

Arabic calligraphy style recognition with edge enhancement and attention

Bilal Abdulrahman Tuama^{1,2}, Farhan Mohamed^{1,3}

¹Department of Emergent Computing, Faculty of Computing, University of Technology Malaysia, Johor Bahru, Malaysia

²Department of Computer Science, College of Applied Science, University of Samarra, Samarra, Iraq

³Media and Game Innovation Center of Excellent, Institute of Human Centered Engineering,
Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

Article Info

Article history:

Received Jan 31, 2026

Revised Jun 18, 2026

Accepted Jul 5, 2026

Keywords:

Arabic calligraphy

Attention mechanism

CNN

Edge extraction

Style recognition

ABSTRACT

Arabic calligraphy style recognition is a challenging task because visually similar styles often share curved strokes, connected structures, and subtle ornamental variations. This study proposes a convolutional neural network (CNN) architecture that combines learnable edge enhancement, squeeze-and-excitation (SE) attention, and optional BiLSTM-based sequential modeling for Arabic calligraphy style classification. The edge module is designed to preserve fine-grained stroke boundaries, while the attention mechanism emphasizes informative channel responses and the BiLSTM models width-wise contextual dependencies. Experiments were conducted on a public Arabic calligraphy style dataset using a stratified 70/15/15 train-validation-test split. The best configuration achieved 98.22% accuracy and 98.20% F1-score on the test subset. These findings indicate that adaptive edge-aware feature learning can improve the discrimination of visually similar Arabic calligraphy styles. Nevertheless, broader validation on additional datasets and repeated runs is still needed to confirm generalizability.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Farhan Mohamed

Department of Emergent Computing, Faculty of Computing, University of Technology Malaysia

Johor Bahru, Malaysia

Email: farhan@utm.my

1. INTRODUCTION

Arabic calligraphy style recognition is an important problem in document image analysis, especially for digital archiving, historical manuscript processing, and cultural heritage retrieval. It is important to distinguish this task from optical character recognition (OCR). While OCR aims to decode textual content at the character or word level, calligraphy style recognition focuses on identifying the visual style category of the script image regardless of the written text [1]. Arabic script introduces additional difficulty because it is written from right to left, characters change shape depending on their position in the word, and calligraphic styles may share highly similar visual structures [2], [3].

Arabic calligraphy is not only a writing system but also an artistic practice characterized by harmony, rhythm, and stylistic variation. Styles such as Thuluth, Ruqaa, and Naskh exhibit distinctive visual patterns, yet many of these differences are encoded in subtle stroke curvature, thickness, spacing, and ornamentation. Conventional handcrafted descriptors can capture part of this structure, but their generalization is often limited when styles are visually close or when samples are noisy, degraded, or low in resolution [4], [5]. This motivates the use of deep models that can preserve both structural and contextual cues. Figure 1 illustrates the same Arabic text presented in different calligraphic styles, highlighting the visual variations that make Arabic calligraphy style recognition challenging.

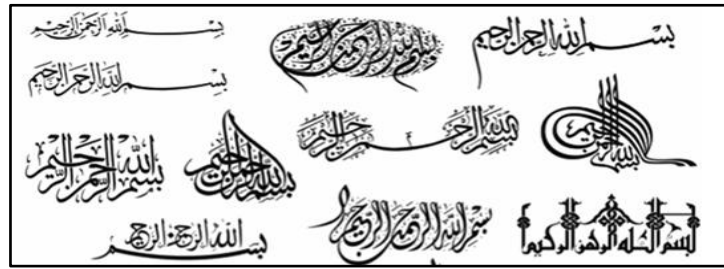


Figure 1. The same text but in different calligraphy styles [4]

Edge detection from images is an essential issue in image-based classification problems, particularly for complex scenarios like Arabic calligraphy and visual similarities [5]. Unlike the printed Latin typefaces, Arabic calligraphy styles exhibit overlapping shapes, curved lines and smoothness in stroke transitions that normally complicate style classification. Standard convolutional networks may have difficulty in modeling these fine-scale details, relying solely on the amplitude of raw pixels [6]. To overcome this, edge extraction techniques are often employed to illuminate the structural boundaries and outlines of the glyphs. Conventional methods such as Sobel [7], [8] filters apply predefined kernels to enhance horizontal, vertical, or diagonal gradients, thus emphasizing shape boundaries. However, these static operators are not adaptive and may fail to capture style-specific stroke variations in diverse calligraphy types. In contrast, learnable edge extractors integrated into deep neural networks provide a data-driven alternative [9]. Thanks to learning proper edge-enhancing filters in training, they are able to selectively emphasize the most discriminative stroke patterns for different styles. This not only makes features more robust, but it also makes models more interpretable as structural cues that are needed to discriminate similar typefaces are retained. An example of edge detection operation on Arabic letters is presented in Figure 2.

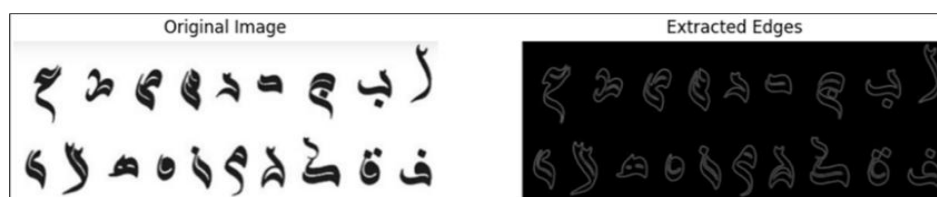


Figure 2. Example of edge extraction

In complex scripts such as Arabic, attention mechanisms help the network focus on the most informative visual cues, while sequence modeling can capture contextual dependencies between neighboring structures. SE attention improves channel-wise feature recalibration by emphasizing informative responses and suppressing irrelevant activations [10], [11]. This is particularly useful in Arabic calligraphy, where small structural differences can have strong stylistic significance [12].

This study evaluates whether adaptive edge enhancement, channel attention, and width-wise sequential modeling can improve Arabic calligraphy style classification. The tested variants include a convolutional neural network (CNN) with attention, a CNN with Sobel edge maps, a CNN with a learnable edge extractor, and a baseline CNN without sequential modeling. Compared with earlier studies that emphasized handcrafted features, conventional classifiers, or transfer learning, the present work investigates the joint role of learnable edge-aware enhancement, attention-guided feature refinement, and contextual modeling in a unified architecture.

