Results in Engineering 27 (2025) 106425

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Research paper

# DaGAM-Trans: Dual graph attention module-based transformer for offline signature forgery detection

Sara Tehsin[a], Ali Hassan[a], Farhan Riaz[b], Inzamam Mashood Nasir[c,*]

[a] Department of Computer and Software Engineering, National University of Sciences and Technology, Islamabad, 44080, Pakistan
[b] School of Computer Science, University of Lincoln, Lincoln, LN6 7DQ, UK
[c] Faculty of Informatics, Kaunas University of Technology, Kaunas, 51368, Lithuania

ARTICLE INFO

*Keywords:*
Offline signature verification
Graph attention networks
Vision transformers
Channel attention
Writer-independent verification
Signature forgery detection
Attention visualization

ABSTRACT

**Objective:** The detection of signature forgeries is a critical challenge in domains such as forensic investigations, financial security, and biometric authentication. This study aims to improve the offline detection of signature forgery, especially in cases of skilled forgeries where traditional methods relying on handcrafted features and statistical models often fail to distinguish between genuine and forged signatures. **Material and Methods:** To address these limitations, we propose DaGAM-Trans, a Dual Attention Graph Attention Module-based Transformer model that integrates Graph Attention Networks (GAT) with a Transformer backbone. The Transformer architecture plays a key role in modeling global contextual dependencies across the entire signature image, enabling the system to capture long-range structural information crucial for distinguishing genuine signatures from skilled forgeries. The architecture comprises a Graph Attention Module (GAM) to capture spatial dependencies using multi-head graph attention and graph convolution layers, and a Channel Attention Module (CAM) to amplify discriminative features and suppress irrelevant information. Additionally, a Graph Multi-Scale Adaptive Pooling (GMSAPool) layer is introduced to optimize feature aggregation by preserving salient details and reducing redundancy. The model is evaluated on four publicly available signature datasets: SigComp2011, BHSig260, CEDAR, and UTSig, within a cross-language verification setting. **Results:** DaGAM-Trans demonstrates superior performance across all datasets. Notably, it achieves 100% accuracy on the CEDAR dataset, with a False Acceptance Rate (FAR) as low as 0.001 and an Equal Error Rate (EER) reduced to 9.5%. Comparative experiments confirm that DaGAM-Trans outperforms existing state-of-the-art methods in terms of accuracy, FAR, FRR, and EER. Visualization through attention maps further validates the model's capacity to localize and highlight discriminative regions in signature images. **Conclusion:** The proposed DaGAM-Trans model effectively enhances offline signature forgery detection by leveraging dual attention mechanisms and adaptive graph-based pooling. Its outstanding performance, especially in reducing error rates and handling cross-linguistic signature variability, demonstrates its robustness and applicability for real-world biometric and forensic authentication tasks.

## 1. Introduction

Biometrics improves the safety and convenience of our lives by estimating an individual's identification through a variety of features. Physiological biometrics and behavioral biometrics are the two primary categories into which biometrics can be classified [1,2]. A biometric characteristic that is both distinctive and ubiquitous is the most effective. Additionally, the attributes must be easily accessible and maintain their consistency over an extended period. Fingerprints, eye scans, and facial recognition are all included in the field of physiological biometrics. The primary focus of behavioral biometrics is the handwriting, speech, and gait of an individual. Despite the differences between these biometrics, they collectively contribute to the development of a dependable personal authentication system. Handwritten signatures, which are among the earliest symbols recognized for verifying identity, offer a variety of benefits [3]. Signatures are intricately linked to the unique writing styles of each individual and are notoriously difficult to replicate. Additionally, they are frequently accessible. As a result, signatures are extensively employed in the military, business, and finance [4].

[*] Corresponding author.
*E-mail address:* inzamam.nasir@ktu.edu (I.M. Nasir).

https://doi.org/10.1016/j.rineng.2025.106425
Received 2 July 2025; Received in revised form 15 July 2025; Accepted 21 July 2025

Available online 29 July 2025

The objective of signature verification is to either confirm or deny the authenticity of an individual's claimed identification by analyzing their handwritten signatures. The verification procedure requires educated specialists, rendering authentication both protracted and inconsistent depending on the human practitioners involved. The rapid advancement of artificial intelligence and pattern recognition has rendered automated signature verification systems increasingly vital due to prevailing obstacles. Signature verification methods are classified into two main types based on the training procedures and recording techniques employed for data collecting [5]. The teaching methodologies are categorized into writer-dependent and writer-independent types. Writer-dependent approaches necessitate the training of models using signature samples specific to individual users, whereas writer-independent methods create universal identification systems capable of processing unknown signatures during execution. In data acquisition, signatures can be categorized into online and offline types. Düzen approaches, such as online signatures, yield data regarding stroke direction alongside pressure parameters, pen angle, and movement speed, whereas offline signatures consist of static signatures derived from manual sources and scanning processes. The optical scanner apparatus is used to acquire offline data, which is then provided as static signature images [6].

Writer-independent approaches possess significant value owing to their capacity to utilize diverse user sets for training and testing, hence enhancing performance for unrecognized participants. Offline signatures derived from scanned documents are more appropriate for real-world use due to their ease of acquisition. The key challenge in this sector is offline signature verification that operates independently of the writers, as implementing such systems in real-world environments closely resembles this work scenario [7,8]. Systems utilizing such authentication technologies categorize handwriting signatures into two types: authentic and fraudulent. Forged signatures are classified into three categories: random forgeries, easy forgeries, and expert (or sophisticated) forgeries. A random forgery occurs when the forger is unaware of the signer's initials and signature patterns, resulting in a fabrication that does not replicate authentic signature characteristics [9]. A straightforward forgery occurs when the forger is only acquainted with their name and has never encountered authentic signatures. The forger who has meticulously exercised the writing style and observed authentic signatures is able to achieve proficient forging. Authentic and counterfeit signatures exhibit substantial similarities in specific circumstances [10].

The recent advancement of deep learning methodologies has significantly enhanced signature verification systems by enabling the automatic identification of discriminative high-level features in raw signature images. Convolutional Neural Networks have robust local texture and shape extraction capabilities; nevertheless, these models are ineffective in managing substantial spatial dependencies necessary for distinguishing authentic signatures from forgeries [11]. The implementation of transformer-based architectures from natural language processing has been advantageous for vision applications, as these systems establish links among global contextual factors via their self-attention processes. The signature verification system employs transformative frameworks to identify significant patterns and channel-based interactions that facilitate complicated forgery detection. The integration of transformers with graph attention modules in the suggested method facilitates the amalgamation of local connectivity and global semantic links, resulting in thorough offline signature analysis solutions [12].

In this paper, we have developed an offline signature verification system, DaGAM-Trans. This system can perform at a level that is comparable to the most advanced methods despite utilizing only a small number of reference signatures. Self-attention and channel attention mechanisms based on Graph Attention Network (GAT) enhance signature forgery verification through structural and feature-based learning. GAT-based self-attention dynamically applies relevance weights to adjacent nodes, enabling the model to concentrate on essential structural linkages in the signature network and minimize irrelevant connections. Feature extraction and the model's ability to distinguish between genuine and fraudulent signatures are enhanced by channel attention, which emphasizes the most critical channels in the signature representation. To robust, high-accuracy signature verification that is resistant to sophisticated forgeries, this hybrid technique integrates discriminative features and geographical dependencies. The primary contributions of this paper are as follows:

- A novel DaGam-Trans has been proposed to capitalize on the offline signature verification assignment by leveraging the capabilities of vision transformers. It employs a dual attention mechanism that integrates self-attention and channel attention knowledge to verify signatures.
- The integration of a Channel Attention Module (CAM) enhances semantic relationships among channel-wise feature maps, enabling the model to concentrate on critical features. This module, based on GAT, establishes the most recent standard for transformer-based signature verification at the channel refinement level.
- To generate a global attention feature, the proposed GMA Pool block is employed to learn the essential information within the features using a self-attention mechanism. The results of the testing conducted indicate that GMA Pool is capable of downsampling the necessary global-attention feature more effectively.
- The proposed work is evaluated on four publicly available offline signature datasets: SigComp2011, BHSig260, CEDAR, and UTSig Dataset. The experimental section illustrates that the proposed model consistently outperforms these datasets. Comprehensive ablation experiments illustrate the proposed model's efficacy.

The remainder of the paper is structured as follows: In Section 2, we have compiled a catalog of past methods implemented in signature verification. In Section 3, we have designated our methodology and other necessary prerequisites for the project. Section 4 summarizes the experimental protocol, datasets utilized, and results attained through the application of the proposed model. In Section 5, we conclude our work by outlining our findings and future work.

## 2. Literature review

An overview of the research done concerning the work related to the signature datasets is reviewed in this work. Hung et al. [13] developed an offline signature verification method employing a Siamese Triplet CNN and a fully connected neural network for binary classification. The approach detected skilled and straightforward forgeries. The main benefit was one-shot learning, which allowed verification with little training data. For signature images, the model retrieved features and compared them using distance. Triplet loss helps identify fake signatures during training. On the BHSig260 dataset, the system achieved a False Acceptance Rate (FAR) of 13.66%, somewhat better than state-of-the-art techniques. This method is suitable for financial applications and authentication systems with limited user samples.

Geometrical characteristics and an ANN classifier were used to verify signatures by Jain et al. [14,15]. The method extracted the center of mass, number of isolated points, connected components, signature length, and height without deep learning. An ANN-based classifier was trained using these attributes to detect counterfeit signatures. An online dataset with solely signature images, MCYT-100, and BHSig260, including Bengali and Hindi signature samples, were used to evaluate the model. The approach produced low EER on MCYT-100 and higher accuracy on BHSig260, proving geometrical features effective for signature verification but failed with complex forgeries.

For offline signature verification, Musleh et al. used deep learning and GA [16]. The model began with a CNN extracting discriminative features from signature photos. Genetic Algorithms (GAs) optimized hyperparameters such as layer count, filter size, and activation function. The GA selected the most accurate CNN architectures from a population. We tested the approach on BHSig260 (Bengali & Hindi), GPDS, and CEDAR. The results showed a 2.5% FRR, 3.2% FAR, 2.35% EER, and 97.73% accuracy. This method enhanced efficiency by automating CNN development and modification, making it suitable for banking and document verification applications.

Jain's feature ensemble-based method combines geometrical feature extraction with MobileNet characteristics to create a verification classifier [17]. The model utilizes handcrafted geometrical features and MobileNet deep learning representations. A combination of classic and modern classifiers was used. On the BHSig260 dataset, the approach achieved 99.4% Bengali and 99.3% Hindi signature accuracy. Combining feature extraction approaches leads to improved verification accuracy and resilience, making it ideal for practical applications.

Ren et al. [18] developed a graph-based method for writer-independent offline signature verification using the Vision Graph Convolutional Network (SigGCN). It converted signature images into graph-structured data, where nodes represented CNN-extracted signature features, edges were K-Nearest Neighbors (KNN)-connected signature components, and a GCN learned signature patterns. This method handles intra-class changes better than CNN-based algorithms because it has spatial correlations between signature parts. By eliminating class imbalance, a margin-based targeted loss function optimized training. Our model was tested on CEDAR, BHSig260-Bengali, and BHSig260-Hindi. Graph-based learning showed remarkable accuracy in signature verification, especially for new writers.

Culqui-Culqui et al. [19] proposed a CNN-classified offline signature verification method. The datasets were CEDAR (public) and GC-DB (locally gathered with 121 signers, each giving 45 signatures). Despite acquisition noise, the model classified handwritten signatures well. The CNN-based technique obtained over 90% accuracy and was computationally lightweight, making it appropriate for mobile and portable devices. It outperformed previous signature verification and fraud detection systems. In addition to conventional and deep learning approaches, recent research in tensor factorization and data recovery offers valuable insights for handling intra-class variability and noise in signature verification. High-dimensional and incomplete data—common in scanned handwritten signatures—pose challenges in representation learning and classification.

For instance, [20] introduced a robust optimization strategy for learning latent factors from sparse and noisy tensors, which can be adapted to improve feature aggregation in complex visual patterns like signatures. Similarly, the study in [21] incorporated Cauchy loss to address outliers and heavy noise, which is highly relevant for dealing with poorly scanned or degraded signature inputs.

Hashim et al. [22] presented the Fusion-based Hybrid Deep Neural Network (FHDNN), designed as an offline signature verification system. FHDNN created a comprehensive hybrid feature vector utilizing appearance features derived from Principal Component Analysis, texture features obtained from the Gray-Level Co-occurrence Matrix, and frequency-domain features from the Fast Fourier Transform. On SigComp2011 and CEDAR datasets, the model outperformed previous approaches with 100% accuracy. The FHDNN model had good precision, recall, and F-score, making it an accurate and efficient signature verification method. For offline signature verification, Ashvitha and Riyazuddin developed a CNN-HOG hybrid feature extraction method [23]. The extracted feature set was optimized using Decision Trees. The performance of three classifiers—LSTM, SVM, and KNN—was assessed. Experimental results on the CEDAR dataset showed high accuracy in recognizing competent frauds. Using Xception with HOG-RFE and a Voting Classifier, the study achieved 100% accuracy, making it reliable for real cybersecurity applications. Another hybrid feature extraction method by Alsuhimat and Mohamad combined CNN and HOG with Decision Tree-based feature selection. The LSTM, SVM, and KNN classifiers achieved 95.4%, 95.2%, and 92.7% accuracy on UTSig and 93.7%, 94.1%, and 91.3% accuracy on CEDAR. CNN's deep feature extraction and HOG's shape-based analysis improved verification accuracy, especially for sophisticated forgeries.

