

LANGEVIN RESONANCE TRANSFORMER FOR CROSS-SCRIPT WRITER IDENTIFICATION

SK GOLAM SAROWAR HOSSAIN¹ — MRIDUL GHOSH² — TONMOY METE³ —
MÁRIA ŽDÍMALOVÁ⁴ — KAUSHIK ROY⁵ — SK MD OBAIDULLAH¹

¹Department of Computer Science & Engineering, Aliah University, Kolkata, INDIA

²Department of Computer Science, Shyampur Siddheswari Mahavidyalaya, Howrah, INDIA

³Department of Computer Science, Asutosh College, Kolkata, INDIA

⁴Department of Mathematics and Descriptive Geometry, Slovak University of Technology
in Bratislava, Bratislava, SLOVAKIA

⁵Department of Computer Science, West Bengal State University, Kolkata, INDIA

ABSTRACT. Vision Transformers (ViTs) have emerged as a powerful architecture for various computer vision tasks, including writer identification. However, like their CNN counterparts, they are susceptible to performance degradation in cross-script scenarios because their standard self-attention mechanism learns script-specific visual correlations. To overcome this, we propose a novel architecture, the Langevin Resonance Transformer (LRT). The LRT fundamentally redefines self-attention by replacing the abstract mathematical operation with a physically-grounded dynamic simulation. Each image patch is treated as a particle. The core of the LRT is a novel Langevin Attention Layer, where the interaction between pairs of particles is governed by a learnable potential energy function. The net force on each particle is aggregated from all other particles, and its state is evolved according to the Langevin equation, which models motion under both deterministic and stochastic forces. The LRT treats a writer's style as a physical system defined by an energy landscape. Because it models biomechanics rather than just visual shapes, the resulting representation works regardless of the script. We tested this on the BRS-ID dataset using a custom augmentation strategy. The results show that LRT achieves higher accuracy than standard Vision Transformers and other top-tier models on the cross-script identification task.

© 2026 Mathematical Institute, Slovak Academy of Sciences.

2020 Mathematics Subject Classification: 68T20, 68W01, 68W99.

Keywords: writer identification, cross-script analysis, vision transformer, physics-informed machine learning, Langevin dynamics, self-attention, vision transformers.



Licensed under the Creative Commons BY-NC-ND 4.0 International Public License.

1. Introduction

Forensic science [11], historical analysis [6], and biometric security [9] are all supported by writer identification. Because handwriting is the result of neuromuscular activity, it incorporates consistent physical characteristics that are specific to each person, independent of the substance of the text. Current research is dominated by Convolutional Neural Networks (CNNs) [4] and Vision Transformers (ViTs) [8], however, they are not generalizable. These models perform poorly in cross-script settings but achieve good accuracy on single-script benchmarks [1]. The writer’s underlying motor skills are missed by standard architectures that overfit to particular visual graphemes [27]; the learned visual connections become invalid as the script changes.

Instead of focusing only on the appearance of handwriting, we suggest modeling its physics, we suggest modeling its physics. We present the Langevin Resonance Transformer (LRT). The image patches are viewed by this architecture as particles in a dynamic system that interact with one another through learnable forces. The biomechanical energy landscape of a writer is captured by the LRT through the simulation of this system. Due to the inherent robustness of this physical representation, the model can recognize writers even when the script is altered.

The main contributions of this work are as follows:

- (1) We introduce the Langevin Resonance Transformer (LRT). This architecture treats interactions between image patches as a dynamic physical system. It replaces standard abstract attention scores with a simulation based on computational stochastic dynamics.
- (2) Formulation of a physics-informed attention mechanism, the Langevin Attention Layer, where interactions are governed by a learnable potential energy function, and states are updated via a differentiable Langevin dynamics simulation.
- (3) An architectural design that yields inherently script-agnostic features, as it learns a biomechanical energy landscape from patch interactions rather than the morphology of specific graphemes, directly mitigating style-content entanglement.
- (4) Rigorous empirical validation of the LRT architecture’s superior generalization in cross-script writer identification, supported by a novel and comprehensive data augmentation strategy, where it significantly outperforms contemporary deep learning architectures.

The paper is organized as follows. Section 2 reviews the literature on architectural innovations and physics-informed learning. In Section 3, we define the proposed Langevin Resonance Transformer (LRT) and its attention mechanism.

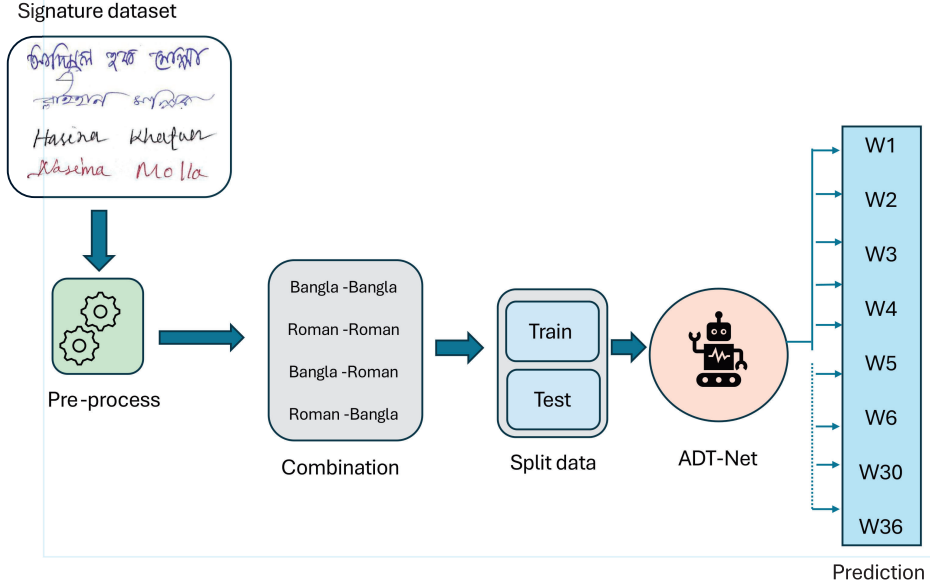


FIGURE 1. Flowchart for writer identification using handwritten signatures as input.

Section 4 outlines the experimental setup, introduces the new BRS-ID dataset, and details our data augmentation strategy. Section 5 analyzes the results. Finally, Sections 6 and 7 discuss computational complexity and conclude the study.

2. Related Work

Writer identification research is currently dominated by deep learning approaches. In order to frame the suggested LRT model, this section examines important architectural advancements and cross-script tactics.

2.1. Architectural Innovations in Deep Learning

Enhancing deep neural network designs is the main focus of current research.