The main contributions of this work are summarized as follows:

- We design a novel CNN architecture for Arabic font classification that integrates attention and edge enhancement modules.
- We compare the effectiveness of fixed (Sobel) and learnable edge extraction in highlighting structural font features.
- We demonstrate the benefit of BiLSTM layers in modeling spatial dependencies for classification.
- We provide a comprehensive evaluation on a public Arabic font dataset with class balancing and stratified validation.

The rest of this paper is organized as follows: Section 2 reviews the related work and highlights the techniques of existing approaches. Section 3 describes the proposed methodology, including dataset preparation and model architecture. Section 4 presents the experimental setup and results comparisons. Finally, Section 5 concludes the paper and outlines directions for future work.

2. RELATED WORKS

Very limited work has been conducted in the development of Arabic calligraphy recognition from computational and dataset construction points of view. Kaoudja *et al.* [13] introduced a new system for the classification of Arabic calligraphy styles based on hand-made texture-based features and ensemble learners. The authors identified and built a new dataset of 1685 images in nine common styles (e.g.) Naskh, Diwani, Thuluth, Kufic from various sources such as historical manuscripts and printed documents. Accurate and discriminative visual features were extracted using local phase quantization (LPQ) that is robust under noise and blur. Classification was conducted by various classifiers: multi-layer perceptron, support vector machine, and k-nearest neighbor (MLP, SVM and KNN), and different decision-level fusion methods were investigated. The Sum rule combination showed the best performance with an accuracy of 96.91% on average and surpassed individual classifiers.

Another significant study, Saber *et al.* [14] introduced an offline recognition system of artistic Arabic calligraphy scripts having complex structural textures, patterns, and different forms of letter shapes. The issues posed by discriminating calligraphic styles such as inter-letter overlap and visual similarity, were tackled with the generation of distinctive features for each script type in a feature vector. We used the genetic algorithm technique for extracting an optimal set of features and an image resolution that could improve the classification result. The classifier was a neural network. The system was tested using two datasets: a local dataset, and the first public dataset that consist of ten different synthetic Arabic fonts. The reported error rates were 8.02 and 7.55% for the two experiments above respectively, indicating that the proposed method achieved good performance both in hand-written calligraphy and computer-generated calligraphy recognition.

Aboud *et al.* [15] introduced an initiative for a specialized dataset and text corpus to leveragemachine learning approaches. The authors collected 3,467 images for Arabic calligraphic letters in 32 letter types and additionally extracted 544 different phrases from calligraphic art pieces. They programmed an automatic handwriting analyst and used ImageJ to manually annotate and obtain position deltas for each letter. For the classification, two feature extraction techniques were used, Bag of SURF Features and HOG features combined with SVM. The HOG-based model had a better classification performance (60%) than the SURF-based model (57%). Imane *et al.* [16] proposed an optical font recognition (OFR) framework for Arabic calligraphy, which overcomes the limitations that are encountered by previous methods due to degraded and historical documents. The features are extracted at pixel level based on edge direction matrices (EDMS) that encode distinctive texture patterns in the calligraphy fonts. A backpropagation neural network (BPNN) classification was implemented, which achieves better recognition performance of font types. The model obtained a texture generation (92.8%) and recognition accuracy (43.7%) on Arabic output degraded images, respectively.

Hussein [17] also proposed the printed Arabic text set (PATS-A01), which contains 2,766 text lines cropped from classical Arabic books of eight fonts: Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic Andalus and traditional Arabic. Multiple feature types were analyzed with two neural models: extreme learning machine (ELM) and fast learning network (FLN). Experiments revealed that the performance of FLN and ELM was sensitive to feature type and network size. This demonstrates the promise of lightweight and fast-learning models in Arabic font recognition, though the role of diverse and content-rich datasets is critical for learning robust classifiers. As for transfer learning techniques, Güreş and Gökbay [18] analyzed two different calligraphy datasets to investigate calligraphic works. A first dataset with 9 unique calligraphy styles and the second one for the classification tasks, consisting of classes representing characters in diverse calligraphic styles (i.e., no overlapping patterns). The experimental results show good accuracy of about 90% and 79% for F1 scores of styles and letter classification.

Finally, the work in [19] attempted to study the impact of applying texture descriptors for the purpose of recognising Arabic artistic writing. The six popular and landmark texture descriptors (GLCM, LBP, WLD, LPQ, HOG and BSIF) were used in our study. These descriptors were fed to five different machine learning classifiers (SVM), random forest (RF), decision tree (DT), K-nearest neighbors (KNN) and linear discriminate analysis (LDA). The test is done on a proprietary dataset that consists of nine styles of Arabic calligraphy. From the experimental results, we find that the filter-based features are more useful than statistical features for this task. In particular, the highest accuracy of 94.7% was obtained with the combination of BSIF descriptor and SVM classifier. Finally, it is also observed that recognition performance was the worst for Thuluth and Mohakik styles as a result of their similar structures.

Overall, previous studies demonstrate the value of handcrafted texture descriptors, classical machine learning, and transfer learning for Arabic script analysis. However, three gaps remain. First, limited work has explicitly examined adaptive edge enhancement for preserving style-specific stroke structure in Arabic calligraphy. Second, the joint contribution of channel attention and sequential modeling has not been sufficiently analyzed for fine-grained style classification. Third, many studies report strong results but provide limited discussion of reproducibility, stability, and the interpretation of learned structural cues. This study addresses these gaps by combining learnable edge extraction, SE attention, and BiLSTM-based contextual modeling within a unified classification framework [20], [21].

3. RESEARCH METHOD

The proposed method was developed to improve fine-grained Arabic calligraphy style classification by combining explicit structural enhancement with deep visual representation learning. The overall pipeline that showed in Figure 3 consists of input preprocessing, edge extraction, hierarchical convolutional feature learning, optional width-wise sequence modeling, and final classification.

The model is structured in a sequence of modular stages, beginning with input preprocessing and edge extraction, followed by hierarchical convolutional layers for feature learning, and ending with a classification block. By incorporating both spatial and contextual information, the architecture aims to offer robust performance across varied calligraphic styles.

