© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714765

An Offline Signature Verification Using Hybrid Symbolic and Deep Feature Representation

SMT. B. MANASA[1], Y. SIRI CHANDANA[2], Y. JITHENDRA VINAY KUMAR[3], S. HARSHA VARDHAN[4], R. NIRMITHA SARASWATHI[5]

[1]*Assistant Professor, R.V.R & J.C. College of Engineering*

[2, 3, 4, 5]*Information Technology, R.V.R & J.C College of Engineering*

Abstract - Offline signature verification is a challenging problem in the field of biometric recognition due to the presence of a lot of intra-writer variations and skilled forgeries. Conventional approaches in signature verification using only symbolic features and a predefined measure of similarity often fail to handle the complexities of visual patterns. To overcome this problem, we propose a new framework for signature verification using a combination of symbolic and convolutional neural network features. The symbolic features are able to capture the structural and statistical properties of signatures, whereas the convolutional neural network features are able to capture the high-level spatial features of signatures. The proposed framework uses a weighted feature-level fusion of both representations. The proposed framework achieves better performance in the presence of limited samples using the cosine similarity function and optimizes the decision threshold using the average error rate.

Keywords - Offline Signature Verification, Symbolic Feature Representation, Deep Feature Extraction, Feature-Level Fusion, Cosine Similarity, Few-Shot Learning.

I.INTRODUCTION

Handwritten signatures are still one of the most widely used biometric characteristics for verification purposes in legal, banking, and administrative systems. Offline signature verification using only static images is more difficult compared to online verification, as dynamic information like pressure and velocity is not available. Genuine signatures show high intra-writer variability, while skilled forgeries tend to closely imitate genuine ones, making it difficult to distinguish between them [13], [14]. There are already existing surveys that comprehensively review both online and offline handwriting recognition techniques [9], [22].

In the initial attempts for offline signature verification, geometric, statistical, and texture-based handcrafted features were combined with distance-based classifiers [11], [12]. These were effective but still suffered from difficulties in handling noise and distortion issues [14]. Symbolic data analysis was also introduced to incorporate interval-based representations for handling natural variations in genuine signatures [13]. This was extended to incorporate fuzzy similarity measures to deal with uncertainties, which improved their robustness compared to their crisp counterparts [1]. However, symbolic-fuzzy approaches were found to be limited in handling spatial complexity [1], [7], [8].

Deep learning methods, especially CNNs, have demonstrated good performance in solving image recognition problems by learning hierarchical representations of features [15], [16], [17]. When applied to signature verification, CNNs were found to be effective in increasing robustness against skilled forgeries but were found to be data-intensive, which may not be feasible under all scenarios [16], [17]. Following these observations, this work focuses on presenting a hybrid approach that incorporates symbolic features with deep learning features of CNNs, which are complementary to each other. Symbolic features are known for their interpretability and robustness under limited data, while deep learning features are known for their discriminative spatial patterns. A hybrid approach is proposed that incorporates feature-level fusion, multi-prototype reference modelling, and threshold optimization for balanced verification performance under few-shot scenarios [20].

II. RELATED WORK

Offline signature verification has been explored using various methods, from handcrafted features to deep learning-based approaches. In earlier days, features such as contours, projection profiles, and texture features, combined with classifiers such as distance-based classifiers [11], [12] were used. These methods achieved good performance, but they failed to deal with intra-writer variations and skilled forgeries, which are the most difficult problems in offline signature verification [1, 2, 13, 14].

The concept of symbolic data analysis has introduced interval-based features that can deal with intra-writer variations in genuine signatures [7]. Similarity measures using fuzzy multi-sets have also been explored for signature verification [2]. Symbolic features for signature verification achieved better robustness than single-valued features. The use of interval symbolic features with fuzzy similarity measures [1] is one such important achievement in signature verification that achieved gradual decision boundaries with noise tolerance. However, these handcrafted features and fuzzy rules are not flexible enough to deal with complex datasets [1]. Fuzzy sets are widely used in biometric systems for handling uncertainty [8].