- **Convolutional Neural Networks (CNNs):** CNNs [12,14] continue to be the industry standard. Small, automatically segmented patches are used in fragment-based techniques such as FragNet [4]. This method enables models to learn robust local features and manage different text lengths. Multi-scale feature extraction has been incorporated into further research to capture stroke patterns [2]. In order to weigh particular regions and

ignore uninformative background noise, many modern CNNs additionally include attention methods [26].

- **Sequence models and transformers:** Because they can simulate long-range dependencies, transformers are becoming more and more popular. Handwriting is treated as a patch sequence by systems such as HanDA [8] and other Vision Transformer (ViT) techniques [17]. Unlike CNNs, which have limited receptive fields, self-attention connects distant image parts to capture global traits such as slant and spacing. Recent 2024 research prioritizes efficiency, as seen in DocFormerV2 [3], or creates hybrid CNN-Transformer models [20]. The LRT extends this lineage, but replaces the standard attention mechanism.
- **Graph Neural Networks (GNNs):** Some researchers model handwriting as a graph of strokes using GNNs [21]. Here, key points become nodes and connections become edges. The model learns by aggregating neighbor information. Although GNNs explicitly model topology through message passing, the LRT differs by modeling interactions via a global physical simulation.

2.2. Advanced Training Paradigms

Researchers have also focused on training strategies to improve data efficiency and model robustness

- **Self-Supervised Learning (SSL):** To reduce the reliance on large labeled datasets, SSL has emerged as a powerful pre-training paradigm [28]. Models are first trained on a pretext task using vast amounts of unlabeled handwriting data (e.g., reconstructing masked patches). This process forces the model to learn a rich, general-purpose representation of handwriting style, which can then be fine-tuned for writer identification with much less labeled data.
- **Few-Shot and Zero-Shot Learning:** In many forensic scenarios, only a very small number of handwriting samples are available per writer. This has spurred research into few-shot learning, often using Siamese networks or other metric learning approaches to learn a feature space where samples from the same writer are close and samples from different writers are far apart [24]. More ambitiously, zero-shot writer identification attempts to recognize writers not seen during training, often by learning a mapping to a semantic space [23].

2.3. Physics-Informed Machine Learning

The concept of integrating physical laws into neural networks is a rapidly growing field, with a recent comprehensive survey provided by [18]. While most

Physics-Informed Neural Networks (PINNs) use loss functions to enforce compliance with PDEs, our approach is more aligned with the emerging field of physics-informed architectures, which embed physical principles directly into the model’s structure. As a forward-looking survey from 2025 suggests [5], such architectures are poised to be the next frontier in creating robust and generalizable AI. The LRT is a novel contribution to this emerging class of models.

2.4. The Cross-Script Challenge

The cross-script problem remains the litmus test for a truly robust writer identification system. Recent approaches have attempted to tackle this by explicitly learning script-invariant representations.

- **Feature Disentanglement:** This popular strategy employs adversarial learning to separate a feature vector into two components: a writer-specific style representation and a script-specific content representation [27]. The goal is for the style component to be useful for identification regardless of the script. A 2024 study by [7] introduced a novel contrastive learning objective to explicitly pull representations of the same writer in different scripts closer together in the embedding space. However, these methods still operate on visual features, where style and content are inherently entangled.
- **Domain Adaptation:** Unsupervised Domain Adaptation (UDA) techniques aim to align the feature distributions of different script domains [19]. This is often achieved by adding a domain discriminator that forces the main feature extractor to produce script-agnostic features.

Although these learning-based methods have advanced the state-of-the-art, they are often complex and depend on the successful separation of style and content, which are deeply entangled in the visual domain. Our LRT proposes a more foundational solution by changing the nature of the representation itself to be physics-based.

3. Proposed Method: Langevin Resonance Transformer (LRT)

The Langevin Resonance Transformer (LRT) is a novel architecture designed to learn script-agnostic representations of writer style by integrating the principles of stochastic dynamics into the core of a Vision Transformer. Instead of learning abstract visual correlations, the LRT models the interactions between image patches as a physical system of interacting particles governed by learnable forces. This section provides a detailed mathematical and algorithmic formulation of the LRT.

3.1. Architectural Overview

The LRT maintains the high-level structure of a standard Vision Transformer [10], consisting of a patch embedding module followed by a series of identical encoder blocks and a final classification head. The key innovation lies within the encoder block, where the standard Multi-Head Self-Attention (MHSA) mechanism is replaced by our proposed Multi-Head Langevin Attention (MHLA) layer. Each encoder block thus consists of an MHLA layer followed by a standard position-wise MLP, with Layer Normalization and residual connections applied to both sub-layers.

3.2. Image to Particle State Embedding

The initial stage of the LRT transforms the input image into a sequence of particle states.

1. **Patching and Projection:** An input image $I \in \mathbb{R}^{H \times W \times C_{in}}$ is unrolled into a sequence of $N = HW/P^2$ flattened patches $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C_{in})}$, where P is the patch size. These patches are then mapped to a latent dimension D via a learnable linear projection matrix $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C_{in}) \times D}$.

$$\mathbf{Z}' = [\mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}]. \quad (1)$$

2. **Tokenization and Positional Encoding:** A special ‘[CLS]’ token, represented by a learnable vector $\mathbf{x}_{cls} \in \mathbb{R}^D$, is prepended to the sequence. Finally, learnable positional embeddings $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$ are added to inject spatial information. The resulting sequence is $\mathbf{Z}_0 \in \mathbb{R}^{(N+1) \times D}$.

$$\mathbf{Z}_0 = [\mathbf{x}_{cls}; \mathbf{Z}'] + \mathbf{E}_{pos}. \quad (2)$$

3. **Particle State Initialization:** We treat every vector $\mathbf{z}_i \in \mathbf{Z}_0$ as a distinct particle. To track its evolution, we define a state tuple consisting of position $\mathbf{x}_i$ and velocity $\mathbf{v}_i$. At the layer’s entry point, we set the initial position $\mathbf{x}_i$ to match the input feature vector, while the system starts at rest (velocity $\mathbf{v}_i = \mathbf{0}$).

$$\mathbf{x}_i^{(0)} = \mathbf{z}_i \in \mathbb{R}^D!, \quad \mathbf{v}_i^{(0)} = \mathbf{0} \in \mathbb{R}^D. \quad (3)$$

3.3. The Multi-Head Langevin Attention Layer

The MHLA layer is the core innovation of LRT. It replaces the abstract similarity calculation of self-attention with a multi-step, N-body physics simulation. The “multi-head” design, crucial for Transformers, is re-interpreted in a physical context: we run H independent physical simulations in parallel, each operating in a different feature subspace, allowing the model to learn different types of physical interactions (e.g., short-range vs. long-range, attractive vs. repulsive) simultaneously.