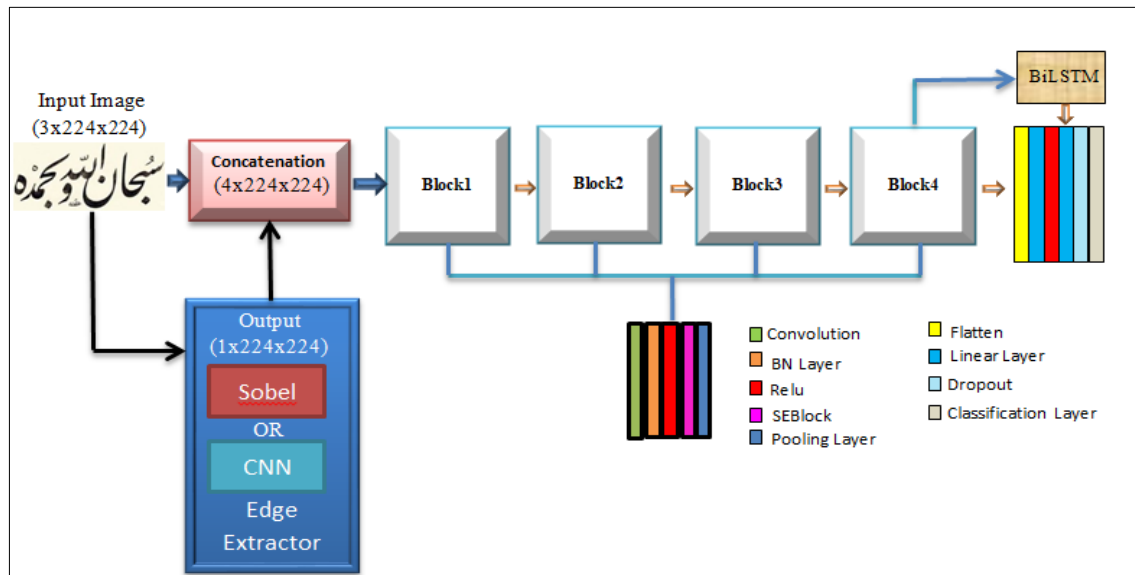


Figure 3. The proposed system architecture

3.1. Dataset

In this study, we used the Arabic calligraphy style classification dataset introduced in [13]. The dataset was selected because it provides multiple classical Arabic calligraphy styles with visible intra-class variation and sufficient samples for supervised evaluation. To preserve class distribution across subsets, the images were divided using a stratified split of 70% for training, 15% for validation, and 15% for testing. The validation subset was used for model selection and early stopping, while the test subset was reserved only for final evaluation. All images were resized to 224×224 pixels before training. No additional data augmentation was applied in the current experiments. This allowed a controlled evaluation of the proposed architecture on the available dataset, although broader augmentation and repeated-run analysis remain important directions for future work. The dataset offers several important characteristics:

- Diversity of styles: It covers a wide spectrum of classical Arabic calligraphy styles.
- Adequate size: Each style includes a sufficient number of labeled images to support learning-based approaches.
- Intra-class variation: The dataset includes variations in scale and phrase length within each style category, which enriches the model's ability to generalize.

Specifically, the dataset consists of nine calligraphic styles: Naskh, Diwani, Parsi, Rekaa, Thuluth, Maghribi, Kufic, Mohakek, and Square-Kufic, with each style representing a unique class label, as shown in Figure 4. A total of 1685 images are included, with 180 to 195 images per class. The images were captured using an 18-megapixel camera, ensuring high resolution and clarity. To ensure visual and contextual diversity, samples were collected from various sources such as books, manuscripts, and online resources. Each image was manually segmented to produce phrases of varying lengths, and all samples were meticulously categorized according to their corresponding calligraphy style.



Figure 4. Examples of calligraphic styles in the dataset

3.2. Edge extraction

In this work, we compare two edge enhancement strategies: a fixed Sobel-based branch and a learnable edge extraction branch. The objective of both branches is to highlight stroke contours and boundary transitions that are informative for style discrimination. We investigate fixed and learnable edge enhancement modules and compare their efficiency in the application of Arabic calligraphy style classification. In our present implementation, we use standard Sobel filters [22], [23] for edge detection as depicted in Figure 5, which consist of the horizontal and vertical operators. These filters are applied to grayscale versions of the input images to highlight directional gradients along both axes.

For the Sobel-based branch, the RGB image is first converted to grayscale and then processed using horizontal and vertical Sobel kernels. The resulting gradient responses are combined to produce a single edge magnitude map, which is concatenated with the original RGB channels. For the learnable branch, the network predicts a soft edge map directly from RGB input through trainable convolutional filters, enabling the edge representation to adapt to calligraphy-specific stroke patterns. Formally, let I denote the input RGB image. The learnable edge extractor is defined as (1):

$$E = \sigma(\text{Conv}_2(\delta(\text{BN}(\text{Conv}_1(I)))))) \quad (1)$$

where Conv_1 and Conv_2 are convolutional layers, BN denotes batch normalization, $\sigma(\cdot)$ is ReLU activation function and $\delta(\cdot)$ is Sigmoid activation function. The output E is a soft edge map that is learned jointly with the classification network.

In the Sobel-based edge enhancement module, the input images are first converted to grayscale by averaging the RGB channels. Fixed horizontal and vertical Sobel kernels are then applied to extract the edge map in both directions, which are combined to produce a composite edge magnitude map. In contrast, the learning edge extracts process RGB input directly and uses trainable convenes layers to learn optimal edge representations during training. This allows the model to be favorably detecting the work-specific shape and stroke pattern, which provides more flexibility than the static edge filter [24].

$$w_{\text{vertical}} = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad w_{\text{horizontal}} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{pmatrix},$$

Figure 5. Standard sobel filters

In contrast to fixed edge detection method, we also experiment with a learnable edge extractor based on trainable convolutional layers. This module receives raw RGB input and applies two consecutive 3×3 convolutional layers with batch normalization and Relu activation as shown in Figure 6. The final layer produces a single channel edge map using a sigmoid activation, allowing the model to learn soft edge representations that are optimized for the downstream classification task. This learnable edge enhancement enables the network to adaptively focus on task relevant contours and stroke patterns, which are often subtle and complex in Arabic calligraphy styles.