Singh et al. [24] used InceptionV3, ResNet, and VGG pretrained modeling for offline signature verification using transfer learning and deep neural networks. CEDAR, ICDAR-2011, and BHSig260 datasets were used to test the model for False Acceptance Rate (FAR) and FRR. Transfer learning works for high-performance signature verification, as shown by the InceptionV3 model's 99.10% accuracy, 1.03% FRR, and 0.74% FAR. To improve verification accuracy, the study proposed strong preprocessing and feature extraction methods.

Hashim et al. [25] proposed a hybrid handwritten signature identification system using LDA for appearance-based features, FFT for frequency-based features, and GLCM for texture-based features. Four machine learning classifiers (Naïve Bayes, K-Nearest Neighbor (KNN), Decision Tree, and AdaBoost) identified signatures using fused characteristics. Using Naïve Bayes, the model scored 100% accuracy on CEDAR and 94.43% accuracy on SigComp2011 datasets. Multiple feature extraction methods and machine learning increase signature identification accuracy in financial and legal document verification applications, according to the study.

Özyurt et al. [26] developed an offline handwriting signature verification system using transfer learning and feature selection. The model extracted features using MobileNetV2 and improved them using NCA, Chi-Square, and MI. SVM (RBF, poly, linear), KNN, Decision Tree, Linear Discriminant Analysis (LDA), and Naïve Bayes classifiers were examined. NCA, with 300 selected features, improved model accuracy to 97.7% from 91.3% without feature selection. The study emphasized that optimal feature selection decreases computational cost and improves accuracy. A deep learning-based signature recognition system employing DenseNet201 for feature extraction and Euclidean Distance for classification was proposed by Nugraha et al. [27]. After binary conversion, ROI extraction, and thinning, the model extracted features. The system achieved 99.44% accuracy on SigComp2009, SigComp2011, and a private dataset. Different dataset augmentation and split ratios were used to test resilience. Deep learning and distance-based categorization improved signature identification.

A 25-layer deep neural network model was used to identify offline handwritten signatures by Hashim et al. [28]. The model used LDA, FFT, and GLCM features. This technique was tested on SigComp2011, CEDAR, and SigArab (a private Arabic signature dataset). The model had 99.23% SigArab accuracy and 100% CEDAR and SigComp2011 accuracy. For secure biometric applications, deep learning models with hybrid feature extraction outperform conventional machine learning methods, according to the study. Zeng introduced an offline handwritten signature verification Multi-Scale Attention-Based Individual Character Network [29]. The Siamese network extracted signature stroke characteristics by comparing original and gray-inversed signatures, ensuring the model learned stroke-based patterns rather than background noise. Our training and testing dataset was SigComp2011, with 82% accuracy. In complex signature verification tasks, attention-based networks increase feature extraction, especially in recognizing experienced forgeries.

The Alsuhimat and Mohamad study suggested an offline signature verification system employing LSTM and HOG for feature extraction [30]. The model trained and tested on UTSig and CEDAR achieved 92.4% and 87.7% accuracy, respectively. The strategy outperforms K-Nearest Neighbor (KNN), Support Vector Machine (SVM), CNN, SURF, and Harris corner identification. The study showed that HOG for feature extraction and LSTM for sequence learning improved skilled forgery classification. They combined CNN and HOG to extract essential features and enhanced them using Decision Tree-based feature selection in another investigation [31]. The model obtained 95.4%, 95.2%, and 92.7% accuracy on UTSig and 93.7%, 94.1%, and 91.3% accuracy on CEDAR using LSTM, SVM, and KNN classifiers. Extraction of robust features improves signature verification accuracy, especially for experienced forgeries, which are tougher to detect, according to the study.

**Table 1**
Comparative Literature Review of Offline Signature Verification Methods.

| Reference | Method | Dataset(s) | Performance | Remarks |
|---|---|---|---|---|
| Hung et al. [13] | Triplet Siamese CNN | BHSig260 | 13.66% FAR | One-shot learning, suitable for low-data settings |
| Jain et al. [14] | Geometrical Features + ANN | MCYT-100, BHSig260 | Low EER | Good for simple forgeries, less effective for skilled ones |
| Musleh et al. [16] | CNN + Genetic Algorithm | CEDAR, BHSig260, GPDS | 97.73% Acc | Hyperparameter optimization via GA for CNN architecture |
| Jain et al. [17] | Geometrical + MobileNet | BHSig260 | 99.4% (Bengali), 99.3% (Hindi) | Combines handcrafted and deep features for enhanced accuracy |
| Ren et al. [18] | Graph-based (GCN) | CEDAR, BHSig260 | High Accuracy | Models spatial relationships, effective in WI setting |
| Culqui-Culqui et al. [19] | CNN | CEDAR, GC-DB | >90% Acc | Lightweight CNN, mobile deployment capable |
| Hashim et al. [22] | Hybrid DNN (FFT + PCA + GLCM) | CEDAR, SigComp2011 | 100% Acc | Multi-domain feature fusion |
| Ashvitha & Riyazuddin [23] | CNN-HOG Hybrid + Voting Classifier | CEDAR | 100% Acc | Reliable for cybersecurity applications |
| Alsuhimat & Mohamad [23] | CNN + HOG + DT | UTSig, CEDAR | ~95% Acc | Enhances skilled forgery detection |
| Singh et al. [24] | Transfer Learning (InceptionV3, etc.) | CEDAR, ICDAR-2011, BHSig260 | 99.1% Acc | Robust pretrained models with strong generalization |
| Hashim et al. [25] | FFT + LDA + GLCM + ML Classifiers | CEDAR, SigComp2011 | 100%, 94.43% Acc | Suitable for legal/financial documents |
| Özyurt et al. [26] | MobileNetV2 + FS + SVM | Custom Dataset | 97.7% Acc | NCA feature selection enhances accuracy |
| Nugraha et al. [27] | DenseNet201 + Euclidean Distance | SigComp2009, SigComp2011 | 99.44% Acc | Robust preprocessing and feature extraction |
| Hashim et al. [28] | Deep Neural Network (25-layer) | CEDAR, SigComp2011, SigArab | Up to 100% Acc | Hybrid features + deep modeling |
| Zeng [29] | Multi-Scale Attention Siamese Network | SigComp2011 | 82% Acc | Learns stroke-based differences for skilled forgery detection |
| Alsuhimat & Mohamad [30] | LSTM + HOG + FS | UTSig, CEDAR | 92.4%, 87.7% Acc | LSTM improves skilled forgery recognition |
| Keykhosravi et al. [32] | DNN + Transfer Learning | Custom Dataset | >90% Acc | Works well even in noisy environments |
| Gutub et al. [33] | OctConv (Frequency-aware CNN) | CEDAR, UTSig, BHSig260 | High Accuracy | Improves frequency-based signature discrimination |
| Badie-Sajedi et al. [34] | Graph Neural Network + SIFT | MCYT-75, UTSig | 100%, 92.7% Acc | GNN effective with few samples and structural features |

The offline writer identification system described by Keykhosravi et al. [32] used a customized deep neural network trained on a novel right-to-left signature dataset from 85 participants in different environmental situations. Transfer learning-based feature extraction captured writer-specific traits. Other datasets, including the gathered one, indicated that the suggested approach could identify writers with 99% accuracy and above 90% accuracy, even under noisy situations. The study showed that the model could discern authorship and detect fake signatures for forensic and financial security applications. For effective offline signature authentication, Gutub et al. developed OctConv [33]. Compared to baseline CNN architectures, OctConv-based models lowered computational complexity while preserving accuracy. The model outperformed CNN and Capsule Networks in signature verification tasks on CEDAR, UTSig, BHSig260-Hindi, and BHSig260-Bengali datasets. The study showed that frequency-aware feature extraction could differentiate faked signatures, making OctConv a good cybersecurity authentication option. The Badie-Sajedi offline signature authentication architecture uses Graph Neural Networks (GNNs) [34]. GNN classified signatures as genuine or forged after extracting essential signature features using the Scale-Invariant Feature Transform (SIFT) technique and converting them into a graph structure. The system had 100% accuracy on MCYT-75 and 92.7% accuracy on UTSig. Graph-based learning enhances signature verification, especially with few training samples, the study found.

Existing offline signature authentication methods exhibit some drawbacks, as indicated by the research results examined. Conventional techniques mostly rely on manually produced features and superficial classifiers yet exhibit restricted efficacy in addressing significant intra-class variability and sophisticated forgeries [35]. Deep learning techniques, such as CNNs and hybrid models, enhance feature extraction; nevertheless, they fail to understand global spatial correlations and do not utilize multiple data sources or vast computational resources. Siamese networks and triplet-based learning systems provide the ability for one-shot learning; nevertheless, they are inadequate at identifying intricate manufactured patterns. The SigGCN method integrates structural model awareness but limits its semantic feature comprehension and cannot utilize global attention mechanisms. Most of the research in this domain focuses on writer-dependent contexts, overlooking the assessment of model generalization in transitioning between writer-independent and cross-linguistic scenarios [36]. Practical applications in forensic or legal fields necessitate models that uphold interpretability, as a non-transparent decision-making system poses a significant disadvantage in these areas. Table 1 provides an overview of proposed studies included in the literature.

Despite the progress in offline signature verification, existing models struggle with capturing long-range dependencies, adapting across languages, and maintaining interpretability. CNNs are limited in global context modeling, while graph-based methods like SigGCN lack semantic refinement. Furthermore, many methods do not generalize well to unseen users or cross-lingual scripts. These challenges motivate the design of a hybrid model that captures both spatial and semantic dependencies while remaining robust and interpretable.

## 3. Proposed methodology

To address the limitations discussed above, we propose DaGAM-Trans, a novel dual-attention-based transformer architecture tailored for offline signature verification. The key design motivation is to combine the strengths of both local structural modeling and global semantic attention. The model comprises three core innovations: a) A Graph Attention Module (GAM) that encodes spatial and topological relationships in signature strokes using multi-head graph attention and convolution; b) A Channel Attention Module (CAM) that emphasizes semantically

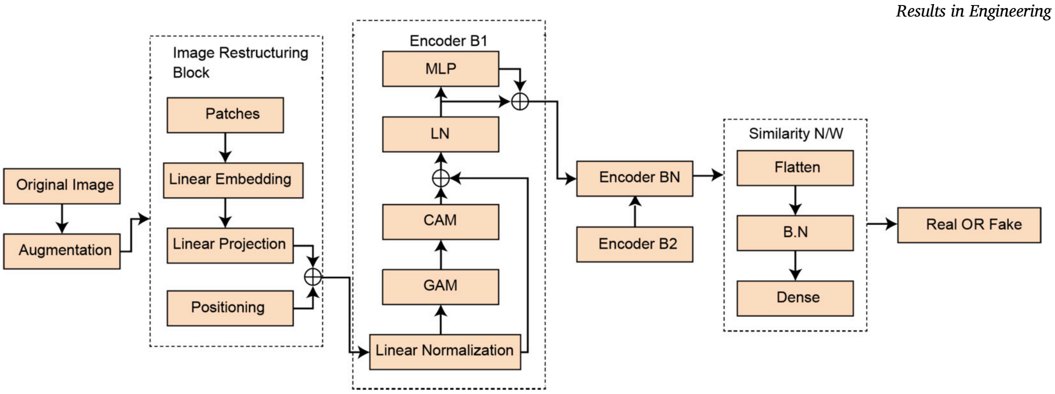

**Fig. 1.** Flow diagram of the proposed model.

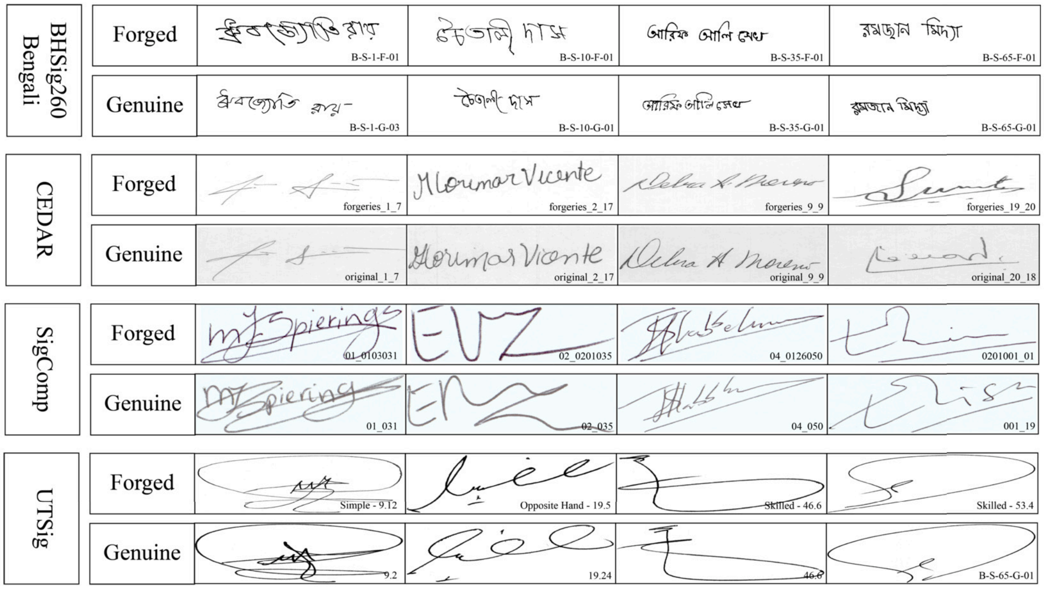

**Fig. 2.** Dataset samples from each selected dataset.

important feature channels, enhancing class separation and reducing noise; and c) A Graph Multi-Scale Adaptive Pooling (GMSAPool) layer that selectively preserves structurally relevant information while compressing redundant features. These components are embedded within a transformer backbone that enables deep contextual modeling of the signature image. The model workflow—from image preprocessing to classification—is illustrated in Fig. 1 and detailed in the following subsections. The overall architecture of proposed model is illustrated in Fig. 1.