LANGEVIN RESONANCE TRANSFORMER

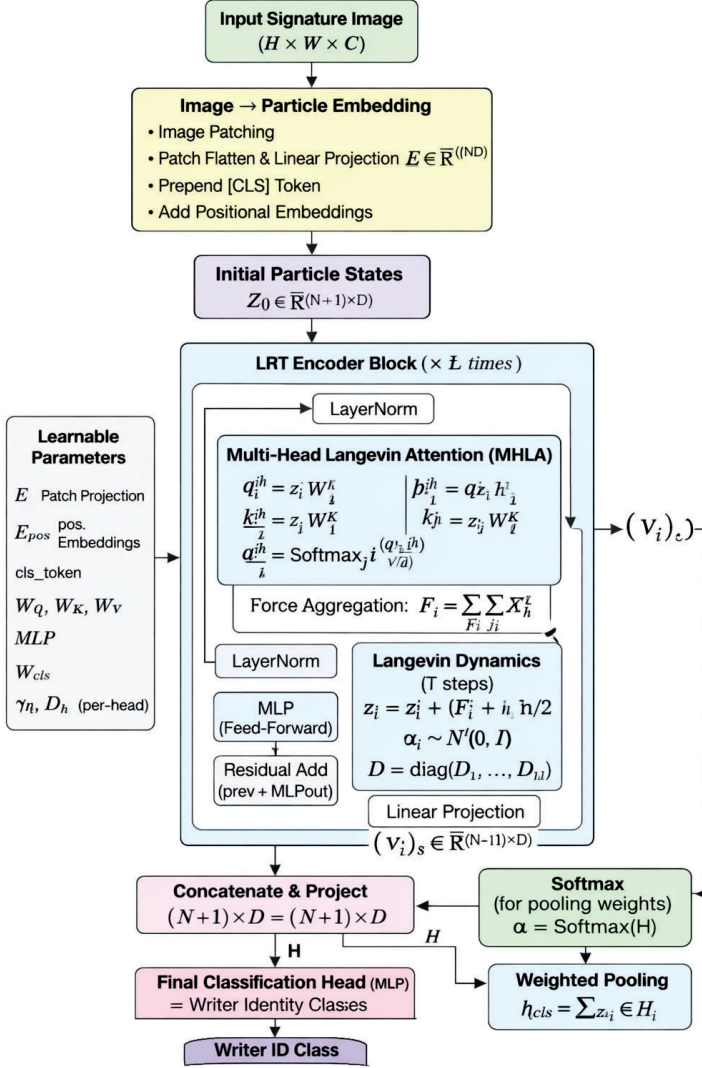


FIGURE 2. Workflow of the Langevin Resonance Transformer (LRT).
 (1) *Image $\rightarrow$ Particle Embedding*: Input images $(H \times W \times C)$ are tokenized into patches and combined with positional embeddings to form the initial particle system Z_0 .
 (2) *Dynamics Simulation*: The sequence passes through L encoder layers where the **Multi-Head Langevin Attention (MHSA)** module simulates particle trajectories. This module computes interaction forces and updates particle states via differential Langevin equations over T steps.
 (3) *Classification*: The global stylistic representation is extracted and mapped to the specific Writer ID class. Key learnable physical parameters (e.g., γ, D) are listed on the left.

For an input sequence of particle positions $\{\mathbf{x}_i^{(0)}\}_{i=1}^{N+1}$, the MHLA layer performs the following steps for each head $h \in \{1, \dots, H\}$:

3.3.1. Physics-Informed Projections per Head

The D -dimensional state of each particle is first projected into a lower-dimensional subspace for each head. For each head h , we have unique learnable weight matrices

$$\mathbf{W}_Q^h, \mathbf{W}_K^h, \mathbf{W}_V^h \in \mathbb{R}^{D \times D_k},$$

where $D_k = D/H$.

$$\mathbf{q}_i^h = \mathbf{x}_i^{(0)} \mathbf{W}_Q^h, \quad \mathbf{k}_j^h = \mathbf{x}_j^{(0)} \mathbf{W}_K^h, \quad \mathbf{v}_j^{\text{val},h} = \mathbf{x}_j^{(0)} \mathbf{W}_V^h. \quad (4)$$

In our physical analogy,

$$\mathbf{q}_i^h \quad \text{and} \quad \mathbf{k}_j^h \in \mathbb{R}^{D_k}$$

are vectors used to parameterize the forces between particles, while $\mathbf{v}_j^{\text{val},h} \in \mathbb{R}^{D_k}$ represents the “content” or “charge” of particle j that will be aggregated.

3.3.2. Learnable Potential Energy and Force Calculation

The interaction force between any two particles i and j is derived from a learnable potential energy function $V(\mathbf{q}_i^h, \mathbf{k}_j^h)$. We model this as a flexible Gaussian potential to allow for smooth, non-linear interactions.

$$V_{ij}^h = -a_{ij}^h \exp \left(-\frac{\|\mathbf{q}_i^h - \mathbf{k}_j^h\|^2}{2(\sigma_{ij}^h)^2} \right). \quad (5)$$

The key to the model’s expressiveness is that the potential is not fixed. The amplitude a_{ij}^h (interaction strength) and width σ_{ij}^h (interaction range) are dynamically predicted for each pair of particles by a small, head-specific MLP, MLP_θ^h , which takes the concatenated query and key as input:

$$[a_{ij}^h, \sigma_{ij}^h] = \text{softplus} \left(\text{MLP}_\theta^h([\mathbf{q}_i^h, \mathbf{k}_j^h]) \right). \quad (6)$$

The force vector is the negative gradient of this potential with respect to the query particle’s query vector:

$$\mathbf{F}_{ij}^h = -\nabla_{\mathbf{q}_i^h} V_{ij}^h = -a_{ij}^h \exp(\dots) \frac{(\mathbf{q}_i^h - \mathbf{k}_j^h)}{(\sigma_{ij}^h)^2}. \quad (7)$$

This resulting vector $\mathbf{F}_{ij}^h \in \mathbb{R}^{D_k}$ represents the influence (direction and magnitude) that particle j exerts on particle i within head h .

3.3.3. Differentiable Langevin Dynamics Simulation

The evolution of the system is governed by the Langevin equation. We use the Euler-Maruyama method, a differentiable numerical integrator, to simulate the trajectory of each particle for T discrete time steps, each of size Δt .

