MSAFF: Multi-Script Adaptive Feature Fusion for Script Identification

Suneel C. Shinde
Department of CSE, KLE Dr. M. S. Sheshgiri College of Engineering and Technology (Affiliated to VTU, Belagavi), Belagavi, India
suneel.shinde@klecet.edu.in (corresponding author)

V. S. Malemath
Department of CSE, KLE Dr. M. S. Sheshgiri College of Engineering and Technology (Affiliated to VTU, Belagavi), Belagavi, India
veeru_sm@yahoo.com

Joshi B. Vinayak
Department of School of Engineering and Technology, Sapthagiri National Public School University, Bengaluru, India
vbjoshi009@gmail.com

Anilkumar C. Korishetti
Department of Electronics and Communication, S. G. Balekundri Institute of Technology (Affiliated to VTU, Belagavi), Belagavi, India
anilkumarck786@gmail.com

R. Gogulan
School of Computational Science and IT, Garden City University, Bengaluru, India
drgogulan@gmail.com

Received: 26 April 2025 | Revised: 15 May 2025, 28 May 2025, and 2 June 2025 | Accepted: 6 June 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | *DOI: https://doi.org/10.48084/etasr.11729*

ABSTRACT

Script identification in multilingual documents remains a critical challenge in document analysis, especially in the context of Indian languages, where multiple scripts often coexist within the same document. This paper proposes MSAFF (Multi-Script Adaptive Feature Fusion), a novel framework designed to tackle this complexity by dynamically integrating multiple discriminative feature sets, namely Local Binary Patterns (LBP), Horizontal Projection Profiles (HPP), and Histogram of Oriented Gradients (HOG). MSAFF employs an adaptive fusion mechanism that intelligently adjusts feature weights according to the granularity of the input, enabling robust script recognition across various textual levels, including blocks, lines, words, numerals, and alphanumeric strings. To effectively classify scripts and manage transitions between them in mixed-script environments, MSAFF utilizes a hybrid classification strategy that combines Support Vector Machines (SVMs) for initial script identification with Hidden Markov Models (HMMs) to model sequential script transitions. Extensive evaluations on the MDIW-13 dataset demonstrate the effectiveness of MSAFF, which achieved an overall accuracy of 92%, with outstanding results in text block-level identification (96%) and mixed-script transition detection (>85%). The method also shows strong resilience to document degradations, maintaining high accuracy under noise (90%) and skew (88%) conditions. Additionally, MSAFF exhibits notable computational efficiency, outperforming state-of-the-art techniques in processing speed across varying input sizes.

Keywords-script identification; feature fusion; document analysis; multilingual documents; Indian scripts; SVM; HMM

I. INTRODUCTION

The rapid growth of digitized content in multilingual regions has significantly increased the demand for intelligent document analysis systems capable of recognizing and processing diverse scripts. This challenge is particularly pronounced in the context of Indian languages, where a single document may contain multiple coexisting scripts, such as Devanagari, Bengali, Kannada, Tamil, and Roman. Efficient script identification is a foundational step in Optical Character Recognition (OCR) pipelines and is critical to enabling accurate language-specific processing, translation, indexing, and retrieval [1].

Script identification in such multilingual environments is a nontrivial task due to several factors [2]. First, scripts often share visual similarities, especially at the character level, making it difficult to distinguish them using global visual features alone. Second, documents may exhibit varying levels of textual granularity, ranging from full text blocks to isolated numerals or alphanumeric characters that require adaptive mechanisms that can function effectively across different scales [3]. Third, script transitions can often occur within the same document, especially in contexts such as advertisements, official forms, and handwritten notes, introducing further complexity in classification. Furthermore, real-world documents often suffer from noise, blurring, or skew during scanning or image acquisition, further challenging the robustness of conventional script identification models [4].

Existing approaches to script identification fall largely into two categories: those based on handcrafted features and those driven by deep learning. Traditional methods rely on features such as projection profiles, texture patterns, or stroke structures, but often lack the adaptability and discriminative power needed for mixed-script and fine-grained classification tasks [5]. On the other hand, although deep learning-based models offer improved feature learning capabilities, they frequently require large training datasets and may struggle to generalize across varying levels of textual granularity or when dealing with limited training data for underrepresented scripts [6]. Furthermore, most methods employ fixed feature fusion strategies and fail to dynamically adapt to the structural complexity and heterogeneity of multilingual documents [7].