### 3.1. Datasets description

The dataset required for this research was downloaded from the publicly available dataset site Kaggle [1]. The handwritten signature data consists of four diverse datasets. These are BHSig [37], CEDAR [38], UTSig [39], and SigComp2011 (offline) [40]. Bhsig-Bengali contains the signatures of 100 authors, each author with 24 original signatures and 30 forged signatures. The SigComp2011 dataset comprises offline (scanned images) signatures, categorized into genuine and forged samples. The dataset contains a total of 961 offline signatures (611 genuine, 350 forged). The Cedar dataset includes the signatures of 100 people; each person has 24 real signatures and 24 forged signatures. Apart from these, we have a Utsig-Persian dataset that contains signatures by 115 people, with each person having 27 real signatures and 42 forgeries. A summary of class-wise sample distribution across all four offline signature datasets is presented in Table 2, whereas sample images of each dataset are shown in Fig. 2.

**Table 2**
Summary of class-wise sample distribution across all four offline signature datasets.

| Name | Style | Author | Genuine | Forged |
|---|---|---|---|---|
| BhSig [37] | Bangali | 100 | 24 | 30 |
| CEDAR [38] | Latin | 55 | 24 | 24 |
| UTSig [39] | Persian | 114 | 27 | 42 |
| SigComp2011 (Offline) [40] | Dutch | 33 | 611 | 350 |

#### 3.1.1. Data augmentation

The process of verifying handwritten signatures is complicated due to the high probability and ease of signature forgery. The type of forgery can differ based on the information that the forger has access to. Class rebalancing is implemented on the signature images in the training set to improve classification performance. The unequal distribution of signature photos across various classes in the dataset necessitates class rebalancing. The dataset is supplemented with oversampling to ensure that the distribution of photos is uniform across all classes. Keras data augmentation techniques are employed to implement oversampling for the image class in the training dataset with a restricted number of images. These techniques include random rotations of 180 degrees, width

**Table 3**
Original and augmented sample distribution for each dataset.

| Dataset | Class Type | Original Samples | Augmented Samples | Total After Augmentation |
|---|---|---|---|---|
| BHSig260 | Genuine | 2,400 | 4,800 | 7,200 |
| | Forged | 3,000 | 3,000 | 6,000 |
| CEDAR | Genuine | 1,320 | 2,640 | 3,960 |
| | Forged | 1,320 | 1,320 | 2,640 |
| UTSig | Genuine | 3,078 | 3,078 | 6,156 |
| | Forged | 4,830 | 2,430 | 7,260 |
| SigComp2011 | Genuine | 611 | 1,222 | 1,833 |
| | Forged | 350 | 700 | 1,050 |

and height shifts of 0.1 fractions, random zooming within a range of 0.1, and random flipping in both horizontal and vertical orientations.

An empirical study confirms the impact of augmentation by evaluating model performance prior to and following its implementation, demonstrating a significant improvement in accuracy and generalization capability. This impact is shown in Table 3. The expanded dataset enhances training stability, hence improving test outcomes, particularly in writer-independent tasks that necessitate generalization to new authors. The training data is enhanced through realistic perturbations throughout the data augmentation process, which improves DaGAM-Trans's ability to distinguish authentic signatures from forgeries with more accuracy and fewer errors.

### 3.2. Image restructuring block

To enable the transformer encoder to process signature images effectively, the raw 2D image must be restructured into a sequence of fixed-length tokens. This is achieved through a four-step pipeline consisting of patch generation, linear embedding, positional encoding, and token sequencing. Each step ensures that the spatial and structural information within the signature is preserved while being reformatted for sequence-based attention mechanisms.

In the first step, the input grayscale signature image $I \in \mathbb{R}^{H\times W}$ is partitioned into a grid of non-overlapping patches of size $P \times P$. This results in a total number of patches $N = \frac{H\times W}{P^2}$. Each patch is then flattened into a 1D vector:

$$x_i \in \mathbb{R}^{P^2}, \quad \text{for } i = 1, 2, \dots, N \tag{1}$$

During this operation, the image is segmented into different localized components that undergo autonomous processing while preserving their original spatial coordinates, which will be reinstated via positional encoding.

In the second phase, the trainable projection matrix linearly maps each patch vector $x_i$ onto a fixed-dimensional embedding space:

$$z_i = W_E x_i + b, \quad z_i \in \mathbb{R}^D \tag{2}$$

where $W_E \in \mathbb{R}^{D\times P^2}$ is the learnable weight matrix, $b \in \mathbb{R}^D$ is the bias term, and $D$ is the embedding dimension expected by the transformer model. This projection step ensures that all input patches are uniformly represented in the same latent space, regardless of their raw pixel structure.

In the third step, to preserve the spatial ordering of patches—an essential feature for understanding signature layouts—learnable positional encodings are added to each embedded patch. The positional encoding matrix $PE \in \mathbb{R}^{N\times D}$ is summed element-wise with the patch embeddings:

$$\hat{z}_i = z_i + PE_i \tag{3}$$

This addition injects relative and absolute location information into the otherwise permutation-invariant transformer input, enabling the model to capture geometric context and sequential stroke flow within the signature.

Finally, in the fourth step, the position-enhanced embeddings $\hat{z}_i$ are concatenated to form the final input sequence to the transformer encoder:

$$\hat{Z} = [\hat{z}_1, \hat{z}_2, \dots, \hat{z}_N] \in \mathbb{R}^{N\times D} \tag{4}$$

Optionally, a class token $z_{[\text{CLS}]} \in \mathbb{R}^D$ may be prepended to the sequence to aggregate global information for classification purposes.

The output of this restructuring block is then passed to the transformer layers for global context modeling and attention-based feature extraction. This entire process effectively transforms the spatially structured signature image into a sequence of position-aware tokens. This allows DaGAM-Trans to leverage both self-attention and graph attention mechanisms, facilitating the extraction of fine-grained features and spatial dependencies crucial for high-accuracy signature forgery detection.

### 3.3. Transformer encoder

The transformer encoder consists of $N$ iterative layers, each with two sublayers. The initial sublayer is the Graph Attention Module. A totally connected feed-forward neural network constitutes the second sublayer. Both sublayers undergo regularization and residual connection. Each sublayer undergoes normalization. Residual connections exist between the input and output of each sublayer [41]. Each sublayer outputs $Output = x + sublayer(Norm(x))$, where $sublayer(x)$ is its function applied to $x$.

#### 3.3.1. Graph attention module

The DaGAM-Trans architecture incorporates a Graph Attention Module (GAM) tailored to simulate intricate spatial dependencies and structural interactions in offline handwritten signatures. Conventional convolutional or transformer-based methods struggle to accurately describe the spatial relationships of signature strokes in signature analysis, particularly for inter-class and intra-class variances, as well as skill-based forgery classification. GAM addresses this restriction by employing a graph representation of signature images, utilizing nodes to represent image patches or characteristics and edges to indicate their related proximity or similarity. Graph Attention Networks (GATs) facilitate automatic weight computation for adjacent nodes via attention processes, allowing the model to concentrate on significant signature regions. The graph-based modeling technique enables the network to identify essential topological features such as loops, crossings, and curvature variations that authenticate handwritten signatures. Fig. 3 illustrates the overall architecture of GAM.

**Graph Autoencoder**: The graph-based algorithms are unable to incorporate the values of $Q$, $K$, and $V$ into their calculations. Given that the input values are $\text{F}_{hi}$, they are passed through a computable graph structure as:

$$G = (\mathcal{GA}, f) = GraphAutoencoder(F_{hi}) \tag{5}$$

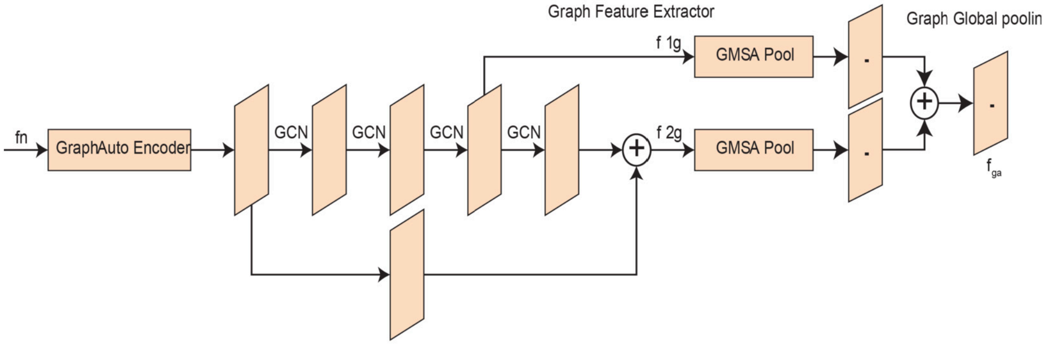

**Fig. 3.** The proposed Graph Attention Module (GAM) has three integral components: a Graph Autoencoder for structural encoding, a Graph Feature Extractor utilizing stacked GCN layers for relational learning, and a Graph Global Pooling mechanism through GMSAPool for adaptive feature aggregation.

The autoencoder component is employed to encode the F$_{hi}$ into the computable graph, where $GA$ is the graph adjacency matrix used for graph construction. The feature node $f$ can be expressed as:

$$f = \{f_1, f_2, \dots, f_n\} \tag{6}$$

Where $F_i \in f$ donates the implanting feature of node $i$, and n donates total nodes.

**Graph Feature Extractor**: The Graph Feature Extractor utilizes a Graph Convolutional Network (GCN) to identify sophisticated features by modeling node-to-node relational relationships. The engineered module expands network receptive fields to collect information from both proximal and distant node neighbors. The model enhances global representation capabilities by incorporating a four-layer GCN structure alongside two-graph global pooling processes, facilitating extensive feature dissemination and efficient summarization of node-level data.