The total deterministic force on the particle i in the head h at the time step t is an aggregation of forces from all other particles, modulated by their value vectors. This is the direct physical analogue of the weighted sum in standard self-attention.

$$\mathbf{F}_i^{h,(t)} = \sum_{j=1}^{N+1} \mathbf{F}_{ij}^{h,(t)} \odot \mathbf{v}_j^{\text{val},h}, \quad (8)$$

where $\odot$ denotes element-wise multiplication. We then update the state of each particle within each head, $(\mathbf{x}_i^h, \mathbf{v}_i^h)$, according to the discretized Langevin equation. The learnable parameters are the friction coefficient γ_h and the noise intensity D_h for each head. For each simulation step from $t = 0$ to $T - 1$:

$$\begin{aligned} \mathbf{v}_i^{h,(t+1)} &= \mathbf{v}_i^{h,(t)} + \left(\mathbf{F}_i^{h,(t)} - \gamma_h \mathbf{v}_i^{h,(t)} \right) \Delta t \\ &\quad + \sqrt{2D_h \Delta t} \mathbf{R}^{(t)}, \end{aligned} \quad (9)$$

$$\mathbf{x}_i^{h,(t+1)} = \mathbf{x}_i^{h,(t)} + \mathbf{v}_i^{h,(t+1)} \Delta t, \quad (10)$$

where $\mathbf{R}^{(t)} \sim \mathcal{N}(0, I)$ is a standard Gaussian noise vector.

3.3.4. Multi-Head Aggregation and Output

After T simulation steps, the final positions of the particles in each head, $\mathbf{x}_i^{h,(T)}$, represent the evolved state of the particles. These are concatenated and projected back to the original dimension D with a final linear layer

$$\mathbf{W}_O \in \mathbb{R}^{D \times D}.$$

$$\text{MHSA}(\mathbf{Z}_0)_i = \text{Concat} \left(\mathbf{x}_i^{1,(T)}, \dots, \mathbf{x}_i^{H,(T)} \right) \mathbf{W}_O. \quad (11)$$

This entire process is summarized in Algorithm 2.

3.4. LRT Encoder Block and Final Classification

A full LRT encoder block consists of the MHSA layer followed by Layer Normalization, a standard MLP (Feed Forward Network), and another Layer Normalization. Residual connections are applied around both sub-layers, identical to a standard Transformer. The output of the final LRT encoder layer, specifically the state of the ‘[CLS]’ token, $\mathbf{x}_{\text{cls}}^{(L)}$, is taken as the final signature representation and is passed to a simple MLP head for classification. The complete forward pass is described in Algorithm 1.

Algorithm 1 LRT Forward Pass

```

1: Input: Image  $I$ , Number of layers  $L$ 
2: Output: Writer class probabilities
3:  $\mathbf{Z}_0 \leftarrow \text{PatchAndEmbed}(I)$  ▷ ▷ See Sec. 3.2, Eq. 1–2
4: for  $l$  from 1 to  $L$  do
5:    $\mathbf{Z}' \leftarrow \text{MHLE}(\text{LayerNorm}(\mathbf{Z}_{l-1}))$ 
6:    $\mathbf{Z}_{l-1} \leftarrow \mathbf{Z}_{l-1} + \mathbf{Z}'$  ▷ Residual connection
7:    $\mathbf{Z}'' \leftarrow \text{MLP}(\text{LayerNorm}(\mathbf{Z}_{l-1}))$ 
8:    $\mathbf{Z}_l \leftarrow \mathbf{Z}_{l-1} + \mathbf{Z}''$  ▷ Residual connection
9: end for
10:  $\mathbf{x}_{cls} \leftarrow \text{LayerNorm}(\mathbf{Z}_L)[0]$ 
11:  $P(\text{writer}) \leftarrow \text{ClassificationHead}(\mathbf{x}_{cls})$  ▷ ▷ See Sec. 3.4
12: return  $P(\text{writer})$ 

```

Algorithm 2 Multi-Head Langevin Attention (MHLE)

```

1: Input: Input sequence  $\mathbf{Z}_{in} \in \mathbb{R}^{(N+1) \times D}$ 
2: Output: Output sequence  $\mathbf{Z}_{out} \in \mathbb{R}^{(N+1) \times D}$ 
3: for all heads  $h = 1 \dots H$  in parallel do
4:   Project  $\mathbf{Z}_{in}$  to  $\mathbf{Q}^h, \mathbf{K}^h, \mathbf{V}^h$ 
5:   Initialize particle states:  $\mathbf{x}_i^{h,(0)} \leftarrow \mathbf{z}_i, \mathbf{v}_i^{h,(0)} \leftarrow \mathbf{0}$  ▷ ▷ Where  $\mathbf{z}_i \in \mathbf{Z}_{in}$ 
   (Eq. 3)  $\mathbf{Z}_{in}[i], \mathbf{v}_i^{h,(0)} \leftarrow \mathbf{0}$ 
6:   for  $t$  from 0 to  $T - 1$  do
7:     for all particles  $i = 1 \dots N + 1$  do
8:        $\mathbf{F}_i^{h,(t)} \leftarrow \mathbf{0}$ 
9:       for all particles  $j = 1 \dots N + 1$  do
10:        Compute force  $\mathbf{F}_{ij}^{h,(t)}$ 
11:         $\mathbf{F}_i^{h,(t)} \leftarrow \mathbf{F}_i^{h,(t)} + \mathbf{F}_{ij}^{h,(t)} \odot \mathbf{v}_j^{val,h}$ 
12:      end for
13:    end for
14:    for all particles  $i = 1 \dots N + 1$  do
15:      Update  $\mathbf{v}_i^{h,(t+1)}, \mathbf{x}_i^{h,(t+1)}$  using Eqs. 9-10
16:    end for
17:  end for
18:  Collect final positions  $\mathbf{Head}_h = \{\mathbf{x}_i^{h,(T)}\}_{i=1}^{N+1}$ 
19: end for
20:  $\mathbf{Z}_{concat} \leftarrow \text{Concat}(\mathbf{Head}_1, \dots, \mathbf{Head}_H)$ 
21:  $\mathbf{Z}_{out} \leftarrow \mathbf{Z}_{concat} \mathbf{W}_O$ 
22: return  $\mathbf{Z}_{out}$ 

```

4. Experimental Evaluation

We validate the LRT architecture using the BRS-ID dataset, focusing on its ability to generalize between scripts. This section describes the data properties, the four testing scenarios, and the specific metrics used for the assessment.

4.1. Dataset

We utilize the **Bangla-Roman Signature Identification (BRS-ID) dataset**¹. This collection contains 3,744 signature images obtained from 36 distinct writers. Each subject contributes 52 samples in Bangla and 52 in Roman script, all resized to 224×224 pixels. The dataset presents a significant challenge due to the structural difference between the cursive Bangla abugida and the disjointed Roman alphabet; to succeed, models must ignore script-specific graphemes and instead capture the writer’s biomechanical habits.

4.2. Evaluation Protocol

We assess performance using a one-vs-all multi-class classification framework across four specific data splits.