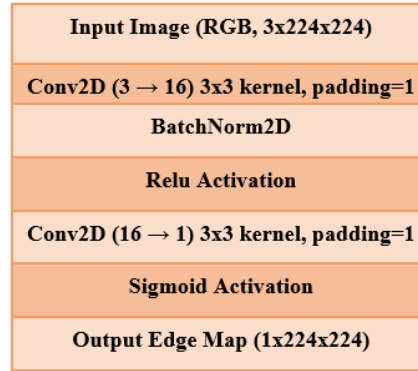


Figure 6. The proposed learnable edge extractor

Although both branches aim to emphasize structural information, the learnable extractor is expected to be more flexible because it can adapt to curvature, stroke thickness variation, and subtle contour transitions that fixed gradient kernels may not capture effectively [25], [26]. In both cases, the goal is to highlight stroke contours and boundary transitions that are characteristic of calligraphic patterns. The output of this stage is an edge map of shape (1×224×224). Table1 shows a comparison between Sobel and learnable edge extractors in number of aspects.

Table1. Comparison between sobel and learnable edge extractors		
Aspect	Sobel edge extractor	Learnable edge extractor
Input type	Converts RGB input to gray scale before processing	Uses raw RGB input with three channels
Filter TYPE	Fixed horizontal and vertical Sobel kernels	Trainable convolutional filters learned from data
Edge orientation	Captures only horizontal and vertical gradients	Can adaptively learn complex, multi-directional edge features
Trainability	Non-trainable (weights are fixed and manually defined)	Fully trainable end-to-end within the network
Adaptability to data	Cannot adapt to specific stroke variations or font styles	Leams edge patterns tailored to the dataset and task
Performance complex Fonts	Limited effectiveness on visually similar or highly stylized calligraphy	Higher accuracy in capturing subtle differences and structural details

The classification backbone consists of four convolutional blocks followed by batch normalization, ReLU activation, and optional SE attention. SE attention recalibrates channel-wise feature responses and helps the network focus on discriminative visual cues. Given an intermediate feature tensor X in $\mathbb{R}^{(C \times H \times W)}$, the squeeze step computes as (2):

$$z_c = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W X_c(i, j) \quad (2)$$

and the excitation step learns channel weights, computed as (3):

$$s = \sigma(W_2 \delta(W_1 z)) \quad (3)$$

The recalibrated output is then obtained by (4):

$$\tilde{X}_c = s_c \cdot X_c \quad (4)$$

where X is the input feature tensor with C channels and spatial dimensions $H \times W$. The squeeze operation computes a channel descriptor z_c using global average pooling. The excitation operation then generates channel-wise weights through two fully connected transformations followed by ReLU and sigmoid activations. Finally, each channel X_c is rescaled by its corresponding weight s_c to produce the recalibrated feature map.

After hierarchical feature extraction, spatial down-sampling is applied to retain the most informative semantic content while reducing resolution. When sequential modeling is enabled, the output of the last convolutional block is reshaped along the horizontal axis and processed by a BiLSTM. This width-wise design is motivated by the connected nature of Arabic writing, in which neighboring visual elements along the horizontal direction often carry contextual stylistic information.

3.3. Classification model

In this work, we propose a novel deep learning architecture tailored for the classification of Arabic calligraphy styles. Once the input tensor is prepared from the previous stage, it is passed through a series of four convolutional blocks designed to extract increasingly abstract and discriminative spatial features from the image. Each block consists of a convolutional layer followed by batch normalization and a Relu activation function. To enhance the model's ability to focus on informative feature maps, each block optionally incorporates a squeeze-and-excitation (SE) attention module [27], [28] that shown in Figure 7, which adaptively recalibrates channel-wise feature responses by modeling interdependencies between channels.

After feature transformation, spatial down sampling is performed using max pooling or adaptive average pooling, which progressively reduces the spatial resolution while retaining semantic information. Across the four blocks, the feature representation is transformed from low-level pixel patterns to high-level semantic abstractions such as character edges, curves, and stylistic elements unique to specific Arabic calligraphy styles. In configurations where sequential modeling is enabled, the output of the last convolutional block (of shape $B \times 512 \times 7 \times 7$) is reshaped to model the horizontal structure of the image where (B) represent the batch size. Specifically, the tensor is permuted and flattened to form a sequence of length 7 (width-wise), where each step in the sequence has a feature dimension of 512×7 . This sequence is then passed through a bidirectional Long Short-Term Memory (BiLSTM) layer, which captures dependence and relationship on locals in both forward and backward directions. This mechanism is particularly useful for Arabic calligraphy, where the shape and connectivity of the letter often depends on the reference within the word or line [29]. BiLSTM enables model to integrate spatial reference at the width of the image, effectively captures structural dependence between neighboring visual elements.

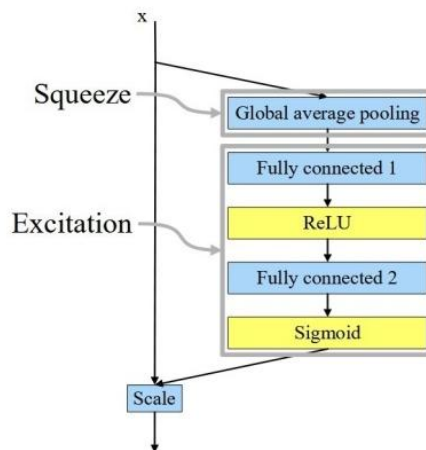


Figure 7. SE attention mechanism

The final classifier predicts the calligraphy style label using either the BiLSTM representation or a fully connected head without sequential modeling. Training is performed with weighted cross-entropy loss, defined as (5):

$$L_{CE} = -\sum_{k=1}^K w_k y_k \log(\hat{y}_k) \quad (5)$$

where K is the number of classes, w_k denotes the class weight for class k and y_k is the ground-truth label. This weighting reduces the effect of class imbalance and supports a fairer contribution of each class during optimization.

Finally, the feature representation is passed through a classification head to predict the font category. If the BiLSTM is used, the last hidden state of the BiLSTM is fed into a fully connected layer to generate logits corresponding to each class. Otherwise, the feature map from the last convolutional block is flattened and passed through a series of fully connected layers with dropout and Relu activation, which serve to reduce over-fitting and introduce non-linearity. The final output layer produces a vector of class logits, which are interpreted as un-normalized probabilities over the set of Arabic calligraphy styles. These logits are then used to compute the cross-entropy loss during training and to assign the most probable label during inference. Table 2 presents a detailed explanation of the proposed model.