The goal of a graph convolution network is to learn a mapping function $f(\cdot)$ over a graph $G(V,E)$, where $V$ and $E$ represent the set of nodes and edges. This function operates on the node feature matrix $F^L \in \mathbb{R}^{n\times D}$ and the corresponding correlation matrix $M \in \mathbb{R}^{n\times n}$ as inputs. Here, $n$ shows the number of nodes, $D$ is the dimensionality of each node feature, and $L$ is the linear function. The output of the node features learned by GCN is $F^{L+1} \in \mathbb{R}^{n\times D'}$. The non-linearity can be added in each GCN by:

$$F^{(L+1)} = f(F^{(L)}, M) \tag{7}$$

The convolution function $f(\cdot)$ Proposed by the author [42], can be further represented by:

$$\begin{aligned} F^{(L+1)} &= \delta_{lr}\left(F^{(L)}\hat{M}T^{(L)}\right) \\ \hat{M} &= \tilde{N}^{-1/2}\tilde{M}\tilde{N}^{-1/2} \\ \tilde{N}_{ii} &= \sum_j \tilde{M}_{ij} \\ \tilde{M} &= M + I_k \end{aligned} \tag{8}$$

where $T^{(L)} \in \mathbb{R}^{d\times d'}$ denotes a learnable transformation matrix, $\hat{M}$ is the normalized representation of the adjacency matrix $M$, $I_k$ is the identity matrix of size $k\times k$, and $\delta(\cdot)$ represents the LeakyReLU activation function.

$$c_{ij} = a(W_{\mathrm{nf}}f_i, W_{\mathrm{nf}}f_j) = \delta_{lr}\left(\alpha^t\left[W_{\mathrm{nf}}f_i \,\|\, W_{\mathrm{nf}}f_j\right]\right) \tag{9}$$

The attention coefficient $c_{ij}$, which emphasizes the significance of node features $nf_j$ to the node function $nf_i$, has been computed by performing the self-attention operation on the nodes. The shared weight matrix is denoted by $W$, while $t$ represents transposition and $\|$ represents the concatenation operation. In particular, the vector $\boldsymbol{\alpha}$ is parametrized and applied to LeakyReLU nonlinearity, and $a(.)$ is a single-layer feed-forward neural network. The disguised attention in GAT is compelled by the computation of $c_{ij}$ for nodes $j \in N_i$ [43]. In this context, $N_i$ represents a set of neighboring nodes of node $i$. The softmax function is employed to normalize the attention coefficients, and the multi-head attention is used to normalize the learning process of self-attention.

$$\begin{aligned} a_{ij} &= \mathrm{softmax}(c_{ij}) = \frac{\exp(c_{ij})}{\sum_z \exp(c_{iz})} \\ nf'_i &= \|_{h=1}^{\hat{H}}\delta\left(\sum_{j\in\mathcal{N}} a^h_{ij}W^h nf_j\right) \end{aligned} \tag{10}$$

The single-layer feed-forward neural network is used with self-attention to save computational costs and to calculate the attention coefficients. It is described as:

$$c_{ij} = a(W_q nf_i, W_k nf_j) = \frac{(W_q nf_i)\cdot(W_k nf_j)^T}{\sqrt{d_s}} \tag{11}$$

To enhance the capacity for exploring subspaces, the multi-head attention operation is used as follows:

$$\begin{aligned} \mathrm{MSA}(nf_i, nf_j) &= \|_{h=1}^{\hat{H}}(H_1, H_2, \dots, H_h)W_0 \\ \mathrm{Attention}(Q,K,V) &= \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_s}}\right)V \\ H_h(nf_i, nf_j) &= \mathrm{Attention}(W^h_q nf_i,\ W^h_k nf_j,\ W^h_{nf} nf_j) \end{aligned} \tag{12}$$

where $d_s$ is a scaling factor and $\{W^h_q, W^h_t, W^h_{nf}, W_0\} \in \mathbb{R}^{d\times d}$ denote projection matrices. The scaling factor $d_s$ is calculated as $d_s = \frac{d}{\hat{H}}$, and we set eight parallel attention heads (i.e., $\hat{H} = 8$). In the last step, the ReLU nonlinear activation is applied to the output of the MSA module, followed by a batch normalization (BN) layer to accelerate the training process of the model.

$$nf'_i = \mathrm{BN}\left(\delta_r\left(\sum_{j\in\mathcal{N}_i}\mathrm{MSA}(nf_i, nf_j)\right)\right) \tag{13}$$

Through the preceding procedure, we have effectively established a graph attention network. In contrast to the conventional graph convolution network, our graph attention network does not necessitate an additional label correlation matrix as input [43]. Furthermore, it can construct a relation matrix A independently to represent the precise correlation between labels.

**Graph Global Pooling**: The final prediction cannot be directly made using the output global descriptor $f_g$ that was achieved by training with the graph features, as it contains an excessive amount of noise that will impact the result. To achieve this objective, the *Global Multi-Head Self-Attention Pooling* (GMSAPool) is suggested. This method assigns identical qualifications to each node, allowing it to assess each node with a self-attention mechanism. Additionally, it aggregates information from various nodes in varying proportions to represent the global attention feature. The final global-attention feature $f_{ga}$ is calculated by adding the two GMSAPool outputs. The first $f_g^1$ and second $f_g^2$ inputs are accumulated separately as:

$$f_{ga} = \text{GMSAPool}(A, f_g^1) + \text{GMSAPool}(A, f_g^2 + \hat{Y}) \tag{14}$$

The Global Multi-Head Self-Attention Pooling (GMSAPool) mechanism enables the assessment of the importance of node feature retention through its method, which is defined in Algorithm 1. The algorithm calculates attention scores for nodes through self-attention, allowing the network to select important nodes for understanding the overall structure of the graph. In our calculations, the undirected graph $G(V, E)$ comprises $n$ nodes, with an adjacency matrix $V \in \mathbb{R}^{n\times n}$ and node feature set $E$. There exists an edge $E_{ij} = 1$ between nodes $i$ and $j$ where $E_i \in E$ and $E_i \in \mathbb{R}^{1\times d}$, with $d$ denoting the feature dimension of $E_i$. To represent the entire graph as a single feature vector, we introduce a virtual node $V_n$ to the graph $G$. The updated adjacency matrix becomes $V \in \mathbb{R}^{(n+1)\times(n+1)}$, where the entry $V_{ij} = 1$ if $i$ or $j$ corresponds to the virtual node, and $V_{ij} = 0$ otherwise. Consequently, all nodes become first-order neighbors of $V_n$. The embedding feature vector of the virtual node, denoted by $V_h \in \mathbb{R}^{1\times 2d}$, is initialized to better evaluate the importance of different nodes to the virtual node $V_n$ as:

$$V_h = \left(\frac{1}{n}\sum_{i=1}^{n} V_i\right) \parallel \left(\max_{i=1}^{n} V_i\right) \tag{15}$$

where $n$ is the number of nodes and $V_i$ is the embedding feature vector of node $i$. To effectively capture the potential relationships between node features, the feature vectors $V_h$ and $V_i$ are first concatenated, forming a joint representation that facilitates more informative interaction modeling. This concatenated vector is then passed through a self-attention mechanism, where a latent information matrix $Z \in \mathbb{R}^{1\times 3d}$ is applied to project the combined representation into a scalar value that reflects the attention score between the two nodes.

$$e_i = Z\left(V_h \parallel V_i\right)^T \tag{16}$$

Here, $e_i$ represents the importance of the neighbor node $V_i$ with respect to the virtual node $V_n$. A higher value of $e_i$ indicates a more substantial contribution of $V_i$ in influencing the representation of $V_n$. However, the raw importance values cannot be directly compared across different nodes due to their unbounded nature. Therefore, a ReLU activation function is applied first to ensure non-negativity, followed by a Softmax normalization to produce relative importance scores. This yields the final attention weight $\beta_i$, which quantifies the relative influence of the neighbor node $V_i$ on $V_n$, and is defined as:

$$\beta_i = \text{softmax}(e_i) = \frac{\exp(\text{ReLU}(e_i))}{\sum_{i\in n}\text{ReLU}(e_i)} \tag{17}$$

Upon calculating the attention weight $\beta_i$ for each node, this weight is multiplied by the respective embedding feature vector $V_i$. This process enables the model to adjust the significance of each node's contribution to the final pooled representation, highlighting more informative nodes and diminishing the influence of less relevant ones. The consolidated representation can be articulated as follows:

$$V_h' = \sum_{i\in n} \beta_i \odot V_i \tag{18}$$

As $V_h'$ cannot be directly used for the final prediction, a linear layer is adopted to map $V_h'$ to the output space:

$$V_h^{\text{out}} = \delta\left(\text{Linear}(V_h')\right) \tag{19}$$

where $V_h^{\text{out}} \in \mathbb{R}^{1\times c}$, and $c$ denotes the number of classes. A multi-head attention mechanism facilitates the network's ability to identify several distinctive and varied properties from node embeddings. Isolated attention units are inadequate for fully comprehending the intricate network connections present throughout graph topologies. We mitigate processing restrictions by employing $K$ separate attention heads that operate concurrently on identical graphs. Diverse representation subspaces might yield complementary information when used by the model. This network incorporates a global pooling layer at its conclusion that amalgamates the outputs of various attention heads. The network architecture minimizes processing demands and effectively captures essential properties, resulting in a comprehensive global representation.

$$V_h^{\text{out}} = \delta\left(\frac{1}{K}\sum_{k=1}^{K} V_h^k\right) \tag{20}$$

where $V_h^k$ is the output of the $k^{th}$ attention head. The overall algorithm for GMSAPool is described below.

**Algorithm 1** Proposed GMSAPool.

```
Require: Number of attention heads $h$, number of nodes $k$, and virtual node
    features $V_h^k$ for each head $k$
Ensure: $h$ graph representation vectors $V_h^{\text{out}}$
1: Initialize the virtual node using Equation (11)
2: Compute attention scores for neighboring nodes using Equation (12)
3: Update attention weights using Equation (13)
4: Update the virtual node using Equation (14)
5: Apply linear transformation using Equation (15)
6: Return $V_h^{\text{out}}$ using Equation (16)
```

### 3.4. Channel attention module

GAM effectively identifies spatial linkages but overlooks the direct connections between various channels, which may contain significant semantic information, such as texture, stroke pressure, or curvature. The DaGAM-Trans architecture, incorporating the Channel Attention Module (CAM) after the GAM, enhances feature representations by filtering the information. The CAM module improves model discrimination by autonomously prioritizing significant feature channels while diminishing noisy or less informative ones during operation. Following the integration of GAM with the pipeline, the design facilitates two attention mechanisms that merge graph-based spatial attention with channel-wise semantic attention. DaGAM-Trans's integrated capabilities enhance performance by comprehending both structural layouts and semantic elements in handwritten signatures.

The development of the proposed CAM comprises three fundamental steps. Initially, the high-level semantic feature map $f_R$ is condensed into a compact vector representation; thereafter, it is partitioned into numerous groups to enhance localized attention modeling. In the second stage, a GAT is utilized for each group to capture inter-group interactions precisely, allowing the model to discern semantic dependencies among various channel groups. In the last stage, the enhanced features are reconstructed into a full-resolution feature map labeled as $f_{R2}'$, which retains the exact dimensions as the original feature map $f_{R2}$. This guarantees smooth interaction with the following layers in the network. Fig. 4 depicts the comprehensive architecture and operating flow of the Channel Attention Module.

To enhance computational efficiency and minimize the parameter count, the channel dimensionality of the high-level semantic feature $f_R$ is initially decreased by a 1 × 1 convolutional layer. This procedure condenses the feature map while retaining critical information. Subsequently, a global pooling operation is executed on the diminished feature map, yielding a concise and informative representation represented as $f_{R3} \in \mathbb{R}^{C_2\times 1\times 1}$, where $C_2$ signifies the reduced channel count. To augment the channel-wise interactions, we implement group-wise convolution with convolution kernels of size 1 × 1 and a specified number of groups. This design facilitates more effective modeling of intra-group channel interdependence while little augmenting computational costs. The associated computational technique is delineated as follows:

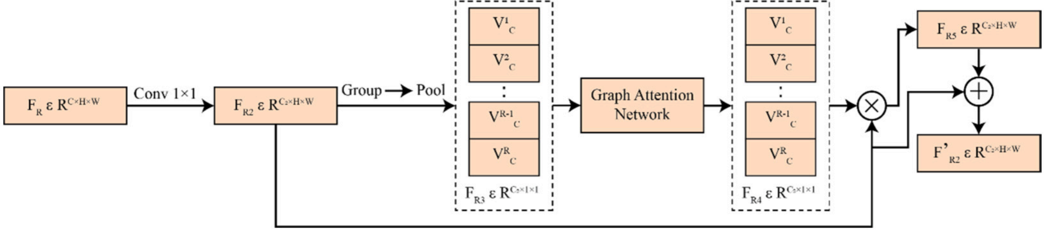

**Fig. 4.** Dataset samples from each selected dataset.

$$f_{R2} = \text{conv}^{1\times1}(f_R)$$
$$f_{R3} = \text{pool}(\text{group}(f_{R2})) \tag{21}$$

To create the node graph for channel attention modeling, the compact feature representation $f_{R3}$ is partitioned into $g$ groups, with each group $V_C^i$ including $C_2/g$ feature channels. The comprehensive representation is articulated as the concatenation of all groups $f_{R3} = \|(V_C^1, V_C^2, \ldots, V_C^g)$. This clustered configuration facilitates localized interaction learning within each channel subset. We model semantic connections among these groups by representing them as nodes in a fully linked graph, utilizing the GAT to learn the inter-group relationships. To attain semantic-relations-enhanced channel characteristics $f_{R4}$, we initially create a channel correlation graph, wherein an affinity matrix delineates the edge set $E_C$. The affinity edge between any two groups, $V_C^i$ and $V_C^j$, is calculated to assess their semantic similarity, and these affinity values constitute the weighted connections within the graph structure.

$$E_C(V_C^i, V_C^j) = \text{Transpose}(V_C^i) \cdot V_C^j \tag{22}$$

The equation above indicates that the affinity scores of edges increase as the correlation between channel feature maps increases. Consequently, it is feasible to generate a channel graph that is entirely connected. The features $f_{R4} \in \mathbb{R}^{C_2\times1\times1}$ can be obtained by employing GAT as:

$$f_{R4} = \text{GAT}(G_C) \tag{23}$$

Utilizing the Graph Attention Network (GAT), the interrelation of channel groups is acquired through a data-driven approach, enabling the network to assess the impact level of each channel feature map within the graph. Feature maps with more remarkable semantic similarity obtain elevated attention scores, therefore augmenting their influence in the consolidated representation. This targeted focus enhances the overall modeling of channel-wise correlations.