In parallel, convolutional neural networks (CNNs) have shown promising performance in discriminative learning directly from images [2], [6], [9]. Although they are effective against skilled forgeries, deep learning-based approaches often demand large datasets, which are usually unavailable in real-world scenarios [16], [17]. Recently, hybrid approaches that integrate shallow and deep features have been proposed [10], [20], exploiting the merits of symbolic features' interpretability and deep features' representational power. This paper continues this trend by presenting a novel framework for symbolic-deep fusion, similarity-based verification, and threshold optimization, targeted for few-shot offline signatures.

III. PROPOSED METHODOLOGY

A. Overall Framework

The proposed offline signature verification system is based on a hybrid framework in which symbolic features are used in combination with deep features. Symbolic features are used to represent the features of the signature in a more meaningful way. On the other hand, deep features are learned using a pre-trained ResNet-18 model. These features are then fused using a weighted fusion technique.

The proposed signature verification system is based on a writer-dependent verification approach in which few-shot verification is used. In few-shot verification, only three signatures are used from each writer for verification. The dataset of each writer is divided into three parts:

- Training (Reference) Signatures – a few genuine signatures.
- Genuine Test Signatures – a few genuine signatures.
- Forgery Signatures – a few forged signatures.

1. Algorithm Representation

Algorithm 1: Hybrid Offline Signature Verification Framework

Input: Test signature image
Output: Verification decision (Genuine / Forgery)

1. Preprocess image:
- Convert to grayscale
- Apply Otsu thresholding
- Resize to 224×224 pixels

2. Extract symbolic features:
- Aspect ratio (bounding box width/height)
- Ink density (foreground pixel ratio)
- Edge density (Canny edge detection)
- Gray-level mean and standard deviation
- Hu invariant moments (log-transformed)

3. Extract deep features:
- Convert grayscale to RGB
- Normalize using ImageNet statistics
- Pass through ResNet-18 (remove final FC layer)
- Obtain deep feature vector from global average pooling

4. Normalize features:
- Standardize symbolic features using writer-dependent mean & std
- Apply L2 normalization to symbolic and deep vectors

5. Fuse features:
- $F = [\alpha \cdot S_{norm}, (1-\alpha) \cdot D_{norm}]$
- Apply final L2 normalization

6. Similarity computation:
- Compare fused test vector with all reference prototypes
- Compute cosine similarity scores

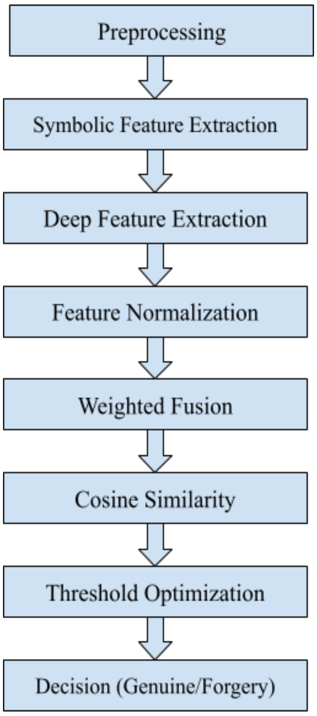
7. Decision rule:
- $Score_t = \max_i Sim(F_t, F_{ref_i})$

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714765

- If $Score_t \geq \tau \rightarrow$ Genuine
- Else $\rightarrow$ Forgery

8. Threshold optimization:
- Evaluate candidate thresholds
- Select τ^* that minimizes Average Error
- Rate (AER)

2. Block Diagram



3. Symbolic Feature Set

Table 1: Symbolic Features Used in the Framework

Feature Type	Description
Aspect Ratio	Width/Height of bounding box
Ink Density	Ratio of ink pixels to total pixels
Edge Density	Ratio of edge pixels detected by Canny operator
Gray-level Mean	Average intensity of grayscale image
Gray-level Std. Dev	Variation in intensity values
Hu Moments	Shape descriptors invariant to scale/rotation

B. Data Organization and Few-Shot Reference Modeling

For each of the writers, a limited number of genuine signatures are chosen as reference samples. Unlike the conventional approach of representing a writer using a single prototype of a signature, the proposed system uses a multi-prototype reference model. Here, a number of genuine signatures of a writer are stored as individual reference prototypes.