Table 2. Description of the proposed model

Stage	Layer / Operation	Shape
Input	RGB image	$B \times 3 \times 224 \times 224$
Edge extract	Sobel or learnable edge	$B \times 1 \times 224 \times 224$
Concatenation	RGB + Edge channel	$B \times 4 \times 224 \times 224$
	Conv2D (4→64, 3×3)	$B \times 64 \times 224 \times 224$
	BN	
Block 1	Relu	
	SEBlock	$B \times 64 \times 112 \times 112$
	MaxPool2d (2×2)	
	Conv2D (64→128, 3×3)	$B \times 128 \times 112 \times 112$
	BN	
Block 2	Relu	
	SEBlock	$B \times 128 \times 56 \times 56$
	MaxPool2d (2×2)	
	Conv2D (128→256, 3×3)	$B \times 256 \times 56 \times 56$
	BN	
Block 3	Relu	
	SEBlock (optional),	$B \times 256 \times 28 \times 28$
	MaxPool2d (2×2)	
	Conv2D (256→512, 3×3)	$B \times 512 \times 28 \times 28$
	BN	
Block 4	Relu	
	SEBlock (optional),	$B \times 512 \times 7 \times 7$
	AdaptiveAvgPool (7×7)	
	Reshape: Permute + Flatten (to sequence)	$B \times 7 \times 3584$
LSTM branch	BiLSTM (input=3584, hidden=256×2)	$B \times 7 \times 512$
	Select last step	$B \times 512$
Classifier (with LSTM)	Linear (512 → num_classes)	$B \times \text{num_classes}$
	Flatten ($B \times 512 \times 7 \times 7 \rightarrow B \times 25088$)	$B \times 25088$
Classifier (no LSTM)	Linear (25088 → 512)	$B \times 512$
	Dropout (0.4), Relu	$B \times 512$
	Linear (512 → num_classes)	$B \times \text{num_classes}$
Output	Class logits (raw scores)	$B \times \text{num_classes}$

4. RESULTS AND DISCUSSION

A comprehensive range of experiments was conducted to evaluate the effectiveness of the CNNs architecture proposed for Arabic calligraphy style classification. These experiments examined the contribution of various architectural components, including attention mechanisms, edge extractors (both hand-crafted and learned), and sequence modeling, which uses BiLSTM layers for overall classification performance. This section underlines experimental settings, ablation configuration, and performance results. The input images representing Arabic calligraphy are first resized to a uniform spatial resolution of 224×224 pixels to ensure consistency across the dataset. The images are then normalized to a range between -1 and 1 using mean and standard deviation values of 0.5 per channel. This normalization step helps accelerate model convergence during training and stabilizes the learning process. The dataset was split into training, validation, and test subsets using a stratified strategy (70/15/15). The validation set was used for early stopping and model selection, while the test set was reserved for final evaluation only. Table 3 summarizes the key training parameters and hyperparameters used in our experiments.

To evaluate the individual contribution of the main architectural components, we performed an ablation study using four model variants. The comparison focuses on three factors: channel attention through

SE blocks, edge extraction type, and width-wise BiLSTM modeling. All results were computed on the held-out testing subset using the same training protocol.

Table 3. Training configuration and hyperparameters

Parameter	Value
Optimizer	Adam
Initial Learning Rate	0.0001
Batch Size	32
Epochs (max)	20
Early Stopping Patience	3 epochs
Loss Function	Cross-Entropy-Loss (weighted)
Device	GPU (NVIDIA RTX 4060)

4.1. Ablation study

To evaluate the individual contributions of the core components of our proposed architecture, we conducted an ablation study comprising two distinct experimental setups. Each configuration isolates or combines specific modules: attention (SEBlock) with sequence modeling via BiLSTM and edge extraction (fixed or learnable). This section provides a detailed analysis of each experiment and its impact on performance in terms of accuracy and F1-score [30]. Table 4 shows a comparison of model variants with different attention, edge extraction, and BiLSTM components on the testing subset.

Table 4. Model variants and performance comparison

No.	Experiment Variant	Attention (SEBlock)	Edge Extraction	BiLSTM	Test Accuracy	F1-score
1	CNN+Attention	✓	X	✓	0.9792	0.9418
2	CNN+Attention+Sobel	✓	Sobel (Fixed)	✓	0.9763	0.9675
3	CNN+Attention+LearnedEdge	✓	Learnable CNN	✓	0.9822	0.9820
4	CNN-NoLSTM	X	X	X	0.9703	0.9290

To assess the role of SE attention and BiLSTM modeling, we compared the attention-based CNN with the baseline CNN-NoLSTM configuration. The attention-based model achieved 97.92% accuracy and 94.18% F1-score, whereas the CNN-NoLSTM configuration achieved 97.03% accuracy and 92.90% F1-score. These results suggest that channel recalibration improves feature discrimination, while width-wise contextual modeling helps preserve stylistic dependencies that are difficult to capture using purely local convolutional responses.

4.1.1. Experiment 1: Effect of attention mechanism and LSTM

To evaluate the role of channel-wise attention, we compared the baseline CNN architecture with and without SE blocks. As shown in Table 4, the model with attention modules (CNN+Attention) achieved an accuracy of 97.92% and an F1-score of 94.18%, compared to 97.03% and 92.90% in the non-attentive configuration (CNN-NoLSTM). This indicates that SE blocks enhance feature discrimination by reweighting informative channels. In the same context, and also from the same experiment we can detect the importance of LSTM on the results. Without LSTM, the model lost the ability to model temporal/spatial dependencies, reducing its overall discriminative power. Figure 8 shows comparison between the two models

4.1.2. Experiment 2: Effect of edge extraction type

To determine the contribution of edge information to our classification performance, we also tested two edge-enhanced baseline CNN variants: one with fixed Sobel filters and another incorporating a learnable edge extractor (both integrated in an SE attention setting). The Sobel-based configuration created a static edge map, which was concatenated with the RGB input to get 4-channel tensor. Compared with fixed Sobel filtering (accuracy 97.63%, F1 0.9675), the learnable edge extractor improved performance (accuracy 98.22%, F1 0.9820), indicating that adaptive edge learning yields better feature representations for Arabic calligraphy. Such improvement demonstrates the benefit of training task-specific edge features which are better adapted to the delicate stroke variations in Arabic calligraphy. It is worth mentioning that learning edge extraction and integrating attention mechanism and LSTM made the model more robust, generalizable. Altogether, these results demonstrate that adaptive edge learning is superior to fixed filtering especially with the use of Attention modeling. Comparison between the two models is given in Figure 9.