To reconstruct the final channel-enhanced feature map, the attention-enriched features $f_{R4}$, derived from the GAT, are multiplied element-wise with the original high-level feature map $f_{R2}$. This operation incorporates enhanced attention data while maintaining the spatial integrity of the original features, yielding a channel-adaptive representation ideal for classification. The acquired output is subsequently incorporated into the feature $f_{R2}$ using the same residual learning to produce the final output $f'_{R2}$:

$$f'_{R2} = f_{R4} \otimes f_{R2} + f_{R2} \tag{24}$$

The symbol ⊗ represents matrix multiplication. This operation is essential to the development of the proposed Channel Attention Module (CAM). Conventional GCN-based methods frequently encounter difficulties in directly calculating the correlation matrix from channel feature maps before training, hence constraining their capacity to predict inter-channel interdependence accurately. Consequently, these techniques fail to fully harness the representational capacity of graph convolution networks for discerning significant correlations among feature channels.

Conversely, our proposed CAM utilizes a Graph Attention Network (GAT), which autonomously and adaptively generates the correlation matrix throughout the training process. This allows the model to dynamically learn the semantic relationships among channel feature maps, leading to a more expressive and data-informed channel attention mechanism.

### 3.5. Classification module

The Classification Module serves as the final operational unit in the DaGAM-Trans architecture, processing output from attention modules to ascertain signature authenticity. The classification head processes the output feature tensor, which encompasses both spatial and semantic discriminative information following its traversal through transformer encoders utilizing GAM and CAM.

This component comprises three essential elements: flattening, normalizing, and a fully connected softmax layer. The dense classification necessitates a one-dimensional vector generated by flattening the attention-refined high-dimensional feature map. Following the flattening of activations, Batch Normalization is applied to facilitate quick training convergence and mitigate overfitting. The feature vector is processed through a fully connected layer with softmax activation to produce class probability outputs for evaluating signature authenticity.

The classification block operates as a similarity network by executing computations to ascertain feature distances within the integrated learning space. The replacement head module provides enhanced inter-class discrimination relative to ViT's default head, resulting in better accuracy across all benchmark datasets. The model integrates graph and channel attention methods to generate an input with optimal expressive capacity, hence facilitating effective decision-making in scenarios involving expert counterfeits or diverse signature language kinds.

## 4. Results and discussion

In this section, we provide a detailed analysis of the results from our extensive testing and training in the proposed model for signature verification. We review essential performance metrics, including Accuracy and Precision. Recall the F-1 score, False Rejection Rate (FRR), False Acceptance Rate (FAR), and Equal Error Rate (EER). [11] We also examine execution and computing time to offer a complete understanding of the models' performance. This thorough evaluation provides insightful conclusions on the practical effectiveness and feasibility of our proposed approach for signature verification.

### 4.1. Experimental setup

The efficacy and resilience of the DaGAM-Trans model were evaluated using the BHSig260, CEDAR, UTSig, and SigComp2011 (Offline) signature datasets, all of which are accessible for research purposes. The datasets comprise materials in many languages, featuring distinct handwriting styles and varying degrees of forgery difficulty, to facilitate comprehensive testing of dependency and independence in signature verification processes. The DaGAM-Trans model was executed utilizing an NVIDIA GeForce RTX 2060 GPU with 6 GB of GDDR6 RAM for training. The training commenced with a learning rate of 0.001, which decreased by a factor of 0.7 every five epochs while maintaining momentum of 0.7 until 350 epochs were completed. The model was evaluated using three distinct split ratios of 50-50, 70-30, and 80-20 to assess its flexibility

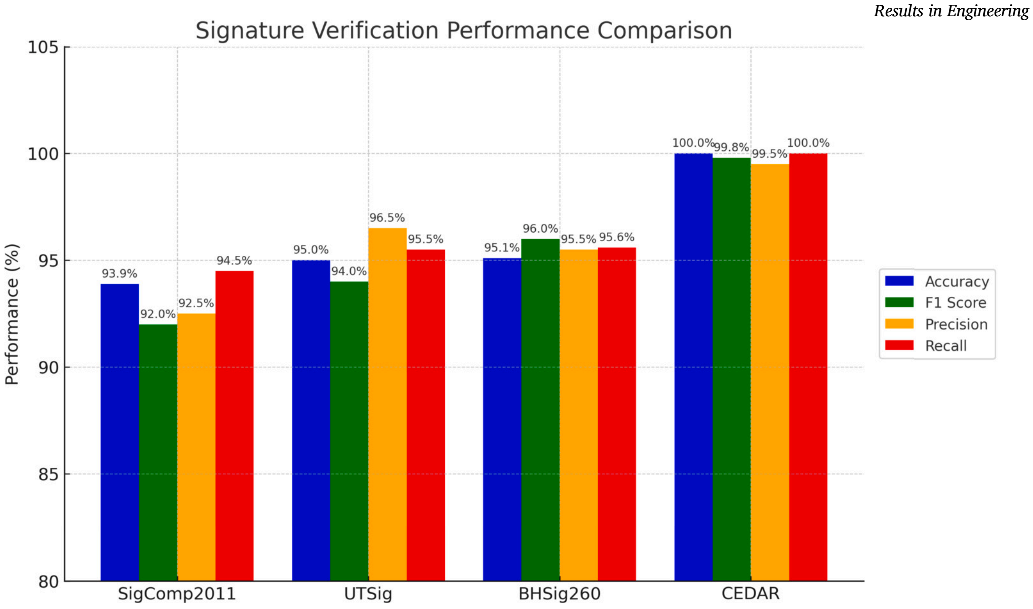

**Fig. 5.** Accuracy, F-1, Precision, and Recall values of Proposed Methodology.

**Table 4**
Performance Measures with Different Distributions of Datasets.

| Data Split | Dataset | FRR | FAR | EER |
|---|---|---|---|---|
| 50/50 | SigComp2011 (Offline) | 1.19 | 0.80 | 14.7 |
| | UTSig | 0.45 | 1.41 | 17.3 |
| | CEDAR | 0.23 | 0.63 | 9.6 |
| | BHSig260 | 0.95 | 1.54 | 12.0 |
| 70/30 | SigComp2011 (Offline) | 0.23 | 0.81 | 15.5 |
| | UTSig | 0.35 | 1.07 | 12.0 |
| | CEDAR | 0.21 | 0.10 | 10.0 |
| | BHSig260 | 0.59 | 1.03 | 14.8 |
| 80/20 | SigComp2011 (Offline) | 0.16 | 0.13 | 14.9 |
| | UTSig | 0.17 | 0.12 | 11.2 |
| | CEDAR | 0.001 | 0.09 | 9.5 |
| | BHSig260 | 0.54 | 0.63 | 15.9 |

**Table 5**
Evaluation of DaGAM-Trans in Terms of Equal Error Rate (EER), Execution Time (ET), and Computation Time (CT) Across Four Signature Verification Datasets.

| Dataset | EER | ET (sec) | CT (sec) |
|---|---|---|---|
| SigComp2011 (Offline) | 14.9 | 300 | 150 |
| UTSig | 11.2 | 500 | 300 |
| CEDAR | 9.5 | 300 | 150 |
| BHSig260 | 15.9 | 723.23 | 650.5 |

and consistency across varying training and testing proportions. A specific 70-15-15 division was employed to evaluate performance during hyperparameter optimization. The model underwent several data augmentation approaches, including random rotations, zooming, shifting, and flipping, to mitigate overfitting and improve generalization. The evaluation of model performance employed standard biometric metrics to quantify Accuracy, Precision, Recall, F1-score, False Acceptance Rate (FAR), False Rejection Rate (FRR), and Equal Error Rate (EER).

### 4.2. Experimental results and analysis

Table 4 assesses the False Rejection Rate (FRR), False Acceptance Rate (FAR), and Equal Error Rate (EER) of four signature verification datasets SigComp2011 (Offline), UTSig, CEDAR, and BHSig260—using three train-test splits: 50/50, 70/30, and 80/20. The findings indicate that the verification accuracy is enhanced by an increase in the training ratio, which results in a reduction in both FRR and FAR across all datasets. CEDAR is the most dependable dataset, as it consistently obtains the lowest EER (9.6% at 50/50, 10% at 70/30, and 9.5% at 80/20). In contrast, UTSig and BHSig260 demonstrate higher EER values, with UTSig reaching 17.3% at 50/50, suggesting a more significant challenge in distinguishing forgeries. The EER of SigComp2011 (Offline) is also high (14.7% at 50/50), indicating that its forgeries are difficult to detect, even with heightened training. Although all datasets benefit from the expansion of training data, BHSig260 and SigComp2011 continue to present challenges because of high intra-class variability and skilled forgeries. This underscores the significance of model robustness and dataset selection in signature verification applications.

Fig. 5 presents a comparative performance assessment of signature verification models across four datasets—SigComp2011, UTSig, BHSig260, and CEDAR—evaluated on Accuracy, F1 Score, Precision, and Recall. CEDAR demonstrates superior performance across all parameters, approaching 100% in accuracy, precision, recall, and F1-score, signifying its appropriateness for dependable signature authentication. BHSig260 exhibits an accuracy of 95%, indicating that the dataset comprises well-delineated authentic and counterfeit signatures that the machine can classify proficiently. The suggested model achieves an accuracy of 95% for UTSig, signifying enhanced confidence in predicting authentic signatures. SigComp2011 exhibits accuracy, F1 score, precision, and recall values between 92% and 94%. The findings indicate that dataset attributes considerably influence verification accuracy, with CEDAR offering the most stable and dependable benchmark, while SigComp2011 necessitates more advanced classification methods to enhance performance.

Table 5 indicates the Equal Error Rate (EER) and efficiency of the model in terms of Execution Time (ET) and Computing Time (CT). EER measures model accuracy when the False Acceptance Rate (FAR) and False Rejection Rate (FRR) are identical, ET measures model execution time, and CT measures computing cost when the image is converted to the graph. With an ET of 300 seconds and a CT of 150 seconds, CEDAR is the most efficient dataset in terms of performance and calculation, with the lowest EER (9.5%) and best verification accuracy. UTSig has an EER of 11.2% but takes 500 sec to evaluate and 300 sec to compute, indicating a more complicated dataset with higher intra-class variability. SigComp2011 (Offline) has an EER of 14.9%, identical evaluation and calculation durations as CEDAR, but poorer accuracy, making its forgeries harder to spot. The greatest EER (15.9%), longest evaluation (723.23 sec), and calculation time (650.5 sec) indicate the complexity

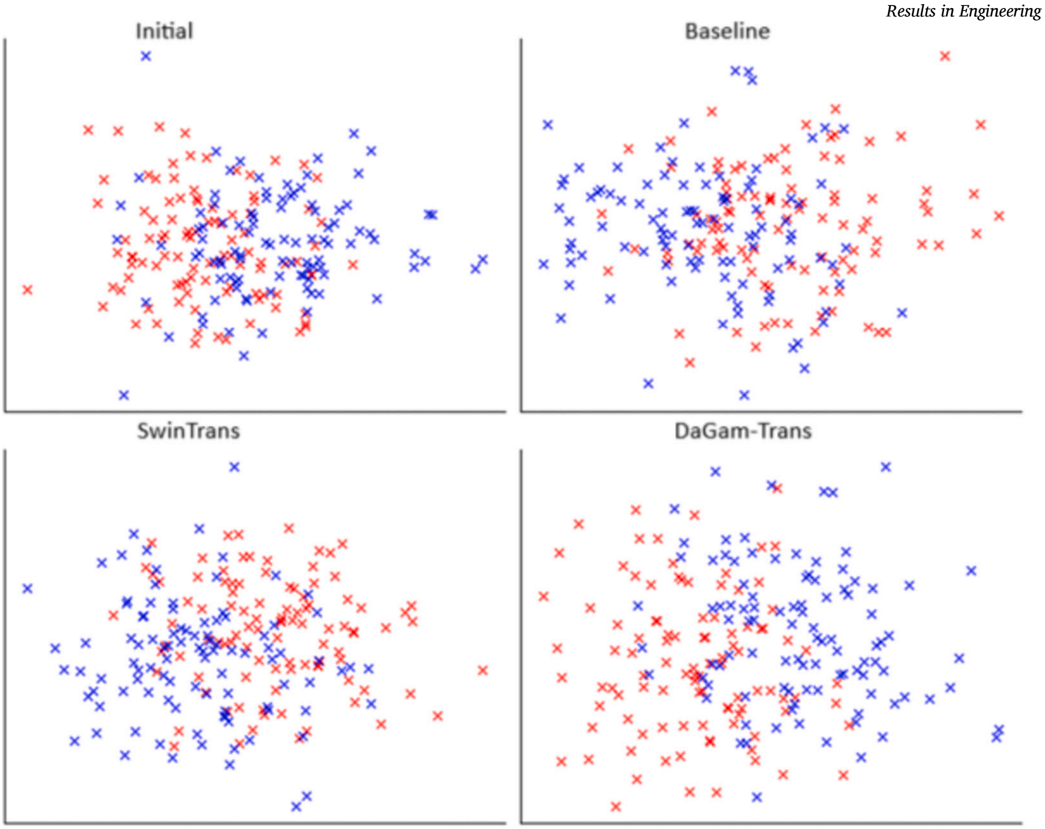

**Fig. 6.** Visualization of features extracted from different models.

**Table 6**
Execution Time (ET) and Computation Time (CT) Comparison of DaGAM-Trans with Baseline Models.

| Model | Dataset | ET (sec) | CT (sec) | Accuracy (%) |
|---|---|---|---|---|
| CNN | CEDAR | 90 | 60 | 94.1 |
| | UTSig | 160 | 100 | 89.5 |
| | BHSig260 | 210 | 180 | 90.7 |
| | SigComp2011 | 120 | 90 | 91.2 |
| ViT | CEDAR | 220 | 100 | 97.1 |
| | UTSig | 330 | 220 | 84.1 |
| | BHSig260 | 410 | 290 | 89.0 |
| | SigComp2011 | 260 | 140 | 89.0 |
| SwinTrans | CEDAR | 270 | 180 | 98.5 |
| | UTSig | 400 | 250 | 86.7 |
| | BHSig260 | 510 | 320 | 94.6 |
| | SigComp2011 | 290 | 180 | 91.7 |
| **DaGAM-Trans** | CEDAR | **300** | **150** | **100.0** |
| | UTSig | **500** | **300** | **95.0** |
| | BHSig260 | **723.2** | **650.5** | **95.1** |
| | SigComp2011 | **300** | **150** | **93.9** |