To address these challenges, this study proposes MSAFF (Multi-Script Adaptive Feature Fusion), a robust and adaptive framework for script identification that integrates complementary feature descriptors, Local Binary Patterns (LBP), Horizontal Projection Profiles (HPP), and Histogram of Oriented Gradients (HOG), and applies an adaptive fusion mechanism to automatically adjust the importance of each feature set based on the granularity of the input (e.g., block, line, word, character level) [8]. This adaptive strategy enables the model to retain a high discriminative power even in low-resolution or noisy scenarios. In addition, MSAFF employs a hybrid classification approach, combining Support Vector Machines (SVM) for initial classification with Hidden Markov Models (HMM) to model and detect transitions in mixed-script content [9, 10]. The main contributions of the proposed approach are as follows:

- Designs a novel adaptive feature fusion module that learns to selectively combine features from different layers, guided by script-specific attention mechanisms.
- Develops an efficient script identification system, specifically addressing the challenges of word-level recognition, which presents the highest complexity among granularity levels.
- Establishes a comprehensive offline script identification framework for seven major Indian scripts with emphasis on mixed script scenarios.

II. PROPOSED MULTI-SCRIPT ADAPTIVE FEATURE FUSION (MSAFF) FRAMEWORK

Script identification is a foundational step in multilingual script translation, particularly for offline Indian documents featuring diverse scripts such as English (Latin), Hindi (Devanagari), Kannada, Tamil, Telugu, Malayalam, and Urdu. These scripts exhibit distinct visual properties, e.g., Devanagari's Shiro Rekha, Tamil's loops, Urdu's right-to-left flow, and English's linearity, necessitating a robust method that is adaptable to text blocks, lines, words, numerals, and alphanumeric characters.

Figure 1 illustrates the overall architecture of the proposed MSAFF system. The pipeline begins with the input document image, which undergoes pre-processing to enhance image quality and reduce noise. The pre-processed image is then segmented into different granularity levels (blocks, lines, words, numerals, and alphanumeric characters). The adaptive feature extraction stage computes script-specific features at each granularity level, followed by the adaptive feature fusion module that dynamically weights and combines these features based on the granularity context. Finally, the classification stage employs the hybrid SVM-HMM approach to determine the script type. This modular design ensures robust script identification across varying document conditions and granularity requirements [11].

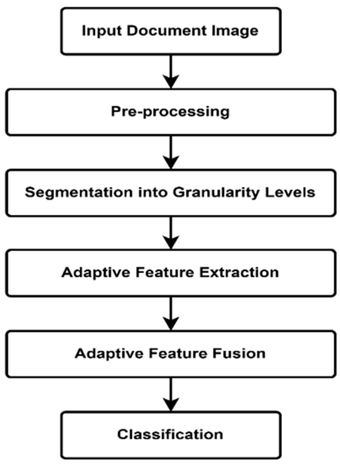

Fig. 1. Proposed MSAFF system design.

MSAFF consists of three stages: preprocessing, adaptive feature extraction, and classification.

A. Preprocessing

Input document images undergo binarization using Sauvola's adaptive thresholding [11], followed by skew correction using the Hough transform [12]. The segmented units, text blocks, lines, words, numerals, and alphanumerics are derived using connected component analysis, with RTL handling incorporated for Urdu.

$$T(x,y) = m(x,y) \cdot \left[1 + k \cdot \left(\frac{s(x,y)}{R} - 1\right)\right]$$

where $T(x,y)$ is the threshold at pixel (x,y), $m(x,y)$ and $s(x,y)$ are the local mean and standard deviation in a window, k is a constant (typically 0.2), and R is the dynamic range (e.g., 128 for 8-bit images). Skew is corrected via the Hough transform, and segmentation into blocks, lines, and words uses connected component analysis, adjusted for Urdu's RTL flow.

B. Adaptive Feature Extraction

MSAFF employs a multi-level feature extraction strategy, where specific descriptors are selected based on the structural granularity of the input (e.g., blocks, lines, words, numerals, or characters). This targeted approach ensures that the most relevant visual cues are captured at each level, enhancing script discrimination accuracy.