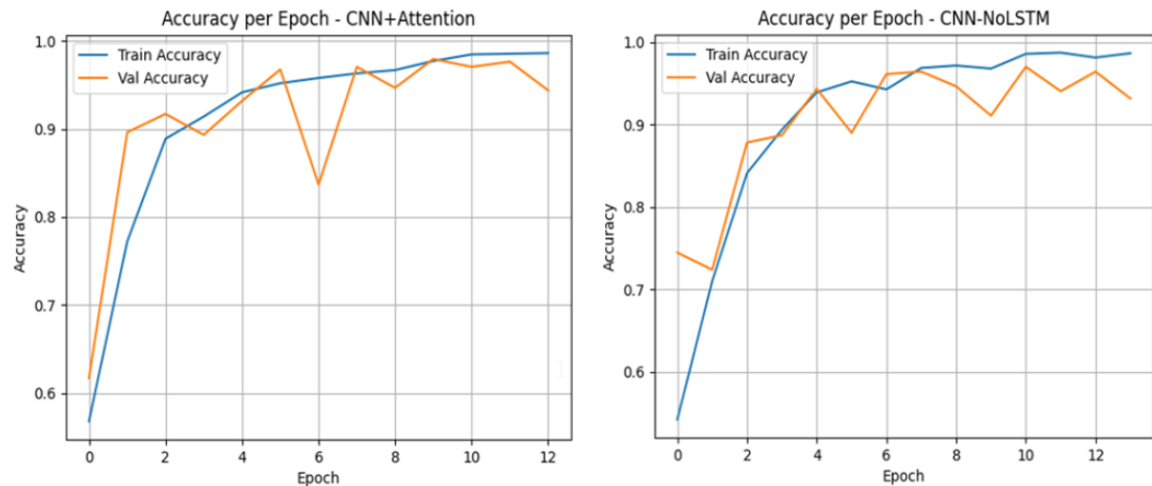


Figure 8. Result comparison of experiment 1

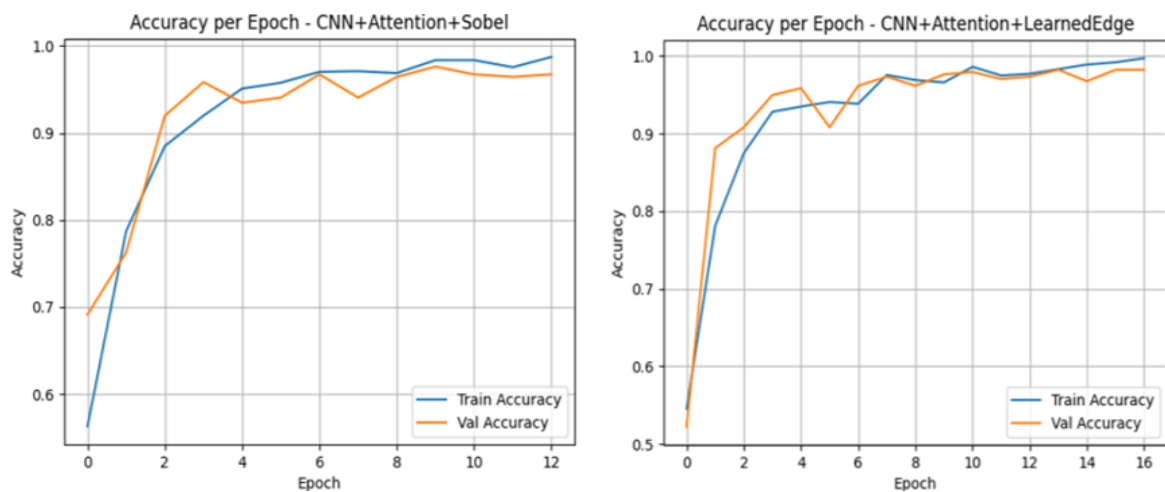


Figure 9. Result comparison of experiment 2

Overall, the quantitative results show a consistent pattern: the full model that combines learnable edge extraction, SE attention, and BiLSTM achieves the strongest performance on the current dataset. The baseline CNN reached 97.03% accuracy and 92.90% F1-score. With additional attention and contextual modeling, the scores improved, and the best configuration achieved 98.22% accuracy and 98.20% F1-score.

A plausible explanation for this trend is that the three components provide complementary benefits. The learnable edge extractor preserves fine-grained structural cues, SE attention emphasizes informative channel responses, and the BiLSTM captures width-wise contextual dependencies across connected script patterns. Together, these elements improve the discrimination of visually similar Arabic calligraphy styles. This study conducted a series of experiments to evaluate the effectiveness of attention mechanisms, temporal modeling, and edge extraction techniques in Arabic calligraphy classification. Starting from a baseline CNN, the model was progressively enhanced with SE attention blocks, LSTM units, and two types of edge extractors: fixed Sobel filters and a learnable CNN-based edge module. Figure 10 shows a comparison of all models in accuracy and F1 score.

The baseline CNN achieved 97.03% accuracy and 0.9289 F1-score. Adding SE attention improved accuracy and balance, reaching 97.92%/0.9418. Incorporating LSTM further boosted performance to 98.09%/0.9563, by modeling sequential patterns. Using Sobel filters slightly reduced accuracy (97.63%) but increased F1-score (0.9675), indicating better precision-recall trade-off due to edge enhancement. The best results came from the learnable edge extractor combined with attention and LSTM, achieving 98.22% accuracy and 0.9820 F1-score, outperforming all other configurations.

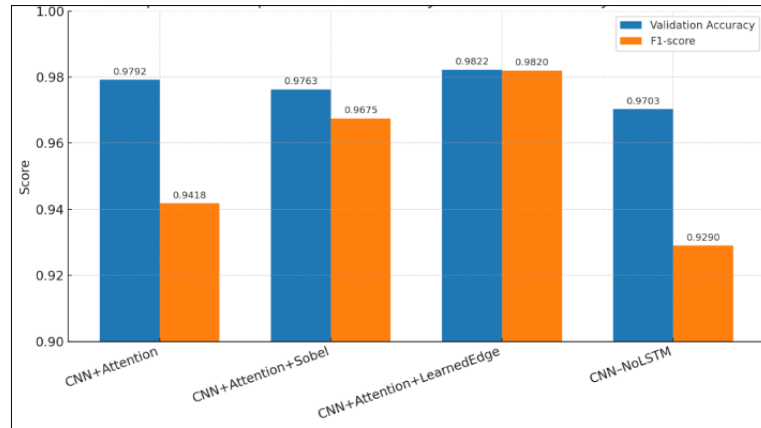


Figure 10. Result comparison of all models

Compared to previous studies, the proposed full model demonstrated superior performance as shown in Table 5, confirming the benefit of combining adaptive edge features with attention and temporal learning in fine-grained image classification. The comparison of F1 scores between [18] and our proposed model is 90% and 98.22%, respectively.