**Table 7**
Accuracy Comparison of DaGAM-Trans Across Four Signature Verification Datasets Using Different Activation Functions.

| Activation Function | SigComp2011 (Offline) | UTSig | BHSig260 | CEDAR |
|---|---|---|---|---|
| ReLU | 92.3 | 97.3 | 95.0 | 100.0 |
| Swish | 91.7 | 86.7 | 94.6 | 98.5 |
| Tanh | 89.0 | 84.1 | 93.2 | 97.1 |
| Sigmoid | 70.8 | 80.6 | 90.7 | 80.6 |

and processing demands of BHSig260, the most difficult dataset to verify.

Table 6 illustrates a comparative view of computation and execution time across DaGAM-Trans and baseline models. While DaGAM-Trans incurs higher compute cost—especially on large datasets like BHSig260—it significantly outperforms others in accuracy, particularly in difficult cross-script scenarios. For example, although DaGAM-Trans requires 650.5 seconds of computation time on BHSig260, it achieves a 5–10% accuracy gain over the closest baseline. This trade-off is justifiable for high-stakes biometric applications such as legal or forensic verification, where model reliability outweighs marginal increases in computation. Nevertheless, future work could explore model compression and hardware optimization to deploy DaGAM-Trans in real-time or embedded systems.

Table 7 shows the accuracy of signature verification models in four datasets across four activation functions: ReLU, Swish, Tanh, and Sigmoid. ReLU is the best activation function for signature verification, with 100% accuracy on CEDAR, 97.3% on UTSig, 95.0% on BHSig260, and 92.3% on SigComp2011. Swish follows with competitive BHSig260 (94.6%) and CEDAR (98.5%) accuracy. However, its performance on UTSig (86.7%) and SigComp2011 (91.7%) declines significantly, demonstrating dataset dependence. Tanh is slightly less accurate than Swish on all datasets, especially UTSig (84.1%) and SigComp2011 (89.0%), suggesting it may not be as helpful for signature classification. Sigmoid provides the worst accuracy on all datasets, including SigComp2011 (70.8%) and CEDAR (80.6%). This highlights its inadequacy in identifying forgeries and sophisticated signature variants. These findings imply that ReLU is the best activation function for signature verification models due to its accuracy and robustness across datasets. Sigmoid's reduced accuracy is likely owing to its vanishing gradient problem, which impairs deep learning model convergence and efficacy. However, Swish and Tanh perform moderately.

Fig. 6 presents a visualization of feature distributions extracted from different models in the signature verification process. The Initial plot shows features taken from an untrained model with pre-trained parameters. The two classes have highly mixed distributions and significant

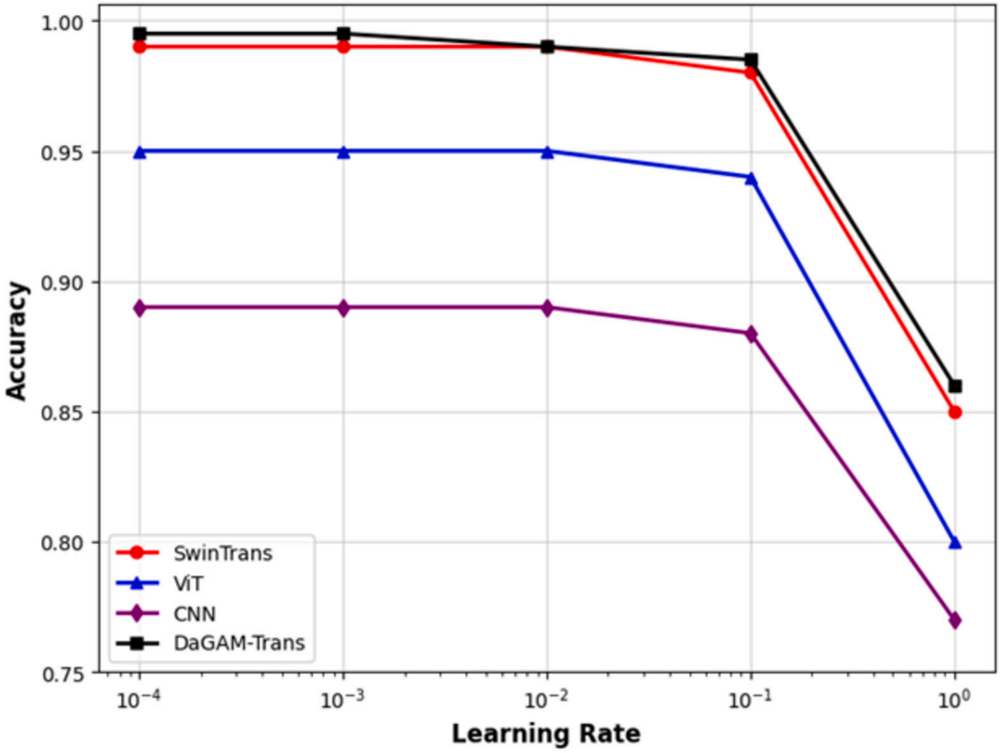

**Fig. 7.** Impact of varying learning rates on the accuracy of signature verification models (ViT, CNN, SwinTrans, and DaGAM-Trans).

**Table 8**
Cross-Language Signature Verification Accuracy of DaGAM-Trans Compared to Baseline Models (ViT, CNN, SwinTrans) Trained on CEDAR and Tested on Other Datasets.

| Test Dataset (Train: CEDAR) | ViT | CNN | SwinTrans | DaGAM-Trans |
|---|---|---|---|---|
| SigComp2011 (Offline) | 50.2 | 59.4 | 55.8 | 65.7 |
| UTSig | 64.7 | 56.4 | 67.8 | 80.4 |
| BHSig260 | 79.9 | 69.4 | 65.3 | 84.4 |

overlap, indicating that the model has not yet acquired meaningful feature representations. The Baseline plot matches the features extracted from the original Vision Transformer (ViT) model, where some structural separation emerges. Still, the overlap remains substantial, suggesting that the ViT model alone cannot distinguish genuine and forged signatures. The SwinTrans figure displays features derived from the original Swin Transformer model, which improves feature clustering relative to the baseline due to better class distributions, though overlap continues. Finally, the DaGAM-Trans plot shows the suggested model's feature distribution, showing the best feature separability and more apparent class distinction. This indicates that the DaGAM-Trans model improves feature extraction, making it better for complex classification tasks like signature verification. The results demonstrate the continual improvement of learned feature representations and the DaGAM-Trans model's enhanced ability to recognize genuine and fraudulent signatures.

Table 8 shows cross-language signature verification results for models trained on CEDAR and tested on SigComp2011, UTSig, and BHSig260 to assess generalization. DaGAM-Trans beats all models on all datasets, including SigComp2011 (65.7%), UTSig (80.4%), and BHSig260 (84.4%), showing its adaptability to unknown signature styles. Although SwinTrans generalizes better than ViT and CNN, it struggles with complicated cross-dataset adaptation. It performs best on UTSig (67.8%) but behind DaGAM-Trans. ViT performs worst on SigComp2011 (50.2%) and UTSig (64.7%), demonstrating its feature extraction is less effective with various signature distributions. CNN is better than ViT on SigComp2011 (59.4%) but worse on UTSig (56.4%) and BHSig260 (69.4%), demonstrating its cross-dataset variability limits. The results show that DaGAM-Trans is the best model for cross-language verification, with higher generalization and robustness across signature datasets. At the same time, ViT and CNN suffer from accuracy decreases in unexplored signature distributions.

Fig. 7 shows how the learning rate affects SwinTrans, ViT, CNN, and DaGAM-Trans signature verification accuracy. Even at lower learning rates ($10^{-4}$ to $10^{-3}$), all models remain stable and accurate. SwinTrans and DaGAM-Trans achieve about 100% accuracy, while ViT and CNN have slightly lower accuracy. As the learning rate increases to $10^{-2}$, accuracy decreases slightly, with CNN being most sensitive to fluctuations. All models lose accuracy at higher learning rates ($10^{-1}$ and $10^{0}$), with CNN performing worst and dropping below 80%. ViT also declines but remains steady. SwinTrans and DaGAM-Trans show greater robustness through higher accuracy at modest learning rates before a steep fall at $10^{0}$. Results show that a high learning rate causes instability and impacts model performance. The optimal learning rate for transformer-based architectures like SwinTrans and DaGAM-Trans is $10^{-4}$, ensuring stability and accuracy.

Fig. 8 shows how learning rate decay approaches affect SwinTrans, ViT, CNN, and DaGAM-Trans signature verification accuracy. The x-axis shows learning rate decay methods—None, Step, Exponential, Cosine Annealing, and Linear—and the y-axis model accuracy. Compared to no decay, learning rate decay improves accuracy for all models. Exponential and Cosine Annealing boost performance the most, especially for the DaGAM-Trans model, which reaches peak accuracy with exponential decay. Decay helps SwinTrans achieve its best accuracy under Exponential and Cosine Annealing, while ViT and CNN improve moderately but consistently. The linear decay strategy affects accuracy marginally, suggesting that dynamically altering the learning rate may be more beneficial than gradually decreasing it. Cosine Annealing improves accuracy, especially for DaGAM-Trans and SwinTrans, optimizing model training stability. This suggests that learning rate decay improves verification accuracy, with Exponential and Cosine Annealing being the best options for advanced transformer-based models like DaGAM-Trans and SwinTrans.

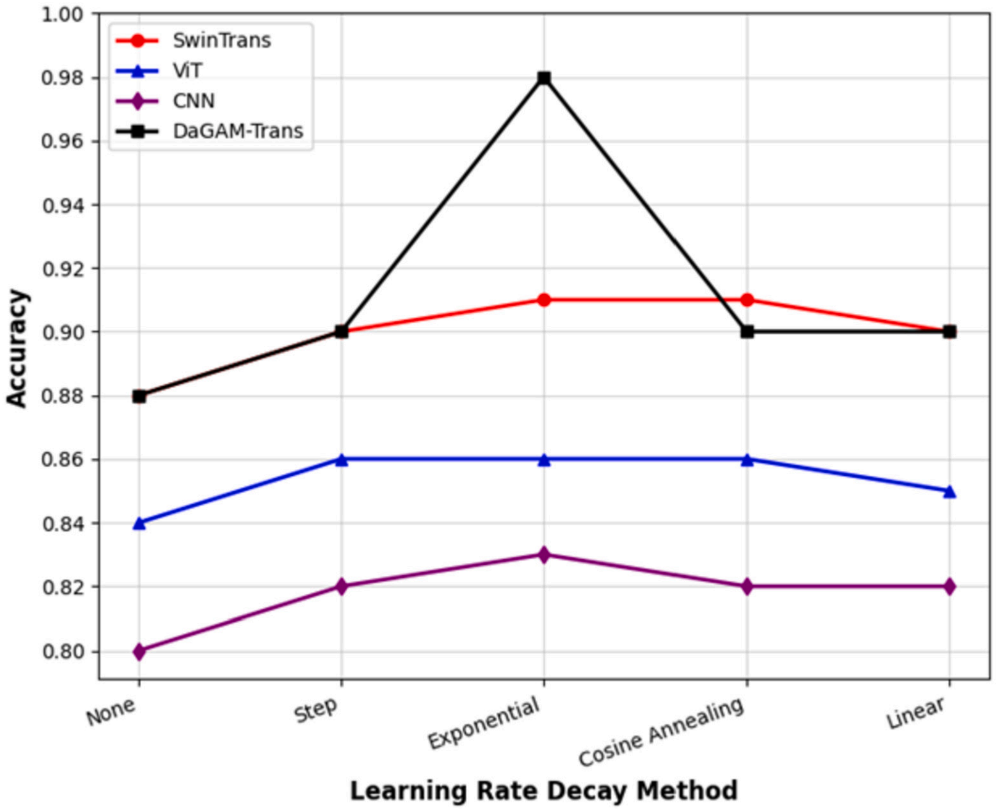

**Fig. 8.** Effect of different learning rate decay strategies—None, Step, Exponential, Cosine Annealing, and Linear—on the accuracy of signature verification models.