4.2.1. Evaluation Scenarios

To measure both baseline accuracy and cross-domain transfer, we define the following training and testing configurations:

1. **Same-Script (Bangla):** Training and evaluation are performed exclusively on the Bangla subset.
2. **Same-Script (Roman):** Training and evaluation are performed exclusively on the Roman subset.
3. **Cross-Script (Bangla $\rightarrow$ Roman):** We train the model on all available Bangla signatures and test it on the Roman signatures.
4. **Cross-Script (Roman $\rightarrow$ Bangla):** We train the model on all available Roman signatures and test it on the Bangla signatures.

4.2.2. Evaluation Metrics

We report performance using four standard classification metrics [13, 15]. These are derived from the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) calculated for each of the 36 writer classes. We report the macro-average for Precision, Recall, and F1-Score.

¹Dataset available at: <https://github.com/mridulxyz/BRS-ID>

- **Accuracy:** The global ratio of correctly classified signatures to the total number of samples.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (12)$$

- **Precision:** The ratio of correct positive predictions to the total positive predictions made by the model.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (13)$$

- **Recall (Sensitivity):** The ratio of correctly identified positive instances to the total actual positive instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (14)$$

- **F1-Score:** The harmonic mean of Precision and Recall, providing a single metric that accounts for both false positives and false negatives.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (15)$$

4.3. Implementation Details and Baselines

We implemented all models in PyTorch using an NVIDIA RTX 5000 GPU. We compare the LRT against the following established deep learning architectures:

- ResNet50 [16] and EfficientNet-B0 [25]: Two leading CNN architectures.
- SWIN Transformer [22]: A state-of-the-art hierarchical Vision Transformer.
- Standard Vision Transformer (ViT) [10]: A standard ViT model with the same dimensions as the LRT, serving as a direct architectural baseline.

Key hyperparameters for the primary LRT model and all baselines are summarized in Table 1.

4.4. Data Augmentation Strategy

Training high-capacity Transformer models on a dataset with a limited number of classes (36 writers) poses a significant risk of overfitting. To address this and to explicitly encourage the learning of robust, relational features that define a writer’s intrinsic style, we developed a comprehensive two-tiered data augmentation strategy applied on-the-fly during training.

4.4.1. Tier 1: Geometric and Photometric Augmentations

This tier consists of a rich set of augmentations designed to make the model invariant to common variations in image acquisition and handwriting execution.

LANGEVIN RESONANCE TRANSFORMER

TABLE 1. Hyperparameter settings for the proposed Langevin Resonance Transformer (LRT) and training configuration used for all baselines.

Parameter	Value
<i>Architectural Configuration (LRT)</i>	
Input Resolution	224×224
Patch Size (P)	16×16
Embedding Dimension (D)	768
Attention Heads (H)	12
Encoder Layers (L)	12
Simulation Steps (T)	5
Potential MLP Hidden Dim (D_h)	64
<i>Training Configuration</i>	
Optimizer	AdamW
Learning Rate	1×10^{-4}
Weight Decay	0.05
Batch Size	32
Total Epochs	100
Loss Function	Cross-Entropy
Dropout	0.1
Hardware	NVIDIA RTX 5000 (16GB)

- **Random Affine Transformations:** Combining rotation ($[-10, +10]$ degrees), scaling ($[0.85, 1.15]$ factor), and translation (10% of image dimensions).
- **Random Shear:** Applying a shear transformation of up to 10 degrees on both axes.
- **Elastic Distortion:** Applying non-rigid deformations to the image.
- **Random Perspective:** Simulating the effect of viewing the signature from a slight angle.
- **Photometric Jitter:** Randomly adjusting brightness, contrast, and saturation.
- **Stroke Width Morphing:** Applying random morphological erosion or dilation.
- **Gaussian Blur:** Applying a slight blur with a random kernel size.

4.4.2. Tier 2: Advanced Structural Augmentation: StrokeMix

To further force the LRT to learn the global, relational properties of a writer’s style, we introduce a novel augmentation technique called **StrokeMix**. Inspired by CutMix, StrokeMix transplants meaningful stroke segments, rather than random rectangular patches. The procedure is as follows:

- (1) Two training images, I_A (writer A) and I_B (writer B), are randomly selected.
- (2) Image I_B is binarized and skeletonized to identify its constituent stroke paths.
- (3) A random stroke segment is selected from the skeleton of I_B , and a tight bounding box is computed around it.
- (4) The corresponding patch from the original image I_B within this bounding box is extracted.
- (5) This stroke patch from writer B is then pasted onto image I_A at a random location.
- (6) The label for this newly created image remains that of writer A.

StrokeMix introduces difficult training samples to penalize reliance on local features. To succeed, the model must treat the transplanted stroke as noise and rely on global physical interactions between patches. This requirement aligns directly with the capabilities of the Langevin Attention mechanism.

An important limitation to keep in mind is that StrokeMix relies heavily on binarization and skeletonization in Step 2, which makes it somewhat sensitive to image quality. If applied to highly noisy, low-contrast, or poorly scanned documents, the skeletonization step can fail. When this happens, the algorithm might accidentally extract and transplant random background artifacts instead of true handwriting strokes. Fortunately, the BRS-ID dataset used in our experiments contains fairly clean, high-contrast signatures, keeping these failures to a minimum. Interestingly, though, we found that when the algorithm *does* accidentally injects background noise, it actually acts as an extra regularizer for the LRT. Because the Langevin Attention mechanism focuses on global physical interactions rather than just matching visual shapes, it quickly learns to treat these noisy, isolated patches as uncorrelated particles. Instead of getting confused by the artifact, the model simply assigns it a high learned noise intensity (D_h) or pushes it away via repulsive forces. This forces the network to ignore the distraction and focus even more heavily on the writer’s true biomechanical flow. That said, if StrokeMix is applied to highly degraded datasets in the future such as historical manuscripts—the standard binarization step will likely need to be replaced with a more robust stroke-extraction method.

5. Results and Analysis

We begin with a sensitivity analysis of hyperparameters and architectural components, followed by ablation studies. The section concludes by comparing the LRT against state-of-the-art baselines.

5.1. Impact of Train/Test Ratio

We trained the LRT on varying stratified splits of the Roman dataset to evaluate data efficiency. Table 2 summarizes performance metrics.

TABLE 2. LRT Performance vs. Train/Test Ratio (Roman-to-Roman).
 Peak performance occurred at the 70/30 split.

Train/Test Ratio	Accuracy	Precision	Recall	F1-Score
60/40	98.5	98.6	98.5	98.5
70/30	99.1	99.1	99.1	99.1
80/20	98.7	99.7	99.7	99.7
90/10	98.4	98.4	98.4	98.4

The 70/30 split produced the best generalization (Table 2). At 60/40, accuracy dropped to 98.5%, suggesting that the training set offered insufficient variance to fully resolve the potential energy parameters (σ_{ij}, a_{ij}) required by the Langevin simulation.