The obtained results provide additional evidence that structural enhancement and contextual modeling can complement convolutional feature learning in Arabic calligraphy style recognition. In particular, the superior performance of the learnable-edge configuration suggests that adaptive edge representations are better suited than fixed operators for capturing subtle calligraphic variations. However, since the current evaluation was conducted on a single dataset and one stratified split, further validation using repeated experiments, additional datasets, and class-wise analysis is still required to confirm the robustness of these findings.

Table 5. Results comparison with previous related works

Model	Accuracy
Traditional ML [13]	96.91%
Transfer learning [18]	91%
Texture descriptors [19]	94.7%
Our proposed model	98.22%

5. CONCLUSION

This paper presented a CNN-based architecture for Arabic calligraphy style recognition that integrates edge enhancement, SE attention, and optional BiLSTM-based sequential modeling. The proposed design was developed to improve the discrimination of visually similar calligraphic styles by combining structural edge information with attention-guided feature refinement. Experimental evaluation on a public Arabic calligraphy dataset showed that the proposed approach achieved strong classification performance, with the best configuration reaching 98.22% accuracy and 98.20% F1-score. The results also indicate that the learnable edge extraction module provides a more effective representation than fixed Sobel filtering, particularly for capturing subtle stroke variations and fine-grained stylistic details. Overall, the findings suggest that integrating adaptive edge learning and attention mechanisms can be a useful direction for fine-grained Arabic calligraphy style classification.

Future work can extend this study in several directions. First, the proposed architecture should be evaluated on additional Arabic calligraphy datasets and more diverse real-world document images to further examine its broader applicability. Second, repeated runs, cross-validation, and class-wise error analysis such as confusion matrices can provide a more comprehensive assessment of model stability and misclassification patterns. Third, computational profiling, including parameter count, FLOPs, memory footprint, and inference time, would help better assess the practical efficiency of the model. Fourth, benchmarking against recent pretrained and transformer-based architectures, such as ResNet50, EfficientNet, ViT, ConvNeXt, and Swin Transformer, would provide stronger comparative evidence within the context of modern visual recognition systems. Finally, future enhancements may explore multi-scale edge modeling, transformer-based sequence learning, and semi-supervised or unsupervised approaches to improve performance while reducing dependence on limited labeled data.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Bilal Abdulrahman Tuama	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓			
Farhan Mohamed						✓				✓		✓	✓	

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nterpretation

R : **R**esources

D : **D**ata Curation

O : **O**riginal Draft

E : **E**diting

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

No new data were created. The study used the publicly available dataset reported in [13].

DECLARATION OF USE OF AI IN THE WRITING PROCESS

The authors used ChatGPT (OpenAI) during the preparation of this manuscript to improve grammar, paraphrase selected passages, and enhance clarity and readability. After using the tool, the authors carefully reviewed and edited the content and take full responsibility for the content of the publication.

REFERENCES

- [1] E. El-Awadly, A. Ebada, and A. Al-Zoghby, "Arabic handwritten text recognition systems and challenges and opportunities," *The Egyptian Journal of Language Engineering*, vol. 10, no. 2, pp. 84–103, 2023, doi: 10.21608/ejle.2023.193993.1043.
- [2] H. Butt, M. R. Raza, M. J. Ramzan, M. J. Ali, and M. Haris, "Attention-based CNN-RNN arabic text recognition from natural scene images," *Forecasting*, vol. 3, no. 3, pp. 520–540, 2021, doi: 10.3390/forecast3030033.
- [3] D. M. Alashari, "The aesthetics values of arabic calligraphy within islamic art: An analytical study," *Asian Journal of Arts*, vol. 3, no. 3, pp. 44–48, 2021, [Online]. Available: <http://myjms.mohe.gov.my/index.php/ajact>
- [4] Z. Alyafeai, M. S. Al-shaibani, M. Ghaleb, and Y. A. Al-Wajih, "Calliar: an online handwritten dataset for Arabic calligraphy," *Neural Computing and Applications*, vol. 34, no. 23, pp. 20701–20713, 2022, doi: 10.1007/s00521-022-07537-2.
- [5] M. S. Kasem, M. Mahmoud, and H. S. Kang, "Advancements and challenges in arabic optical character recognition: a comprehensive survey," *ACM Computing Surveys*, vol. 58, no. 4, Dec. 2025, doi: 10.1145/3768150.
- [6] T. B. A. Gader and A. K. Echi, "Attention-based deep learning model for arabic handwritten text recognition," *Machine Graphics and Vision*, vol. 31, no. 1–4, pp. 49–73, 2022, doi: 10.22630/MGV.2022.31.1.3.
- [7] P. L. Chithra and P. Bhavani, "A study on various image processing techniques," *International Journal of Emerging Technology and Innovative Engineering*, vol. 5, no. 5, 2019.
- [8] R. S. Shaw and D. Jenilarani, "Comparative image evaluation of lung tuberculosis for improved accuracy by executing sobel and robert operators," in *AIP Conference Proceedings*, 2025, p. 020076. doi: 10.1063/5.0258709.
- [9] Z. Li et al., "Edge feature extraction-based dual CNN for LDCT denoising," *Journal of the Optical Society of America A*, vol. 39, no. 10, p. 1929, 2022, doi: 10.1364/josaa.462923.
- [10] M. M. Khan, M. S. Uddin, M. Z. Parvez, and L. Nahar, "A squeeze and excitation ResNeXt-based deep learning model for Bangla handwritten compound character recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3356–3364, 2022, doi: 10.1016/j.jksuci.2021.01.021.
- [11] M. Z. Nisar, M. S. Ibrahim, M. Usman, and J. A. Lee, "A lightweight deep learning model for automatic modulation classification using residual learning and squeeze–excitation blocks," *Applied Sciences (Switzerland)*, vol. 13, no. 8, p. 5145, 2023, doi: 10.3390/app13085145.
- [12] S. Yuan, Z. Zou, H. Zhou, Y. Ren, J. Wu, and K. Yan, "SECNN: Squeeze-and-excitation convolutional neural network for sentence classification," *IEEE Access*, vol. 13, pp. 42858–42865, 2025, doi: 10.1109/ACCESS.2025.3548111.
- [13] Z. Kaoudja, M. L. Kherfi, and B. Khaldi, "An efficient multiple-classifier system for Arabic calligraphy style recognition," in *Proceedings - ICNAS 2019: 4th International Conference on Networking and Advanced Systems*, 2019. doi: 10.1109/ICNAS.2019.8807829.