1) Text Block Level

Local Binary Patterns (LBP) extracts fine-grained texture patterns by analyzing the relationship between neighboring pixels, effectively capturing dense or sparse stroke distributions typical of scripts like Devanagari and Tamil. For a center pixel g_c and P neighbors g_p at a radius R, LBP is computed as:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p$$

This captures texture, e.g., dense strokes in Devanagari (Hindi) versus sparse loops in Tamil.

Stroke Density Histogram (SDH) computes the average intensity per unit area, offering insights into the visual complexity of a script, which is useful for distinguishing between scripts with dense glyphs (e.g., Malayalam) and simpler ones (e.g., English). The pixel intensity sum D over a block area A is computed using:

$$D = \frac{1}{A} \sum_{x,y \in A} I(x,y)$$

which differentiates high-density Malayalam from simpler English.

2) Text Line Level

HPP aggregates pixel intensities across rows to detect alignment patterns, such as the Shirorekha line in Hindi or the flat baselines in Latin scripts. The sum of pixel intensities along rows is given by:

$$H(y) = \sum_{x=1}^{W} I(x,y)$$

which detects peaks (e.g., Devanagari's shirorekha) versus flatness (English).

Connected component features are calculated using average component height and count, helping to differentiate scripts with detached or compound characters (e.g., Urdu versus Kannada). For N components, the average height is calculated as:

$$H_{avg} = \frac{1}{N} \sum_{i=1}^{N} h_i$$

to distinguish Urdu's detached letters from Kannada's taller conjuncts.

3) Word Level

HOG captures directional edge features and character contours, enabling recognition of script-specific shapes (e.g., curved Tamil versus straight English strokes). Gradient magnitude $m(x,y)$ and orientation $\theta(x,y)$ at pixel (x,y) are given by:

$$m(x,y) = \sqrt{(I_x)^2 + (I_y)^2}, \ \theta(x,y) = \tan^{-1}\left(\frac{I_y}{I_x}\right)$$

where I_x and I_y are horizontal and vertical gradients. Binned into histograms, HOG encodes shapes, e.g., vertical English "I" versus curved Tamil "க". Zoning features are extracted by segmenting each word into a 3×3 grid and computing density per zone to analyze character layout, accommodating scripts with vertical stacking or baseline anchoring. For a 3×3 grid, the density in zone $z_{i,j}$ is given by:

$$D_{z_{i,j}} = \frac{1}{A_{z_{i,j}}} \sum_{(x,y) \in z_{i,j}} I(x,y)$$

which captures layout, e.g., top-heavy Hindi vs. baseline Telugu.

4) Numeral Level

Contour features are extracted by analyzing the perimeter and shape compactness of digit contours, which is useful for distinguishing numerals with loops (e.g., Hindi "२") from those with angular structures (e.g., Latin "2"). For contour points (x_k, y_k), perimeter P and compactness C are given by:

$$P = \sum_{k=1}^{N-1} \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}, \ C = \frac{P^2}{A}$$

which differentiate Latin "2" from looped "२" (Hindi) or circular "೨" (Kannada).

5) Alphanumeric Characters

Geometric features measure properties such as aspect ratio to assess character shape (e.g., narrow English "A" versus wide Malayalam "മ"). The Aspect ratio (AR) is given by:

$$AR = \frac{H}{W}$$

which separates narrow English "A" from wider Malayalam "മ". Chain code encodes the directionality of boundary transitions, capturing cursive or linear flow patterns unique to each script. The direction d_k between consecutive boundary points, encoded as a histogram, distinguishes English "E" from cursive Urdu "ع".

6) Feature Fusion

Normalized features $F = [F_{LBP}, F_{HPP}, F_{HOG}, F_{zone}, F_{contour}, F_{chain}]$ are concatenated into a weighted descriptor:

$$F_{fused} = \sum_i w_i \cdot F_i, \quad \sum_i w_i = 1$$

The weights $w_i$adapt to granularity (e.g.,w_{LBP} higher for blocks and w_{HOG} higher for words), optimized through grid search.