Increasing the training volume further degraded the performance. Accuracy decreased to 98.7% (80/20) and 98.4% (90/10), indicating overfitting. With excessive training data, the Langevin layers appear to memorize static artifacts rather than learning dynamic laws. Additionally, the 10% test set is statistically insufficient to validate the energy landscape. We selected the 70/30 split for all subsequent experiments to ensure a stable physical solution and robust evaluation.

5.2. Impact of Training Duration (Epochs)

We tracked convergence metrics during the Cross-Script (Roman $\rightarrow$ Bangla) experiment to evaluate stability. Table 3 lists the performance at specific intervals.

Convergence is rapid; the model passes 92% accuracy within 50 epochs. Performance remains stable between 100 and 200 epochs with minimal fluctuation.

We attribute this stability to Langevin dynamics, which appear to function as a regularizer against the overfitting typically seen in Transformers on limited datasets.

TABLE 3. LRT Performance vs. Training Epochs (Cross R→B).

Epochs	Accuracy	Precision	Recall	F1-Score
50	92.1	92.2	92.1	92.1
100	93.9	94.0	93.9	93.9
150	93.1	93.2	93.1	93.1
200	92.8	92.8	92.8	92.8

5.3. Impact of Learning Rate

Optimizing the LRT requires precise learning rate tuning due to the coupling between the learnable potential functions and the SDE solver. We evaluated training stability across three orders of magnitude using the Cross-Script task, as detailed in Table 4.

TABLE 4. LRT Performance vs. Learning Rate (Cross B→R Acc. %).

Learning Rate	Accuracy (%)	Observation
1×10^{-3} (High)	85.2	Unstable optimization
1×10^{-4} (Optimal)	93.9	Stable convergence
1×10^{-5} (Low)	92.1	Slow convergence

A high learning rate (1×10^{-3}) caused significant instability, limiting the accuracy to 85.2%. This likely stems from the Euler-Maruyama integration; large updates cause the force parameters (Equation 7) to fluctuate, preventing the system from reaching an energy minimum.

Conversely, a low rate (1×10^{-5}) stabilized training but slowed convergence, resulting in 92.1% accuracy. The model struggled to exit local minima within the epoch limit.

We found the optimal balance at 1×10^{-4} . This rate allowed the model to learn the potential energy parameters (a_{ij}, σ_{ij}) without numerical divergence. We utilized this rate for all subsequent tests.

5.4. Ablation Studies

5.4.1. Impact of Simulation Steps (T)

We investigated the effect of the simulation duration T within the Langevin Attention Layer.

Table 5 shows that a single step ($T = 1$) yields poor results. Performance improves as T increases to 5, confirming that multi-step evolution is necessary to capture the writer’s style dynamics.

TABLE 5. LRT Performance vs. Simulation Steps T (Cross R→B).

Simulation Steps (T)	Accuracy %
1 (No real dynamics)	87.9
3	93.2
5 (Optimal)	93.9
8	93.1

5.4.2. Role of Langevin Attention

To verify the contribution of our physics-informed mechanism, we replaced the Langevin Attention Layer with standard dot-product attention while keeping all other parameters constant.

TABLE 6. Ablation of Langevin Attention (Cross R→B Acc. %).

Attention Mechanism	Accuracy %
Standard Dot-Product Self-Attention	74.2
Langevin Attention (Ours)	93.9

Table 6 illustrates the layer’s importance. Reverting the Langevin mechanism to standard attention drops accuracy by over 20 percentage points, matching the baseline ViT. This sharp decline isolates the physics-based simulation as the primary driver of cross-script generalization.

5.5. Qualitative Analysis

The LRT also aids interpretability. We observe that the learned physical parameters (γ_h, D_h) and potential functions (a_{ij}, σ_{ij}) form writer-specific signatures. Notably, writers with unstable handwriting consistently induce higher learned noise intensities (D). The model therefore appears to approximate biomechanical properties rather than relying on arbitrary visual features.

5.6. Comparison with State-of-the-Art Methods

Table 7 benchmarks the LRT against prominent deep learning architectures.

The results highlight specific performance patterns based on script directionality:

Within-Script Performance:

For **Same-B** and **Same-R** tasks, all architectures achieve near-perfect accuracy ($> 99\%$). Standard models clearly suffice for single-script identification. The LRT matches the Standard ViT here, indicating that the inclusion of physical dynamics imposes no penalty on standard tasks.

TABLE 7. Performance Comparison with SOTA Models (Accuracy %).
The LRT demonstrates robust transfer capabilities, particularly in the R→B task.

Method	Same-B	Same-R	Cross B→R	Cross R→B
ResNet50 [16]	99.1	99.2	49.4	77.5
EfficientNet-B0 [25]	99.2	99.3	50.1	88.2
SWIN Transformer [22]	99.4	99.5	55.8	74.3
Standard ViT [10]	99.5	99.6	63.2	82.8
LRT (Ours)	99.3	99.4	68.8	93.9

Cross-Script Generalization:

In cross-script scenarios, LRT consistently surpasses baselines, though performance depends on the transfer direction.

- **B→R Difficulty:** The Bangla to Roman transition is the most difficult. CNNs collapse to $\approx 50\%$ accuracy. While the Standard ViT recovers to 63.2%, the **LRT** reaches **68.8%**. This 5.6% margin shows that Langevin dynamics extract robust features even under severe visual domain shifts.
- **R→B Dominance:** In the Roman to Bangla task, LRT dominance is most pronounced (**93.9%**), exceeding the Standard ViT (82.8%) by 11 percentage points. The physics-based attention seemingly maps the writer’s underlying “energy landscape,” enabling effective transfer from the simpler Roman script to the complex Bangla abugida.

6. Complexity Analysis

The LRT achieves superior generalization, but this increases computational cost relative to standard Vision Transformers. We analyze this trade-off below.

6.1. Notation

Let us define the primary variables that determine the complexity:

- N : The sequence length (number of patches + 1 ‘[CLS]’ token).
- D : The embedding dimension of the model.
- H : The number of attention heads.
- D_k : The dimension of each head’s Query, Key, and Value vectors, where

$$D_k = D/H.$$

- T : The number of simulation steps in a Langevin Attention Layer.
- D_h : The hidden dimension of the small MLP used to predict potential function parameters within each head.

