often have subtle differences
   - Challenges in generalizing across different writing styles within the same class
2) **HOG+SVM Effectiveness**: The superior performance of HOG+SVM suggests that:
   - Gradient-based features are particularly well-suited for capturing the distinctive aspects of signatures
   - HOG features effectively represent the stroke patterns and structural characteristics
   - SVMs are effective at creating decision boundaries between similar signature classes
   - The combination of engineered features with traditional machine learning may be more sample-efficient for this task
3) **SIFT+SVM Limitations**: The poor performance of SIFT+SVM could be due to:
   - The bag-of-visual-words approach may lose spatial information critical for signature recognition

- SIFT keypoints may focus on local details without capturing the global structure of signatures
- The clustering approach for creating visual words may not effectively represent signature characteristics

B. Genuine vs. Forged Signature Recognition

An important finding in our study is the difference in performance between genuine and forged signature recognition:

- All methods performed better on genuine signatures than on forged ones
- HOG+SVM achieved 91.38% accuracy on genuine signatures but only 63.46% on forged ones
- This performance gap highlights the inherent difficulty in detecting forgeries, which are deliberately designed to mimic genuine signatures

These results align with the real-world challenges in signature verification systems, where detecting sophisticated forgeries remains difficult even for human experts.

C. Challenges and Limitations

Several challenges were encountered during this study:

1) **Dataset Limitations**: While the dataset contained both genuine and forged signatures, the number of samples per class was relatively small, limiting the model's ability to learn robust representations.
2) **Computational Resources**: Training the CNN model required significant computational resources, particularly for the large number of parameters (over 17 million).
3) **Hyperparameter Optimization**: Due to resource constraints, extensive hyperparameter tuning was not performed, potentially limiting the performance of all models.
4) **Class Imbalance**: Variations in the number of samples per class could have affected model training and evaluation.

V. Conclusion

This study compared three approaches for signature recognition: CNN, HOG+SVM, and SIFT+SVM. Our findings demonstrate that HOG feature extraction combined with SVM classification significantly outperforms both deep learning (CNN) and other traditional feature extraction methods (SIFT) for this task.

The HOG+SVM approach achieved 78.18% overall accuracy, with 91.38% accuracy on genuine signatures and 63.46% on forged signatures. This substantial performance gap between genuine and forged signature recognition highlights the inherent challenge in detecting sophisticated forgeries.

Despite the growing popularity of deep learning for image recognition tasks, our results suggest that traditional feature engineering approaches may still be more effective for specialized tasks like signature recognition, particularly when training data is limited.

Future work could explore:

1) Hybrid approaches combining deep learning and traditional feature extraction
2) Advanced data augmentation techniques specific to signature images
3) Ensemble methods that leverage the strengths of multiple approaches
4) Transfer learning from pre-trained networks to address the limited data challenge

These findings contribute to the growing body of knowledge on signature verification systems and provide practical insights for implementing effective signature recognition in real-world applications.

Acknowledgment

We would like to acknowledge the creators of the Signature Verification Dataset for making their data publicly available for research purposes.

References

[1] R. Reni, "Signature Verification Dataset," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/robinreni/signature-verification-dataset

[2] T. M. Leyn, "Signature Verification using CNN, SNN, CSNN," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/code/tmleyncodes/signature-verification-using-cnn-snn-csnn