**Table 9**
Ablation Study: Impact of GAM, CAM, and GMSAPool on Signature Verification Performance.

| Dataset | Model Variant | Accuracy (%) | FAR (%) | EER (%) |
|---|---|---|---|---|
| CEDAR | Baseline (Transformer only) | 94.1 | 2.9 | 13.7 |
| | + GAM | 96.7 | 1.8 | 11.3 |
| | + GAM + CAM | 98.6 | 0.8 | 10.1 |
| | + GAM + CAM + GMSAPool (Full Model) | **100.0** | **0.001** | **9.5** |
| BHSig260 | Baseline (Transformer only) | 89.6 | 3.8 | 18.9 |
| | + GAM | 91.9 | 2.2 | 16.0 |
| | + GAM + CAM | 93.7 | 1.3 | 15.2 |
| | + GAM + CAM + GMSAPool (Full Model) | **95.1** | **0.54** | **15.9** |
| SigComp2011 | Baseline (Transformer only) | 90.4 | 3.1 | 15.7 |
| | + GAM | 92.2 | 1.9 | 14.3 |
| | + GAM + CAM | 93.1 | 1.1 | 13.2 |
| | + GAM + CAM + GMSAPool (Full Model) | **93.9** | **0.16** | **14.9** |
| UTSig | Baseline (Transformer only) | 91.8 | 2.7 | 13.9 |
| | + GAM | 93.6 | 1.6 | 12.7 |
| | + GAM + CAM | 94.3 | 0.9 | 11.9 |
| | + GAM + CAM + GMSAPool (Full Model) | **95.0** | **0.17** | **11.2** |

### 4.3. Ablation study

To evaluate the individual contributions of the core components in DaGAM-Trans, we performed an ablation study by progressively adding the Graph Attention Module (GAM), Channel Attention Module (CAM), and GMSAPool to a baseline transformer architecture. The results, summarized in Table 9, reflect performance on both the CEDAR and BHSig260 datasets. Introducing GAM to the baseline model significantly improved accuracy by capturing essential spatial dependencies between signature components, which also resulted in a notable reduction in Equal Error Rate (EER). The addition of CAM further enhanced the model's ability to focus on semantically rich feature channels, leading to improvements in classification precision and a decrease in the False Acceptance Rate (FAR). Finally, the integration of GMSAPool provided an effective global pooling mechanism that preserved discriminative information while reducing redundancy. This resulted in the best overall performance across all metrics.

### 4.4. Comparison with previous work

Table 10 presents a comprehensive comparison of DaGAM-Trans with various offline signature verification methods utilizing the BHSig260, CEDAR, SigComp2011 Offline, and UTSig datasets. ST-CNN achieves an accuracy of 99.34 percent on BHSig260 but exhibits compromised security, with an equal error rate of around 28 percent alongside a 28 percent false acceptance rate. DaGAM-Trans demonstrates a

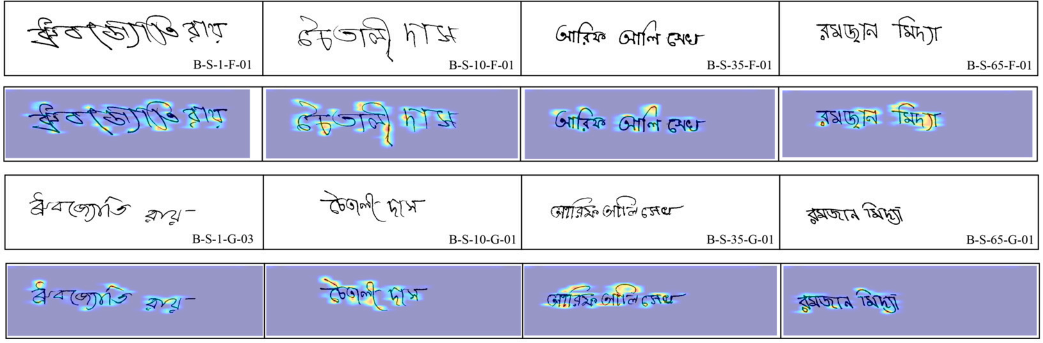

**Fig. 9.** Grad-CAM heatmaps on BHSig260 Bengali signatures showing attention differences between genuine and forged samples.

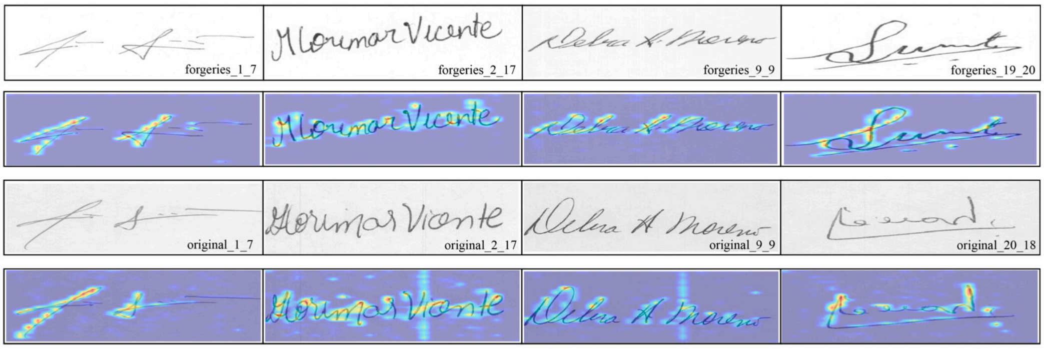

**Fig. 10.** Visual explanation of model focus using heatmaps on CEDAR signatures for forgery detection.

commendable performance, achieving 95.1 percent accuracy, including minor erroneous acceptance at 0.54 percent and false rejection at 0.63 percent. The equal error rate exhibits minor increases due to significant intra-class heterogeneity within the sample. In the CEDAR dataset, DaGAM-Trans achieves an accuracy of 100 percent, exhibiting remarkable precision with a false acceptance rate of 0.001 percent and a false rejection rate of 0.09 percent. The system exhibits efficacy in consistently processing neatly aligned signature inputs. Numerous deep learning models attain superior accuracy on the SigComp2011 Offline dataset; nonetheless, they fail to provide comprehensive evaluation measures. The DaGAM-Trans model exhibits a remarkable accuracy rate of 93.9 percent, alongside a low false acceptance rate of 0.16 percent and a false rejection rate of 0.13 percent, indicating its robustness against intricate signature forgeries. The UTSig dataset exhibits DaGAM-Trans's 95.0 percent accuracy, rendering it superior to LSTM-HOG due to its enhanced performance in erroneous acceptance and rejection rates. The model attains performance comparable to that of advanced models utilizing GNN-SIFT while operating with graph-based structures. Data from diverse testing scenarios indicates that DaGAM-Trans exhibits effective and interoperable signature authentication capabilities, evidenced by its consistent error measurement performance across multiple databases.

**Table 10**
Comparative Analysis of DaGAM-Trans with State-of-the-Art Signature Verification Methods Across Four Benchmark Datasets.

| Method | ACC (%) | FAR (%) | FRR (%) | EER (%) |
|---|---|---|---|---|
| **BHSig260 Dataset** | | | | |
| ST-CNN [13] | 99.34 | 14.18 | 13.66 | 27.97 |
| DL-GA [16] | 97.73 | 2.50 | 3.20 | 2.35 |
| SigGCN [18] | 95.00 | 3.95 | 4.06 | 4.00 |
| DaGAM-Trans | 95.10 | 0.54 | 0.63 | 15.90 |
| **CEDAR Dataset** | | | | |
| FHDNN [22] | 100.00 | 0.00 | 0.00 | 0.00 |
| DNN-LDA-FFT [23] | 95.00 | 14.41 | 13.42 | 13.43 |
| TL-InceptionV3 [24] | 99.10 | 1.03 | 0.74 | - |
| DaGAM-Trans | 100.00 | 0.001 | 0.09 | 9.50 |
| **SigComp2011 (Offline) Dataset** | | | | |
| MobileNetV2-FS [26] | 97.70 | - | - | 0.0057 |
| DenseNet201-ED [27] | 99.44 | - | - | - |
| DNN-25L [28] | 100.00 | - | - | - |
| DaGAM-Trans | 93.90 | 0.16 | 0.13 | 14.90 |
| **UTSig Dataset** | | | | |
| LSTM-HOG [30] | 92.40 | 10.12 | 12.68 | 11.40 |
| OctConv-Net [33] | 82.85 | - | - | - |
| GNN-SIFT [34] | 100.00 | 0.00 | 0.007 | 0.092 |
| DaGAM-Trans | 95.00 | 0.17 | 0.12 | 11.20 |

### 4.5. Interpretability of network

The proposed model generates attention maps to validate its predictions. Figs. 9-12 illustrate the heatmaps of both original and fabricated signature photos across all chosen datasets. These heatmaps are vital for model interpretability since they disclose the spatial areas that significantly influence the categorization decision. In authentic signatures, the focus is typically directed toward well-structured characters and stable stroke junctions that are consistently replicated across occurrences. The model primarily examines abnormalities, tremors, unnatural pen pressure, and inconsistencies in stroke continuity in counterfeit samples, as these are essential markers of forgery. By graphically pinpointing the model's focus, these heatmaps not only bolster the credibility of the DaGAM-Trans architecture but also provide forensic insights for human experts to authenticate AI-generated choices. Moreover, in writer-independent contexts, these visualizations elucidate the model's generalization capacity across various writing styles and scripts.

These heatmaps are essential for model interpretability since they disclose the spatial areas that significantly influence the categorization decision. In authentic signatures, the focus is typically directed toward well-formed characters and stable stroke junctions that are consistently replicated across instances. The model for faked samples emphasizes abnormalities, tremors, unnatural pen pressure, and inconsistencies in stroke continuity, all of which are essential signs of forgery. By graphi-

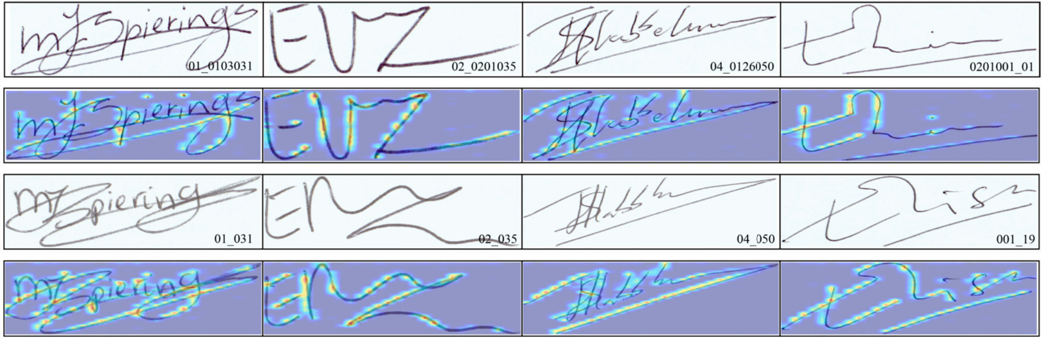

**Fig. 11.** Signature attention localization from SigComp2011 dataset highlighting critical stroke areas.

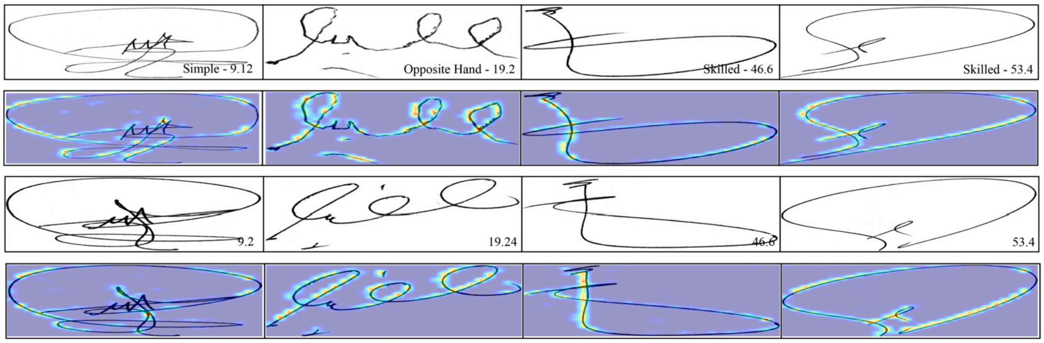

**Fig. 12.** UTSig signature heatmaps illustrating the model's ability to distinguish subtle forgery variations.

cally pinpointing the model's focus, these heatmaps not only bolster the credibility of the DaGAM-Trans architecture but also provide forensic insights for human experts to authenticate AI-generated choices. Moreover, in writer-independent contexts, these visualizations elucidate the model's generalization capacity across many writing styles and scripts.