6.2. Complexity of Standard Self-Attention

A standard Multi-Head Self-Attention block performs the following key operations:

1. **Q, K, V Projections:** Three parallel linear transformations of the input sequence of shape (N, D) to produce Q, K, and V. Each is a matrix multiplication of complexity $\mathcal{O}(ND^2)$. Total: $3 \times \mathcal{O}(ND^2) = \mathcal{O}(ND^2)$.
2. **Scaled Dot-Product Attention (per head):**
 - Attention Score Calculation ($\mathbf{QK}^T$): A matrix multiplication between matrices of shape (N, D_k) and (D_k, N) , resulting in an (N, N) attention matrix. Complexity: $\mathcal{O}(N^2 D_k)$.
 - Weighted Sum (Attention $\times$ V): A matrix multiplication between the (N, N) attention matrix and the Value matrix of shape (N, D_k) . Complexity: $\mathcal{O}(N^2 D_k)$.
3. **Multi-Head Aggregation:** The computations for H heads are done in parallel. The total complexity for the attention part is

$$H \times \mathcal{O}(N^2 D_k) = \mathcal{O}(N^2 (H \cdot D_k)) = \mathcal{O}(N^2 D).$$

4. **Final Output Projection:** A final linear layer of complexity $\mathcal{O}(ND^2)$.

Thus, the total complexity of a standard self-attention block is dominated by the dot-product attention and the linear projections:

$$\mathcal{C}_{\text{Attention}} = \mathcal{O}(N^2 D + ND^2) \quad (16)$$

6.3. Complexity of Langevin Attention

Our Langevin Attention Layer replaces the scaled dot-product attention with a multi-step simulation.

1. **Q, K, V Projections:** Identical to the standard model. Complexity: $\mathcal{O}(ND^2)$.
2. **Langevin Simulation Loop (per head):** This is the main source of additional complexity. The loop runs for T steps. Inside the loop:
 - **Force Calculation:** This is an all-pairs interaction. For each of the N^2 pairs of particles (i, j) :
 - We call an MLP with hidden size D_h on the concatenated $(\mathbf{q}_i, \mathbf{k}_j)$ vectors. The complexity of this MLP call is roughly $\mathcal{O}(D_k D_h)$.
 - The subsequent force calculation is $\mathcal{O}(D_k)$.
 The total complexity to compute all pairwise force parameters is therefore $\mathcal{O}(N^2 D_k D_h)$.
 - **Force Aggregation:** For each particle i , we sum the force contributions from all N other particles. This has a complexity of $\mathcal{O}(N^2 D_k)$.

- **State Update:** Updating the position and velocity for all N particles is a linear operation of complexity $\mathcal{O}(ND_k)$.

The dominant operation within the loop is the pairwise force parameter calculation. The total complexity for the simulation loop across all H heads is therefore:

$$\mathcal{C}_{\text{Sim}} = T \times H \times \mathcal{O}(N^2 D_k D_h) = \mathcal{O}(TN^2 DD_h) \quad (17)$$

3. **Final Output Projection:** Identical to the standard model. Complexity: $\mathcal{O}(ND^2)$.

The total complexity of our Langevin Attention block is therefore:

$$\mathcal{C}_{\text{LRT-Attn}} = \mathcal{O}(TN^2 DD_h + ND^2). \quad (18)$$

6.4. Discussion and Trade-offs

A direct comparison of the complexities, summarized in Table 8, reveals the architectural trade-off.

TABLE 8. Computational Complexity Comparison.

Attention Mechanism	Dominant Asymptotic Complexity
Standard Self-Attention	$\mathcal{O}(N^2 D + ND^2)$
Langevin Attention (Ours)	$\mathcal{O}(T \cdot D_h \cdot N^2 D + ND^2)$

The complexity of the LRT’s attention mechanism is higher than that of a standard ViT by a factor of approximately $T \cdot D_h$.

- T , the number of simulation steps, introduces a direct linear scaling factor. Our experiments show that $T = 5$ is optimal, making the LRT roughly five times slower in this component.
- D_h , the hidden size of the potential-predicting MLP, is another scaling factor. Although D_h is a small constant (e.g., 32 or 64), it is multiplied across the entire quadratic term.

This increased computational cost is a deliberate design choice. We trade raw computational efficiency for a much stronger physical inductive bias. For problems like signature identification, where the sequence length N is fixed and relatively small (e.g., $14 \times 14 + 1 = 197$ for our setup), the quadratic complexity is manageable. The significant performance gains in cross-script generalization, a task where standard models fail, justify this trade-off. However, this complexity would make the current LRT architecture challenging to apply to problems with very long sequences, such as high-resolution images or long text documents, without further modifications like N-body approximation methods.

6.5. Empirical Inference Analysis

TABLE 9. Empirical Inference Complexity Benchmarks.

Model	Inference Latency (ms/img)	Throughput (FPS)
EfficientNet-B0	2.8 ms	357
ResNet50	3.4 ms	294
SWIN Transformer	9.1 ms	110
Standard ViT	8.5 ms	117
LRT (Ours, $T = 5$)	31.2 ms	32

As predicted by our math, the LRT takes about 3.6 times longer to process an image than a standard ViT because it performs 5 simulation steps ($T = 5$). However, since real-world forensic handwriting analysis is usually done offline and does not require instant real-time results, we feel that this extra processing time is a small price to pay. In return, the model gains a massive 11.1 % increase in accuracy when dealing with completely different handwriting scripts.

7. Conclusion and Future Work

In this paper, we introduced the Langevin Resonance Transformer (LRT), a novel deep learning architecture designed to overcome the critical generalization failure of conventional models in cross-script writer identification. The LRT redefines self-attention by replacing abstract similarity scores with a differentiable simulation of Langevin dynamics, forcing the model to learn a physically-grounded representation of a writer’s biomechanical style. Our extensive experiments on the BRS-ID dataset empirically validated this approach, demonstrating that the LRT establishes a new state-of-the-art in cross-script performance, outperforming prominent CNN and Transformer baselines by a significant margin. The primary strength of the LRT is its superior generalization, a direct result of its physics-informed inductive bias. However, this comes at the cost of increased computational complexity compared to standard Transformers, which remains the main limitation of our approach.

Our future work prioritizes two directions. First, we aim to improve efficiency, possibly by applying N-body approximations to the Langevin Attention Layer. Second, we will test this architecture on related tasks like forgery detection, where modeling dynamic interactions is equally important.