- [14] A. Saber, A. Taha, and K. Abd El Salam, "A comprehensive approach to arabic handwriting recognition: deep convolutional networks and bidirectional recurrent models for arabic scripts," *International Journal of Telecommunications*, vol. 04, no. 02, pp. 1–11, 2024, doi: 10.21608/ijt.2024.291347.1052.
- [15] A. Abood, E. Abdelrazek, and A. M. Saad, "Recognizing handwritten arabic characters using deep learning techniques in educational platforms," *Scientific Journal of Educational Development*, vol. 2024, no. 9, pp. 324–344, 2024, doi: 10.21608/sjeud.2024.284278.1032.
- [16] B. Imane, K. Ghizlane, M. Mostafa, and A. Alae, "Boosting arabic OCR performance with multi-head selfattention in an end-to-end CNN-BiLSTM deep learning Pipeline," in *2025 International Conference on Circuit, Systems and Communication, ICCSC 2025*, 2025, doi: 10.1109/ICCSC66714.2025.11135323.
- [17] A. K. Hussein, "Fast learning neural network based on texture for arabic calligraphy identification," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1794–1799, 2021, doi: 10.11591/ijeecs.v21.i3.pp1794-1799.
- [18] D. Z. Güler and İ. Z. Gökbay, "Arabic calligraphy image analysis with transfer learning," *Electrica*, vol. 24, no. 1, pp. 201–209, 2024, doi: 10.5152/electrica.2023.23102.
- [19] Z. Kaoudja, B. Khaldi, and M. L. Kherfi, "Arabic artistic script style identification using texture descriptors," in *CCSSP 2020 - 1st International Conference on Communications, Control Systems and Signal Processing*, 2020, pp. 113–118, doi: 10.1109/CCSSP49278.2020.9151569.
- [20] Z. Kaoudja, M. L. Kherfi, and B. Khaldi, "A new computational method for arabic calligraphy style representation and classification," *Applied Sciences (Switzerland)*, vol. 11, no. 11, p. 4852, 2021, doi: 10.3390/app11114852.
- [21] B. Hakim and B. Ahror, "Deep learning for accurate recognition of arabic handwritten words in historical documents," *Procedia Computer Science*, vol. 244, pp. 57–65, 2024, doi: 10.1016/j.procs.2024.10.178.
- [22] R. Ranjan and V. Avasthi, "Edge detection using guided sobel image filtering," *Wireless Personal Communications*, vol. 132, no. 1, pp. 651–677, 2023, doi: 10.1007/s11277-023-10628-5.
- [23] J. S. Yadav and P. S. Bharathi, "Edge detection of images using prewitt algorithm comparing with sobel algorithm to improve accuracy," in *Proceedings of 3rd International Conference on Intelligent Engineering and Management, ICIEM 2022*, 2022, pp. 351–355, doi: 10.1109/ICIEM54221.2022.9853193.
- [24] K. A. Kadhim, F. Mohamed, F. H. Najjar, and G. A. Salman, "Early diagnose alzheimer's disease by convolution neural network-based histogram features extracting and canny edge," *Baghdad Science Journal*, vol. 21, no. 2, pp. 643–654, 2024, doi: 10.21123/bsj.2024.9740.
- [25] J. Ding *et al.*, "Learnable edge detectors can make deep convolutional neural networks more robust," *Plos One*, vol. 20, no. 9, September, p. e0330299, 2025, doi: 10.1371/journal.pone.0330299.
- [26] Y. Li, W. Kang, W. Zhao, and X. Liu, "MLEDNet: Multi-directional learnable edge information-assisted dense nested network for infrared small target detection," *Electronics (Switzerland)*, vol. 14, no. 17, p. 3547, 2025, doi: 10.3390/electronics14173547.
- [27] X. Jin, Y. Xie, X. S. Wei, B. R. Zhao, Z. M. Chen, and X. Tan, "Delving deep into spatial pooling for squeeze-and-excitation networks," *Pattern Recognition*, vol. 121, p. 108159, 2022, doi: 10.1016/j.patcog.2021.108159.
- [28] M. Al and R. Alif, "State-of-the-art bangla handwritten character recognition using a modified resnet-34 architecture," *Article in International Journal of Innovative Science and Research Technology*, vol. 9, pp. 438–448, 2024.
- [29] B. A. Tuama and F. Mohamed, "A systematic literature review of deep learning methods for handwritten text recognition in historical arabic manuscripts," *Engineering, Technology and Applied Science Research*, vol. 15, no. 4, pp. 25772–25782, 2025, doi: 10.48084/etasr.12123.
- [30] R. Diallo, C. Edalo, and O. O. Awe, "Machine learning evaluation of imbalanced health data: A comparative analysis of balanced accuracy, MCC, and F1 Score," in *STEAM-H: Science, Technology, Engineering, Agriculture, Mathematics and Health*, 2024, pp. 283–312, doi: 10.1007/978-3-031-72215-8_12.

BIOGRAPHIES OF AUTHORS



Bilal Abdulrahman Tuama    is currently an Associate Professor and faculty member at the College of Applied Science, University of Samarra. He has published many research articles on image processing and text recognition. He received his M.Sc. in Computer Science from the University of Science, Malaysia (USM), in 2012, with a specialization in image processing. His research interests include pattern recognition, handwritten text recognition, data visualization, and visual analytics. He has led and contributed to multiple international and local projects in AI-driven visualization, digital twins, and interactive systems. He can be contacted at email: tuamabilal@gmail.com.



Farhan Mohamed    is currently an Associate Professor at the Faculty of Computing, Universiti Teknologi Malaysia (UTM). He is an IEEE Senior Member and serves on the Executive Committee of the IEEE Computer Society Malaysia. He is also a member of ACM MyHCI-UX Malaysia. He received his Ph.D. in Computer Science from Swansea University, United Kingdom, with a specialization in data visualization. His research interests include data visualization, visual analytics, immersive analytics, and extended reality. He has led and contributed to multiple international, national, and industry-funded projects in AI-driven visualization, digital twins, and interactive systems. He can be contacted at email: farhan@utm.my.