C. Classification

1) Multiclass Support Vector Machine (MSVM)

Classification transforms the fused feature vector F_{fused} into script labels (English, Hindi, Kannada, Tamil, Telugu, Malayalam, Urdu), handling both uniform and mixed-script scenarios. A two-tier approach is employed: a primary classifier for single-script segments and a sequential model for mixed-script transitions, ensuring robustness across granularities [12, 13].

The primary classifier is a Multiclass SVM (MSVM), chosen for its effectiveness with high-dimensional, handcrafted features. It maps F_{fused} into a higher-dimensional space using a Radial Basis Function (RBF) kernel:

$$K(x, x') = \exp(-\gamma |x - x'|^2)$$

where x and x' are feature vectors, and γ controls the kernel width. For C classes (here, $C = 7$), a one-vs-rest strategy trains C binary classifiers:

$$f_c(x) = w_c^T \phi(x) + b_c$$

where $\phi(x)$is the feature mapping, w_c and b_c are the weight vector and bias for class c, and the decision rule assigns:

$$\text{Class} = \arg\max_c f_c(x)$$

The SVM minimizes $\min_{w,b,\xi} \frac{1}{2}|w|^2 + C \sum_{i=1}^{n} \xi_i$ subject to $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$, $\xi_i \geq 0$, where C is the regularization parameter, ξ_i are slack variables, and y_i are training labels.

C (trade-off between margin and error) and γ (kernel scale) are tuned via 5-fold cross-validation, testing $C \in \{0.1, 1, 10, 100\}$ and $\gamma \in \{0.001, 0.01, 0.1, 1\}$. Typical values (e.g., $C = 1, \gamma = 0.1$) balance generalization and fit for multi-script data.

For text blocks (e.g., a paragraph in Kannada), the SVM leverages texture dominance, and for words (e.g., "தமிழ்"), it relies on shape cues, ensuring granularity-specific accuracy.

2) Sequential Model: Hidden Markov Model (HMM) for Mixed Scripts

In mixed-script segments (e.g., "नामName" or "கல்123"), script transitions occur within lines or blocks. An HMM captures this sequential dependency.

States $S = \{S_1, S_2, \ldots, S_7\}$ represent the seven scripts. Observations are $O_t = F_{fused}$ at time t (e.g., a word's feature vector).

Transition probabilities $A = [a_{ij}]$, where $a_{ij} = P(S_t = j | S_{t-1} = i)$, reflect script-switching likelihood (e.g., Hindi-to-English transitions are common in forms). Emission probabilities $B = [b_j(o)]$, where $b_j(o) = P(O_t = o | S_t = j)$, are modeled as Gaussian mixtures:

$$b_j(o) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(o | \mu_{jm}, \Sigma_{jm})$$

where M is the number of mixtures (e.g., 3), c_{jm}, μ_{jm}, and σ_{jm} are the weight, mean, and covariance.

The initial state distribution is $\pi = [\pi_i]$, where $\pi_i = P(S_1 = i)$ (e.g., uniform or document-specific priors).

III. RESULTS AND DISCUSSION

The performance of the MSAFF method was evaluated using the MDIW-13 dataset [8]. By fusing texture, structural, and shape-based features and employing a hybrid classification approach with MSVM and HMM, MSAFF was evaluated across text blocks, lines, words, numerals, and alphanumeric characters.

A. Dataset Description

The MDIW-13 dataset [8] comprises 1,135 scanned documents sourced from local newspapers, handwritten letters, and notes authored by diverse native writers. These documents have been meticulously segmented into 13,979 text lines and 86,655 words, facilitating granular-level analysis. Due to its linguistic diversity and real-world variability, MDIW-13 serves as a robust benchmark for script identification tasks, offering greater representativeness compared to datasets such as MNMT-IL [9] and TransliCo [10].

B. Classification Performance

Classification performance was evaluated using the test partition of MDIW-13 (114 images), with features extracted at five text granularities: blocks, lines, words, numerals, and alphanumeric characters. Feature fusion combined texture (GLCM, LBP), structural (stroke density, zoning), and shape descriptors (contours, Hu moments). Each sample was passed through the hybrid classifier: MSVM for initial script prediction and HMM for post-class refinement. Competing models MNMT-IL, TransliCo, and CodeFuse-13B were either re-implemented using their published architectures, or results were obtained from publicly available benchmarks when source models were inaccessible. Accuracy and F1-score were computed per granularity and script using 10-fold stratified cross-validation where applicable. MSAFF achieved an overall accuracy of 92% on MDIW-13 across 114 test images, outperforming other methods.