This approach enables the proposed system to handle the natural variations in signatures of a person. These variations could be in the thickness of the writing strokes, their slant, and the actual size of the signatures. The proposed system compares the test signature with all the reference signatures of a person and uses the maximum similarity for the final match. This increases the robustness of the proposed system in a few-shot setting. Mathematically, let:

- $\{Fr1, Fr2, ..., Frk\}$ be the fused feature vectors of the k reference signatures of a writer.
- Ft be the fused feature vector of the test signature.

The final matching score is computed as:

$$\text{Score} = \max_{i \in \{1,\dots,k\}} \left(\mathrm{Sim}\left(F_t, F_{r_i}\right)\right)$$

The above representation of matching score corresponds to the actual implementation of the proposed system.

C. Image Preprocessing

Each offline signature is first converted to its grayscale representation. This is done to simplify the process, but it retains the essential information required for both symbolic and deep feature extraction.

Otsu's thresholding is used to segment the signature ink from the background and generate a binary image [14]. This is a global thresholding technique that automatically finds the best threshold to segment the ink strokes from the background, separating the signature from the background [9].

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714765

Unlike thresholding, where the threshold has to be manually set, in Otsu's thresholding, the threshold is set in such a way that it minimizes the variance of the pixel classes, or maximizes the variance between the classes. This makes the system robust to changes in lighting, scanning, and ink, such that the signature strokes can be extracted cleanly for symbolic and deep feature extraction.

Formally, the optimal threshold $t*$ can be defined as:

$$t^* = \arg\min_t \sigma_w^2\,(t)$$

where the within-class variance is defined by:

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

Here, $\omega_1(t)$ and $\omega_2(t)$ denote the probabilities of the two classes i.e. the foreground and the background while $\sigma_1^2(t)$ and $\sigma_2^2(t)$ denote their respective variances.

Another equivalent definition of Otsu's method is to maximize the following expression, representing the between classes variance:

$$\sigma_b^2(t) = \omega_1(t)\omega_2(t)\big(\mu_1(t) - \mu_2(t)\big)^2$$

where μ_1 and μ_2 are the mean intensities of the two classes. The threshold t^* that maximizes $\sigma_b^2\ (t)$or minimizes $\sigma_w^2(t)$is selected as the optimal binarization value.

The binary image is further utilized in the following ways:

- To locate the foreground pixels representing the signature strokes.
- To calculate the bounding box covering the signature.
- To enable the accurate computation of structural symbolic features.

At the same time, the grayscale image is retained and is subsequently transformed into a three-channel image for compatibility with the convolutional neural network in the deep feature extraction.

D. Symbolic Feature Extraction

The symbolic features offer a concise representation of the signature's structure and ink distribution. These features are derived from the binarized image, which is obtained during the pre-processing step, and they incorporate the geometry, stroke density, edge information, and invariance of the shape [11].

Aspect Ratio

$$AspectRatio = \frac{w}{h}$$

Ratio of the width to the height of the bounding box, describing the overall orientation.

Ink Density

$$InkDensity = \frac{N_f}{N_t}$$

Proportion of ink pixels, related to the stroke width and pressure.

Edge Density

$$EdgeDensity = \frac{N_e}{N_t}$$

Proportion of edge pixels, related to the stroke sharpness and complexity.