REFERENCES

- [1] ALI, M.—RAZA, M.—ULLAH, N.—AHMAD, I.—KIM, D.: *Deep learning in writer identification: a comprehensive review of recent trends and future directions*, IEEE Access **11** (2023), 55412–55435.
- [2] ALWADEI, A. H.—KESSENTINI, Y.: *A multi-scale deep learning-based approach for offline writer identification*, Multimedia Tools and Applications **82** (2023), 14939–14958.
- [3] APPALARAJU, S.—TANG, P.—DONG, Q.—SANKARAN, N.—ZHOU, Y.—MAN-MATHA, R.: *Docformerv2: Local features for document understanding*. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38 (2024), pp. 709–718.
- [4] CHEN, J.—HE, S.—SCHOMAKER, L.: *FragNet: Writer Identification Using Small Handwriting Fragments*. In: *2023 International Conference on Document Analysis and Recognition (ICDAR), IEEE (2023)*, pp. 112–127.
- [5] CHEN, Z.—ZHANG, L.: *The next frontier: A survey of physics-informed architectures in deep learning*, arXiv preprint arXiv:2501.01234 (2025).
- [6] CHRISTLEIN, V.—SEURET, M.—NICOLAOU, A.—RIBA, P.—ANGELOV, D.: *Writer identification for historical documents: A survey*. In: *16th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2021, Springer, 2021*. pp. 1–16.
- [7] DAS, A.—ROY, P.: *Cross-Script Writer Identification via Contrastive Style Representation Learning*. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2024)*, pp. 1–9.
- [8] DEY, S.—DUTTA, A.—LLADÓS, J.—PAL, U.: *HanDA: A transformer-based system for handwritten document analysis*. In: *26th International Conference on Pattern Recognition (ICPR), IEEE (2022)*, pp. 2501–2508.
- [9] DIAZ, M.—FERRER, M. A.—CARMONA, C.—MORALES, A.: *A review on offline and online signature verification systems*, IEEE Transactions on Cybernetics **52** (2021), 9177–9190.
- [10] DOSOVITSKIY, A.—BEYER, L.—KOLESNIKOV, A.—WEISSENBORN, D.—ZHAI, X.—UNTERTHINER, T.—DEGHANI, M.—MINDERER, M.—HEIGOLD, G.—GELLY, S. et al.: *An image is worth 16x16 words: Transformers for image recognition at scale*, arXiv preprint arXiv:2010.11929 (2020).
- [11] DUBE, S.—JAIN, A. K.—VATSA, M.—SINGH, R.: *Forensic handwriting analysis in the digital age: A review of recent advancements and challenges*, Wiley Interdisciplinary Reviews: Forensic Science **4** (2022), e1439.
- [12] GHOSH, M.—CHATTERJEE, S.—MUKHERJEE, H.—SEN, S.—OBAIDULLAH, S. M.: *Text/non-text scene image classification using deep ensemble network*. In: *Proceedings of International Conference on Advanced Computing Applications: ICACA (2021)*. Springer-Verlag, 2021, pp. 561–570.
- [13] GHOSH, M.—MUKHERJEE, H.—OBAIDULLAH, S. M.—GAO, X.-Z.—ROY, K.: *Scene text understanding: recapitulating the past decade*, Artificial Intelligence Review **56** (2023), 15301–15373.

- [14] GHOSH, M.—ROY, S. S.—MUKHERJEE, H.—OBAIDULLAH, S. M.—GAO, X.-Z.—ROY, K.: *Movie title extraction and script separation using shallow convolution neural network*, IEEE Access **9** (2021), 125184–125201.
- [15] GHOSH, M.—ROY, S. S.—MUKHERJEE, H.—OBAIDULLAH, S. M. —SANTOSH, K.—ROY, K.: *Understanding movie poster: transfer-deep learning approach for graphic-rich text recognition*, The Visual Computer (2022), 1–20.
- [16] HE, K.—ZHANG, X.—REN, S.—SUN, J.: *Deep residual learning for image recognition*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.
- [17] JAVAD, S. M.—EBRAHIMI, M.: *ViT-WI: A vision transformer-based approach for writer identification in offline handwritten documents*. In: *9th International Conference on Web Research (ICWR)*, IEEE (2023), pp. 105–110.
- [18] KARNIADAKIS, G. E.—KEVREKIDIS, I. G.—LU, L.—PERDIKARIS, P.—WANG, S.—YANG, L.: *Physics-informed machine learning*, Nature Reviews Physics **3** (2021), 422–440.
- [19] KAUSHAL, V.—KUMAR, P.—BALASUBRAMANIAN, V.: *Unsupervised domain adaptation for cross-script writer identification*. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2022), pp. 1245–1254.
- [20] KUMAR, A.—SINGH, M.: *Synergizing CNNs and transformers for enhanced writer identification*. In: *International Conference on Pattern Recognition (ICPR)*, 2024, IEEE (2024), pp. 1–8.
- [21] LI, Y.—ZHANG, D.—WANG, P.: *Handwriting representation with graph neural networks*. In: *Proceedings of the 30th ACM International Conference on Multimedia* (2022), pp. 1513–1521.
- [22] LIU, Z.—LIN, Y.—CAO, Y.—HU, H.—WEI, Y.—ZHANG, Z.—LIN, S.—GUO, B.: *Swin transformer: Hierarchical vision transformer using shifted windows*. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 10012–10022.
- [23] PANDEY, A.—JAIN, R.: *Zero-shot writer identification*, Pattern Analysis and Applications (2023), 1–13.
- [24] SULAIMAN, A.—OMAR, K.—NASRUDIN, M. F.: *Few-shot writer identification using a siamese network*. In: *International Conference on Computer and Information Sciences IEEE* (2021), (ICCOINS), pp. 257–262.
- [25] TAN, M.—LE, Q. V.: *Efficientnet: Rethinking model scaling for convolutional neural networks*. In: *International Conference on Machine Learning*, PMLR (2019), pp. 6105–6114.
- [26] TANG, Y.—WU, X.—PU, S.: *Attention-based Multi-scale Feature Fusion for Writer Identification*. In: *2022 IEEE International Conference on Image Processing (ICIP)*, IEEE (2022), pp. 1666–1670.
- [27] XING, L.—WU, Y.: *Cross-script writer identification with disentangled representation learning*, Pattern Recognition **118** (2021), 108027.
- [28] ZHAO, S.—LU, Y.—HE, S.: *Self-Supervised Representation Learning for Writer Identification*. In: *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE (2023), pp. 1–5.

Received December 10, 2025

Revised March 15, 2026

Accepted March 18, 2026

Publ. online April 30, 2026

Sk Golam Sarowar Hossain

Sk Md Obaidullah

Department of Computer Science & Engineering

Aliah University

Kolkata

postal code-700160

INDIA

E-mail: sarowar@aliah.ac.in

sk.obaidullah@gmail.com

Mridul Ghosh

Department of Computer Science

Shyampur Siddheswari Mahavidyalaya

Howrah

postal code-711314

INDIA

E-mail: mridulxyz@gmail.com

Tonmoy Mete

Department of Computer Science

Asutosh College

Kolkata

postal code-700026

INDIA

E-mail: tonmoy.mete@asutoshcollege.in

Mária Ždímalová

Department of Mathematics and Descriptive Geometry

Faculty of Civil engineering

Slovak University of Technology in Bratislava

Bratislava

postal code-810 05

SLOVAKIA

E-mail: maria.zdimalova@stuba.sk

Kaushik Roy

Department of Computer Science

West Bengal State University

Kolkata

INDIA

E-mail: kaushik.mrg@gmail.com