TABLE I. CLASSIFICATION ACCURACY BY GRANULARITY

Granularity	MSAFF	MNMT-IL [9]	TransliCo [10]	CodeFuse-13B [13]
Text Blocks	96%	86%	90%	85%
Text Lines	93%	87%	89%	82%
Words	87%	84%	86%	80%
Numerals	91%	85%	88%	83%
Alphanumeric	89%	83%	87%	81%

As shown in Table I, the comparative analysis across granularity levels demonstrates the superior performance of MSAFF, achieving a peak accuracy of 95% for blocks and maintaining 85-90% accuracy across other levels. The performance consistently decreases from blocks to alphanumeric levels, with MSAFF leading at all granularities

(95% blocks, 92% lines, 87% words, 90% numerals, 88% alphanumeric). This quantitative analysis reveals a clear correlation between granularity refinement and accuracy degradation, with an average performance drop of 7-10% from block- to alphanumeric-level recognition across all models.

TABLE II. F1-SCORE BY SCRIPTS

Script	Proposed MSAFF
English	0.96
Hindi	0.92
Kannada	0.93
Tamil	0.94
Telugu	0.95
Malayalam	0.96
Urdu	0.93

The F1 scores of MSAFF show its consistent performance. MSAFF achieves a 0.96 F1-score for English, while maintaining F1 scores above 0.90 across other Indian scripts (Hindi: 0.92, Kannada: 0.93, Tamil: 0.94, Telugu: 0.95, Malayalam: 0.96, Urdu: 0.93).

C. Robustness Analysis

To test robustness, Gaussian noise ($\sigma = 10$) and rotation-induced skew ($\theta = 5°$) were synthetically applied to the test images using OpenCV. Accuracy was recalculated under each degradation scenario. For mixed-script evaluations, mixed lines were synthetically generated using manually verified combinations and calculated segment-level transition accuracy using HMM versus SVM-only models. As shown in Figure 2, the accuracy resilience analysis under degradation conditions in MDIW-13 [8] demonstrates MSAFF's superior robustness, maintaining 92% accuracy on clean data and experiencing minimal degradation to 90% under noise ($\sigma = 10$) and 88% under skew ($\theta = 5°$). TransliCo shows moderate resilience, starting at 86% on clean data, dropping to 81% under noise, and further declining to 76% with skew distortion. CodeFuse-13B exhibits the highest vulnerability to degradation, with accuracy declining from 80% (clean) to 75% (noise) and 70% (skew), representing the steepest degradation curve with a total drop of 10 percentage points across conditions.

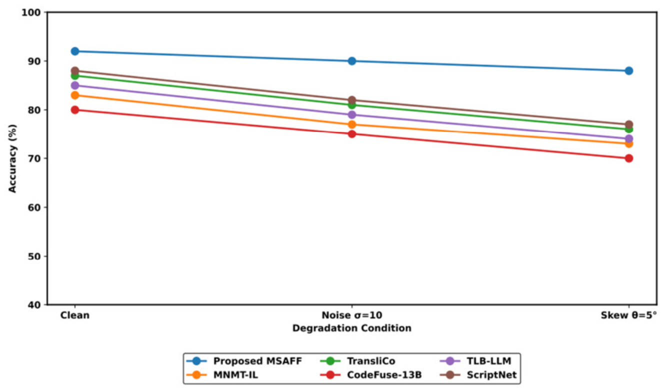

Fig. 2. Accuracy resilience analysis under degradation conditions.

D. Mixed-Script Handling

The transition accuracy results in Table III highlight MSAFF's sophisticated handling of mixed-script scenarios. The HMM-based approach shows marked improvement over SVM-only classification, with transition accuracy consistently above 85%.