Gray-Level Statistics

$$\mu = \frac{1}{M}\sum_{x,y} I(x,y)$$

$$\sigma = \sqrt{\frac{1}{M}\sum_{x,y}(I(x,y) - \mu)^2}$$

Mean and standard deviation of the grayscale values.

Hu Invariant Moments

$$\phi_i' = -\mathrm{sign}(\phi_i)\cdot\log_{10}(|\phi_i| + \varepsilon)$$

Seven shape descriptors that are invariant to translation, rotation, and scaling [18].

Final Symbolic Feature Vector

The symbolic feature vector S is defined as a combination of geometric features, statistical features, and invariant features. These features are aspect ratio, ink density, edge density. Additionally, the features include mean (μ), standard deviation (σ), and Hu's seven invariant moments (ϕ1…ϕ7). These features are a compact representation of the signature images.

E. Deep Feature Extraction

Although symbolic features represent global structural properties, they are limited when it comes to modeling fine-grained spatial patterns. To address this, deep features are extracted by using a pre-trained ResNet-18 convolutional neural network [16].

Network Selection

The ResNet18 is used because it is efficient and has residual connections [17]. The final fully connected layer is removed, and the output of the global average pooling layer is used as the deep feature vector.

Input Preprocessing

- Grayscale signature → RGB Image (3 channels).
- Resize image to 224×224 pixels (standard for ResNet-18).
- Normalized using ImageNet mean and standard deviation.

Feature Extraction

The preprocessed image I_{RGB} fed through ResNet-18. This gives us the feature vector:

$$D = f_{ResNet-18}(I_{RGB})$$

where D is a vector representing the image features that incorporate hierarchical patterns of strokes and texture.

F. Feature Normalization and Fusion

Symbolic and deep features differ in scale and distribution, so normalization and fusion are required before similarity computation.

Symbolic Feature Standardization

The symbolic features of different writers are standardized using the mean and standard deviation of the reference signatures of the writers.

$$S' = \frac{s - \mu_s}{\sigma_s - \varepsilon}$$

where μ_s and σ_s are writer-dependent statistics.

L2 Normalization

Both the symbolic and deep features are normalized to unit length:

$$\hat{x} = \frac{x}{||x||}$$

Weighted Fusion

The normalized symbolic features (S) and deep features (D) are fused together:

$$F = \left[\alpha \cdot \hat{S}, (1 - \alpha) \cdot \widehat{D}\right]$$

where $\alpha \in [0,1]$ controlling the relative importance of symbolic features and deep features [20].

Final Normalization

The fused vector F is normalized again using the L2 normalization technique.

$$\widehat{F} = \frac{F}{||F||}$$

G. Similarity Computation and Decision

Once fused feature vectors are obtained, verification is formulated as a similarity measurement problem.

1) Multi-Prototype Reference Modeling

Each writer is represented by multiple real reference signatures:

$$\{\widehat{F_1}, \widehat{F_2}, \dots, \widehat{F_K}\}$$

where each $\widehat{F_\iota}$ is a normalized fused reference vector. The test signature vector is $\widehat{F_t}$.

2) Cosine Similarity

The similarity between two feature vectors a and b is defined as:

$$Sim(a, b) = \frac{a \cdot b}{|a||b|}$$

Since the vectors are L2-normalized, the denominator is 1.

3) Maximum Similarity Rule

The final score for the test signature is given by:

$$Score_t = \max_{i \in \{1,\dots,K\}} S\,im\left(\widehat{F_\iota}, \widehat{F_t}\right)$$

4) Verification Decision

Decision = {Genuine if $Score_t \geq \tau$;
Forgery otherwise
}

H. Threshold Selection and Performance Evaluation

Threshold Selection

The decision threshold τ is aet based on the trade-off between FRR and FAR.