Notwithstanding their utility, heatmap-based visualizations possess specific restrictions. Initially, Grad-CAM offers merely a rudimentary localization of attention and may fail to accurately delineate stroke-level distinctions, especially in intricate or highly patterned signatures. Secondly, the interpretability of attention maps can fluctuate considerably based on image quality, contrast, and background noise. In certain instances, the focus may seem dispersed or misdirected because of overlapping strokes or scanner artifacts. Furthermore, these heatmaps illustrate model-specific attention and may differ among architectures or training epochs, complicating the establishment of standardized forensic criteria. Consequently, although heatmaps are a useful interpretability instrument, they must be supplemented by quantitative analysis and expert assessment for critical signature verification jobs.

Despite the usefulness of attention heatmaps for visual interpretability, we recognize that Grad-CAM provides only a coarse approximation of the regions influencing the model's decision. It does not explicitly capture fine-grained, stroke-level variations that are often critical in identifying skilled forgeries. Subtle differences in curvature, pen lift, or stroke smoothness may remain invisible in such overlays. Moreover, attention maps can be significantly affected by variations in image quality, such as background noise, low contrast, or scanning artifacts. These issues can lead to misaligned or dispersed focus, thereby weakening the reliability of the interpretability outputs. In our current model, we do not yet validate the alignment of model attention with human-expert-annotated critical regions, which would be essential for forensic trust in real-world scenarios.

To address these limitations in future work, we plan to explore fine-grained visual explanation techniques such as:

- Stroke-level relevance maps using skeletonized signature paths or contour overlays.
- Multi-layer self-attention visualization across transformer blocks.
- Attribution methods such as SHAP or LIME tailored for spatial image input.
- Benchmarking model-attention regions against human-labeled regions of interest.

Additionally, preprocessing methods such as contrast enhancement and denoising filters may be integrated to stabilize attention output under degraded input conditions. These enhancements will improve both interpretability and user trust in real-world applications like legal authentication or forensic investigations.

## 5. Conclusion

The study describes DaGAM-Trans, a Transformer that is based on a Dual Attention Graph Attention Module, as a novel and highly effective method for the detection of offline signature forgeries. Through the integration of Transformer-based architectures with Graph Attention Networks (GAT), the model effectively incorporates both local and global dependencies in signature structures, thereby improving the classification of genuine and forged signatures. Dynamic feature refinement is facilitated by the introduction of the Graph Attention Module (GAM) and Channel Attention Module (CAM), which enhances robustness against competent forgeries and intra-class variations. The Graph Multi-Scale Adaptive Pooling (GMSAPool) layer, in addition, optimizes feature aggregation to ensure efficient and discriminative representation learning. The superiority of DaGAM-Trans over state-of-the-art models, such as CNN-based and Transformer-based architectures, is demonstrated by extensive evaluations of four publicly available datasets: SigComp2011, BHSig260, CEDAR, and UTSig. The model excels in the detection of competent forgeries and cross-language verification, achieving a higher accuracy, a lower False Acceptance Rate (FAR), a lower False Rejection Rate (FRR), and an improved Equal Error Rate (EER). DaGAM-Trans is a highly dependable solution for forensic and biometric authentication applications, as the results have confirmed that the integration of graph attention mechanisms with self-attention architectures significantly improves signature verification.

## CRediT authorship contribution statement

**Sara Tehsin:** Writing – original draft, Resources, Methodology, Formal analysis, Conceptualization. **Ali Hassan:** Writing – review & editing, Validation, Supervision, Resources, Data curation. **Farhan Riaz:** Writing – review & editing, Visualization, Supervision, Project administration. **Inzamam Mashood Nasir:** Writing – original draft, Project administration, Methodology, Funding acquisition, Conceptualization.

## Funding

This research did not receive any external or internal funding.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors would like to acknowledge Department of Computer and Software Engineering, National University of Sciences and Technology for their support.

## Data availability

Data will be made available on request.

## References

[1] I. Kathuria, Handwritten signatures dataset, https://www.kaggle.com/datasets/ishanikathuria/handwritten-signature-datasets, 2023. (Accessed 17 July 2023).

[2] I.M. Nasir, S. Tehsin, R. Damaševičius, A. Zielonka, M. Woźniak, Explainable cubic attention-based autoencoder for skin cancer classification, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2024, pp. 124–134.

[3] S. Tehsin, I.M. Nasir, R. Damaševičius, Explainability of disease classification from medical images using xgrad-cam, in: International Conference on Advanced Research in Technologies, Information, Innovation and Sustainability, Springer, 2024, pp. 221–236.

[4] S. Tehsin, I.M. Nasir, R. Damaševičius, Interpreting cnn for brain tumor classification using xgrad-cam, in: International Conference on Advanced Research in Technologies, Information, Innovation and Sustainability, Springer, 2024, pp. 282–296.

[5] S.N. Yousafzai, H. Shahbaz, A. Ali, A. Qamar, I.M. Nasir, S. Tehsin, R. Damaševičius, X-news dataset for online news categorization, Int. J. Intell. Comput. Cybern. 17 (4) (2024) 737–758.

[6] S.N. Yousafzai, I.M. Nasir, S. Tehsin, D.S. Malik, I. Keshta, N.L. Fitriyani, Y. Gu, M. Syafrudin, Multi-stage neural network-based ensemble learning approach for wheat leaf disease classification, IEEE Access (2025).

[7] S.N. Yousafzai, I.M. Nasir, S. Tehsin, N.L. Fitriyani, M. Syafrudin, Fltrans-net: transformer-based feature learning network for wheat head detection, Comput. Electron. Agric. 229 (2025) 109706.

[8] S. Tehsin, I.M. Nasir, R. Damaševičius, Gatransformer: a graph attention network-based transformer model to generate explainable attentions for brain tumor detection, Algorithms 18 (2) (2025) 89.

[9] I.M. Nasir, S. Tehsin, R. Damaševičius, R. Maskeliūnas, Integrating explanations into cnns by adopting spiking attention block for skin cancer detection, Algorithms 17 (12) (2024) 557.

[10] D.S. Malik, T. Shah, S. Tehsin, I.M. Nasir, N.L. Fitriyani, M. Syafrudin, Block cipher nonlinear component generation via hybrid pseudo-random binary sequence for image encryption, Mathematics 12 (15) (2024) 2302.

[11] S. Tehsin, A. Hassan, F. Riaz, I.M. Nasir, N.L. Fitriyani, M. Syafrudin, Enhancing signature verification using triplet Siamese similarity networks in digital documents, Mathematics 12 (17) (2024) 2757.

[12] I.M. Nasir, M.A. Alrasheedi, N.A. Alreshidi, Mfan: multi-feature attention network for breast cancer classification, Mathematics 12 (23) (2024) 3639.

[13] P.D. Hung, P.S. Bach, B.T. Vinh, N.H. Tien, V.T. Diep, Offline handwritten signature forgery verification using deep learning methods, in: Smart Trends in Computing and Communications: Proceedings of SmartCom 2022, Springer, 2022, pp. 75–84.

[14] A. Jain, S.K. Singh, K.P. Singh, Signature verification using geometrical features and artificial neural network classifier, Neural Comput. Appl. 33 (12) (2021) 6999–7010.

[15] F.R. PP, S. Tehsin, A framework for breast cancer classification with deep features and modified grey wolf optimization, Mathematics 13 (8) (2025) 1236.

[16] A.M.O. Al-Azzani, A.M.Q. Musleh, Offline signature verification using deep learning and genetic algorithm, 2024.

[17] P. Chaturvedi, A. Jain, Feature Ensemble Based Method for Verification of Offline Signature Images, 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), vol. 1, IEEE, 2022, pp. 710–714.

[18] C. Ren, J. Zhang, H. Wang, S. Shen, Vision graph convolutional network for writer-independent offline signature verification, in: 2023 International Joint Conference on Neural Networks (IJCNN), IEEE, 2023, pp. 1–7.

[19] G. Culqui-Culqui, S. Sanchez-Gordon, M. Hernández-Álvarez, An algorithm for classifying handwritten signatures using convolutional networks, IEEE Latin Am. Trans. 20 (3) (2021) 465–473.

[20] H. Wu, Y. Qiao, X. Luo, A fine-grained regularization scheme for nonnegative latent factorization of high-dimensional and incomplete tensors, IEEE Trans. Serv. Comput. (2024).

[21] H. Wu, J. Mi, A Cauchy loss-incorporated nonnegative latent factorization of tensors model for spatiotemporal traffic data recovery, Neurocomputing 626 (2025) 129575.

[22] Z. Hashim, H. Mohsin, A. Alkhayyat, Signature verification based on proposed fast hyper deep neural network, IAES Int. J. Artif. Intell. 13 (1) (2024) 961–973.

[23] F.M. Alsuhimat, F.S. Mohamad, A hybrid method of feature extraction for signatures verification using cnn and hog a multi-classification approach, IEEE Access 11 (2023) 21873–21882.

[24] S. Singh, S. Chandra, A.R. Verma, Enhancing offline signature verification via transfer learning and deep neural networks, Augment. Hum. Res. 9 (1) (2024) 4.

[25] Z. Hashim, H. Mohsin, A. Alkhayyat, Handwritten signature identification based on hybrid features and machine learning algorithms, Iraqi J. Comput. Commun. Control. Syst. Eng. 24 (1) (2024).

[26] F. Ozyurt, J. Majidpour, T.A. Rashid, C. Koç, Offline handwriting signature verification: a transfer learning and feature selection approach, arXiv preprint, arXiv:2401.09467, 2024.

[27] M.P. Nugraha, A. Nurhadiyatna, D.M.S. Arsa, Offline signature identification using deep learning and Euclidean distance, Lontar Komputer: Jurnal Ilmiah Teknologi Informasi 12 (2) (2021) 102–111.

[28] Z. Hashim, H. Mohsin, A. Alkhayyat, Offline handwritten signature identification based on hybrid features and proposed deep model, Iraqi J. Comput. Sci. Math. 5 (1) (2024) 220–236.

[29] Z. Zeng, Multi-scale attention-based individual character network for handwritten signature verification, in: 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA), IEEE, 2022, pp. 1–5.

[30] F.M. Alsuhimat, F.S. Mohamad, Offline signature verification using long short-term memory and histogram orientation gradient, Bull. Electr. Eng. Inform. 12 (1) (2023) 283–292.

[31] F.M. Alsuhimat, F.S. Mohamad, A hybrid method of feature extraction for signatures verification using cnn and hog a multi-classification approach, IEEE Access 11 (2023) 21873–21882.

[32] D. Keykhosravi, S.N. Razavi, K. Majidzadeh, A.B. Sangar, Offline writer identification using a developed deep neural network based on a novel signature dataset, J. Ambient Intell. Humaniz. Comput. 14 (9) (2023) 12425–12441.

[33] A. Gutub, S. Altalhi, B. Ghazwani, Offline efficient signature authentication using octave convolution neural network, Arab. J. Sci. Eng. (2025) 1–16.

[34] A. Badie, H. Sajedi, Offline handwritten signature authentication using graph neural network methods, Int. J. Inf. Technol. (2024) 1–11.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[36] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv preprint, arXiv:1607.06450, 2016.

[37] U. Pal, T. Wakabayashi, F. Kimura, Offline handwritten signature verification: a new benchmark, in: International Conference on Document Analysis and Recognition (ICDAR), 2011, pp. 481–485.

[38] I.K. Sethi, et al., Cedar signature verification dataset, http://www.cedar.buffalo.edu/NIJ/data/signatures.rar, 2004.

[39] S. Setayeshi, S. Karamzadeh, M. Hamdollahzadeh, Utsig: a Persian offline signature dataset, IET Biometrics 7 (2) (2018) 130–139, https://doi.org/10.1049/iet-bmt.2017.0042.

[40] Icdar 2011 signature verification competition dataset (sigcomp2011), http://www.iapr-tc11.org/mediawiki/index.php/Signature_Verification_Competition_2011_(SigComp2011), 2011.

[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[42] J. Li, C. Zhang, X. Wang, L. Du, Multi-scale cross-modal spatial attention fusion for multi-label image recognition, in: Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part I 29, Springer, 2020, pp. 736–747.

[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint, arXiv:1710.10903, 2017.