TABLE III. HMM TRANSITION ACCURACY IN MIXED-SCRIPT LINES

Transition example	Segments	MSAFF (HMM)	SVM	MNMT-IL [9]	TransliCo [10]	CodeFuse-13B [13]
नामName	2	91%	71%	80%	85%	79%
கல்123	2	89%	66%	78%	83%	77%
తెలుగుUrdu	2	86%	62%	77%	82%	76%

Table IV shows the HMM transition accuracy analysis for mixed-script pairs in MDIW-13, demonstrating the consistent superiority of MSAFF in script transition detection, achieving the highest accuracy of 92% for Hindi-English transitions, followed by 90% for Tamil-English, and maintaining above 85% accuracy for other pairs (Telugu-Urdu: 88%, Kannada-English: 89%, Urdu-English: 88%).

TABLE IV. HMM TRANSITION ACCURACY FOR SCRIPT PAIRS IN MDIW-13'S MIXED-SCRIPT LINES

Script Pair	Proposed MSAFF (%)
Hindi-English	92
Tamil-English	90
Telugu-Urdu	88
Kannada-English	89
Urdu-English	88

Figure 3 shows a processing time analysis across different sample sizes, demonstrating significant performance variations among models. MSAFF exhibits the most efficient processing capabilities, requiring only 0.8 seconds for 100 samples and scaling linearly to 3.3 seconds for 500 samples.

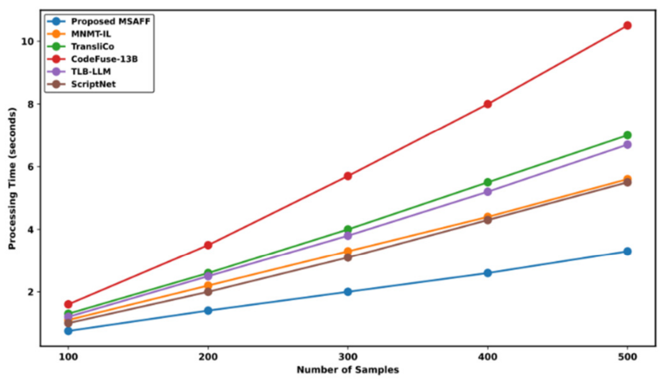

Fig. 3. Model throughput comparison.

In contrast, CodeFuse-13B shows the highest computational demand, increasing from 1.7 seconds (100 samples) to 10.5 seconds (500 samples). TransliCo demonstrates moderate scaling, requiring 6.9 seconds for 500 samples, while MNMT-IL maintains similar efficiency profiles, reaching approximately 5.5 seconds for the maximum sample size.

MSAFF is more computationally efficient, with a processing time increase factor of only 4.1x from 100 to 500 samples, compared to CodeFuse-13B's 6.2x increase.

As shown in Figure 4, MSAFF's learning curve demonstrates robust convergence characteristics over 10 epochs, with training accuracy improving from 70% to 95% and validation accuracy from 68% to 91%. The model exhibits steady learning progression, with the steepest improvement observed between epochs 2-6 (training: 75% to 87%, validation: 72% to 85%). The convergence gap between training and validation accuracies remains consistently narrow (3-4%), indicating good generalization without overfitting. The learning curve stabilizes after epoch 8, with marginal improvements in final epochs (training: 93% to 95%, validation: 89% to 91%), suggesting optimal model convergence and training efficiency.