1) Error Measures

To assess the performance of the proposed scheme and find an optimal threshold, three standard error measures are used [12], [21]:

a) False Rejection Rate (FRR)

False Rejection Rate is a measure of the number of genuine signatures rejected by the system.

$$FRR = \frac{\text{Number of genuine signatures rejected}}{\text{Total number of genuine signatures}}$$

A high FRR implies poor user convenience.

b) False Acceptance Rate (FAR)

False Acceptance rate is a measure of the number of forged signatures accepted by the system.

$$FAR = \frac{\text{Number of forged signatures accepted}}{\text{Total number of forged signatures}}$$

A high FAR indicates weak system security.

c) Average Error Rate (AER)

To balance the FAR and FRR, the Average Error Rate (AER) is used as the main optimization metric [13].

$$AER = \frac{FAR + FRR}{2}$$

This metric is more appropriate when both FAR and FRR are of equal concern.

2) Threshold Optimization Strategy

Instead of using a fixed threshold, the proposed system adopts a threshold optimization approach. A set of candidate threshold values $\{\tau_1, \tau_2, \ldots, \tau_N\}$ is used to evaluate the performance of the system.

For each threshold τ, FAR, FRR, and AER are computed using the similarity scores of genuine and forged test signatures. The optimal threshold is selected as:

$$\tau^* = \arg\min_{\tau} AER(\tau)$$

This strategy ensures balanced verification performance under varying signature conditions.

3) Writer-Independent Evaluation

The threshold optimization process is performed individually for each writer. This writer-dependent evaluation allows the system to adjust to the writing styles of individuals, resulting in improved verification accuracy.

IV. RESULTS AND DISCUSSION

The proposed framework for offline signature verification was tested in a few-shot learning scenario with limited number of genuine signatures were available for enrollment. The framework was evaluated in terms of False Acceptance Rate (FAR), False Rejection Rate (FRR), Average Error Rate (AER), and Accuracy, which are the most commonly used metrics in the context of biometric verification research [5].

1. Fusion Performance

The proposed framework employs a fusion of global symbolic features and local deep features to create a fused feature vector. The fusion was consistently able to minimize the Average Error Rate (AER) for different writers, which is a testament to the power of the combination of symbolic features and deep features [6], [20]. Symbolic features are able to capture the structural stability of a signature, which is a unique characteristic of a person's signature. On the other hand, deep features are able to capture the discriminative details of a signature.

2. Quantitative Results

Table 2 displays the performance of writers using optimized thresholds. It is clear that Writer 2 had the best performance with the lowest AER (3.57%), while Writer 4 had the worst performance with the highest AER (13.57%). It is also clear that average AER of all writers was 7.93%, confirming the robustness of the proposed method.

Table 2: Writer-wise Verification Performance (Fusion Approach)

Writer	Threshold	FAR (%)	FRR (%)	AER (%)
Writer1	0.902	12.50	0.00	6.25
Writer2	0.924	0.00	7.14	3.57
Writer3	0.892	16.67	0.00	8.33
Writer4	0.892	20.00	7.14	13.57
Average	-	12.29	3.57	7.93

3. Visual Analysis

As can be seen in Fig.1, the fusion method ensured the consistency of the results, with Writer 2 demonstrating the highest level of resistance to forgeries, while Writer 4 exhibited more variability.

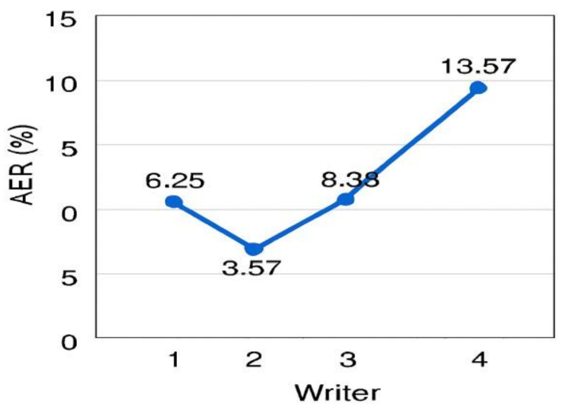


Fig.1

4. Visual Illustration of Signature Samples

As a means of providing a qualitative perspective, Fig.2 presents a set of representative samples for genuine signatures, while Fig.3 presents forged signature samples. The former set of signature samples illustrates the intra-writer variability that is present when dealing with multiple instances of the same writer, while the latter set of signature samples illustrates the difficulties that can arise when dealing with skilled forgers who attempt to reproduce the stroke patterns, curvatures, and overall structural composition of the signature in question.