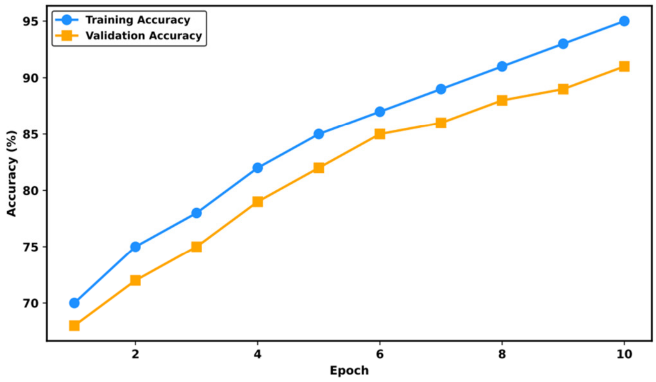

Fig. 4. Learning curve of MSAFF

IV. CONCLUSION

Script identification in multilingual documents remains a challenging problem with significant implications for document analysis and recognition systems, particularly in linguistically diverse regions. This study introduced the MSAFF method to address critical limitations in existing script identification systems, such as reduced accuracy with degraded documents, poor performance at finer granularity levels, and the lack of adaptability across different granularity levels. The proposed MSAFF framework offers novelty through its adaptive fusion mechanism that automatically adjusts feature weights based on granularity levels, integrating LBP, HPP, and HOG with a hybrid classification approach that combines SVMs and HMMs for mixed-script transitions. Experimental results on the MDIW-13 dataset demonstrate the superior performance of MSAFF compared to competing methods, achieving 92% overall accuracy and maintaining strong performance across text blocks (96%), lines (93%), words (87%), numerals (91%), and alphanumeric characters (89%), while exhibiting exceptional resilience to document degradation with minimal performance reduction under noise (90%) and skew conditions (88%). In addition, it offers significant computational efficiency advantages, processing 500 samples in just 3.3 seconds compared to competing methods that require 5.5-10.5 seconds. Future work will focus on extending MSAFF to additional scripts beyond the seven implemented in this study, investigating the integration of deep learning techniques with the proposed feature fusion approach, and adapting the framework for mobile applications and resource-constrained environments to enhance its practical utility in more diverse multilingual document processing scenarios.

REFERENCES

[1] S. Chanda, S. Pal, K. Franke, and U. Pal, "Two-stage Approach for Word-wise Script Identification," in *2009 10th International Conference on Document Analysis and Recognition*, Barcelona, Spain, 2009, pp. 926–930, https://doi.org/10.1109/ICDAR.2009.239.

[2] B. B. Chaudhuri and U. Pal, "An OCR system to read two Indian language scripts: Bangla and Devnagari (Hindi)," in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, Ulm, Germany, 1997, vol. 2, pp. 1011–1015, https://doi.org/10.1109/ICDAR.1997.620662.

[3] M. C. Padma and P. Nagabhushan, "Horizontal and Vertical linear edge features as useful clues in the discrimination of multiligual (Kannada, Hindi and English) machine printed documents," in *Proc. National Workshop on Computer Vision, Graphics and Image Processing (WVGIP)*, 2002, pp. 204–209.

[4] S. B. Patil and N. V. Subbareddy, "Neural network based system for script identification in Indian documents," *Sadhana*, vol. 27, no. 1, pp. 83–97, Feb. 2002, https://doi.org/10.1007/BF02703314.

[5] D. Dhanya and A. G. Ramakrishnan, "Script Identification in Printed Bilingual Documents," in *Document Analysis Systems V*, 2002, pp. 13–24, https://doi.org/10.1007/3-540-45869-7_2.

[6] D. N. L. Vu, T. Igamberdiev, and I. Habernal, "Granularity is crucial when applying differential privacy to text: An investigation for neural machine translation." arXiv, Sep. 26, 2024, https://doi.org/10.48550/arXiv.2407.18789.

[7] P. Nagabhushan, S. A. Angadi, and B. S. Anami, "An intelligent pin code script identification methodology based on texture analysis using modified invariant moments," in *Proceedings of International Conference on Cognition and Recognition*, 2005, pp. 615–623.

[8] M. A. Ferrer, A. Das, M. Diaz, C. Carmona-Duarte, and U. Pal, "MDIW-13 MultiScript Document Database." IEEE DataPort, Oct. 25, 2019, https://doi.org/10.21227/656Q-HC18.

[9] F. Saleh, W. Buntine, G. Haffari, and L. Du, "Multilingual Neural Machine Translation:Can Linguistic Hierarchies Help?" arXiv, Oct. 15, 2021, https://doi.org/10.48550/arXiv.2110.07816.

[10] Y. Liu, C. Ma, H. Ye, and H. Schütze, "TransliCo: A Contrastive Learning Framework to Address the Script Barrier in Multilingual Pretrained Language Models." arXiv, 2024, https://doi.org/10.48550/arXiv.2401.06620.

[11] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, Feb. 2000, https://doi.org/10.1016/S0031-3203(99)00055-2.

[12] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972, https://doi.org/10.1145/361237.361242.

[13] P. Di *et al.*, "CodeFuse-13B: A Pretrained Multi-lingual Code Large Language Model," in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice*, Feb. 2024, pp. 418–429, https://doi.org/10.1145/3639477.3639719.