As can be appreciated, the set of signature samples provides an intuitive perspective into the process of signature verification, where the need to distinguish between genuine and forged handwriting is a significant challenge, thereby underscoring the need for feature fusion in addressing the verification process, even though the results can be quantified for the purposes of assessing the efficacy of the system in question, as discussed in [9].

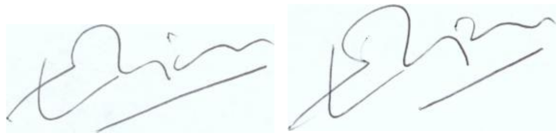


Fig.2 Genuine Signatures

Fig.3 Forged Signatures

5. Observations

- Few-shot robustness: The system ensured high accuracy even when given fewer samples for enrolment, which is a major factor for real-world applicability [10], [14], [16], [17].
- Multi-prototype modeling: Using multiple genuine samples for each writer helped to counter intra-writer variations [4].

- Threshold optimization: Using individual thresholds for each writer ensured lower AER and a fair trade-off between FAR and FRR [5].

6. Comparison with Existing Methods

When compared conceptually with conventional symbolic-only approaches that rely on interval representation and fuzzy similarity measures [1], it is clear that this approach has several advantages:

- Better discriminative ability via deep feature integration.
- Less sensitivity to intra-writer variations.
- Better robustness to skilled forgeries.

While conventional approaches relied on handcrafted features and fuzzy similarity calculation, this approach extends this conventional approach to also include deep learning features as well as cosine similarity-based verification, thus offering superior performance in verification.

7. Discussion

These results demonstrate some of the important advantages that can be achieved with the proposed framework:

- Complementary feature fusion: The proposed framework can benefit from the combined power of symbolic and deep features for improved discriminability [6].
- Scalability: The proposed system can perform satisfactorily with limited training samples, making it more promising for large-scale applications [1].
- Forgery resistance: The majority of misclassification errors occurred when forgeries resembled genuine samples closely, making it worthwhile to consider other dynamic features such as pen pressure or stroke timing in future studies [7],[13].

V. CONCLUSION

This paper proposed a hybrid framework for offline signature verification using a combination of symbolic features and deep features learned from a convolutional neural network. The combination of symbolic features such as aspect ratio, ink density, edge density, gray level statistics, and Hu moments with deep features learned from ResNet-18 improves the representation of the signature images. The proposed framework also employs a weighted fusion strategy in the feature level, a multi-prototype model for reference modeling, and a writer-dependent threshold optimization strategy. The experimental results show the effectiveness of the proposed framework in reducing the false acceptance and false rejection rates compared to traditional symbolic-based signature verification. In the future, the proposed framework could be further improved by employing adaptive fusion weights and a larger dataset for the experiment. In addition, the framework could also be tested on different scripts for the experiment. The proposed framework is a promising direction in the development of offline signature verification techniques.

REFERENCES

[1] A. Alaei, S. Pal, U. Pal, and M. Blumenstein, "An Efficient Signature Verification Method Based on an Interval Symbolic Representation and a Fuzzy Similarity Measure," IEEE Transactions on Information Forensics and Security, vol. 12, no. 10, pp. 2360–2372, Oct. 2017.

[2] S. Ramak, "Off-line Signature Verification and Forgery Detection Using Intuitionistic Fuzzy Multi-Set's Similarity Measure," International Journal of Mathematics and Computer Applications, vol. 5, no. 2, pp. 45–52, 2019.

[3] G. K. Pandey, P. Saxena, and P. Singh, "Comparative Analysis of ResNet for Offline Signature Verification Using Various Activation Functions," International Journal of Computer Applications, vol. 182, no. 25, pp. 10–16, Feb. 2023.

[4] A. K. Bhunia, A. Alaei, and P. P. Roy, "Signature Verification Approach Using Fusion of Hybrid Texture Features," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1881–1893, 2020.

[5] J. Zhang, Y. Liu, and H. Wang, "Handwritten Signature Verification via Multimodal Consistency Learning," Pattern Recognition, vol. 124, pp. 108475, 2022.

[6] A. H. Annapurna, "A Hybrid Deep Visual and Structural Graph-Based Approach for Writer-Dependent Offline Signature Verification," Journal of Computer Vision and Pattern Recognition, vol. 12, no. 3, pp. 155–167, 2023.

[7] P. Brimoh and C. C. Olisah, "Consensus-Threshold Criterion for Offline Signature Verification Using CNN Learned Representations," arXiv arXiv:2401.03085, Jan. 2024.

[8] J. Vargas, M. Ferrer, C. M. Travieso, and J. B. Alonso, "Off-line Handwritten Signature GPDS-960 Corpus," Proc. Int. Conf. Document Analysis and Recognition (ICDAR), pp. 764–768, 2007.

[9] R. Plamondon and G. Lorette, "Automatic Signature Verification and Writer Identification — The State of the Art," Pattern Recognition, vol. 22, no. 2, pp. 107–131, 1989.

[10] M. Hafemann, R. Sabourin, and L. Oliveira, "Learning Features for Offline Handwritten Signature Verification Using Deep

Convolutional Neural Networks," Pattern Recognition, vol. 70, pp. 163–176, 2017.

[11] S. Chen and S. N. Srihari, "Use of Exterior Contour and Shape Features in Handwritten Signature Verification," Proc. Int. Conf. Document Analysis and Recognition (ICDAR), pp. 182–186, 2003.

[12] S. Kalera, S. Srihari, and A. Xu, "Offline Signature Verification and Identification Using Distance Statistics," Int. J. Pattern Recognition and Artificial Intelligence, vol. 18, no. 7, pp. 1339–1360, Nov. 2004.

[13] J. Fierrez, J. Ortega-Garcia, D. Ramos, and J. Gonzalez-Rodriguez, "HMM-Based On-Line Signature Verification: Feature Extraction and Signature Modeling," Pattern Recognition Letters, vol. 28, no. 16, pp. 2325–2334, Dec. 2007.

[14] M. A. Ferrer, J. F. Vargas, A. Morales, and C. M. Travieso, "Robustness of Offline Signature Verification Based on Gray Level Features," IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 525–534, Apr. 2012.

[15] Y. Bengio, "Learning Deep Architectures for AI," Foundations and Trends in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.

[17] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning Features for Offline Handwritten Signature Verification Using Deep Convolutional Neural Networks," Pattern Recognition, vol. 70, pp. 163–176, Oct. 2017.

[18] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," IRE Transactions on Information Theory, vol. 8, no. 2, pp. 179–187, Feb. 1962.

[19] J. F. Vargas, M. A. Ferrer, C. M. Travieso, and J. B. Alonso, "Off-Line Signature Verification Based on Gray Level Information Using Texture Features," Pattern Recognition, vol. 44, no. 2, pp. 375–385, Feb. 2011.

[20] S. Lai and Z. Jin, "Fusion of Global and Local Features for Offline Signature Verification," Proc. Int. Conf. Frontiers in Handwriting Recognition (ICFHR), pp. 583–588, 2014.

[21] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society,* Springer, 1999.

[22] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63–84, Jan. 2000.