BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 74(5), 2026, Article number: 1619
DOI: 10.24425/bpasts.2026.1619

EDU: English Document Understanding, a novel transformer-based approach for recognition of handwritten academic text

Liqun CHENG* and **Shanshan GUO**

College of General Education, Xianning Polytechnic, Xianning 437100, Hubei, China

Abstract. Understanding handwritten text from images plays a critical role in various domains including education, healthcare, and transportation. Contemporary text recognition systems are mostly based on traditional Optical Character Recognition (OCR) methods, which need well-structured printed text. Such systems are inadequate in the realm of education where handwritten content prevails. With this research work, a novel framework is presented intermingling multiple deep learning models for effective understanding of handwritten English script. A transformer-based hand-text recognition engine (HRE) is employed to detect and recognize handwritten text. To attain higher-level language understanding, Microsoft's Phi-2 language model is incorporated for semantic analysis. Uncertainty in language understanding is estimated by standard Bayesian inference through MC-dropout-based sampling. Furthermore, the approach of Parameter-Efficient Fine-Tuning (PEFT) is used to minimize overhead in the encoder and decoder modules. The method requires only 1.9% of the trainable parameters, ensuring speedy training and improved recognition accuracy. Results of the systematic evaluation confirm state-of-the-art performance of the method, with remarkable scores of 0.963, 0.954, and 0.958 achieved for the standard metrics of precision, recall, and F1, respectively. Moreover, the model attains a Character Error Rate (CER) of 2.41% and a Word Error Rate (WER) of 8.13%. These low error rates demonstrate the effectiveness of the proposed framework in accurately recognizing and understanding handwritten scripts. Besides automated grading and assignment analysis, the framework is extendable to a wide range of digital archival and legal document analysis.

Keywords: handwritten recognition; English text understanding; document analysis; auto-comprehension.

1. INTRODUCTION

Despite the established use of digital technology, handwritten content persists in academia and archives [1]. Particularly, recognition and understanding of handwritten text is of significant importance in marking exam papers and mastering proficiency of a language [2]. The conventional OCR methods, primarily designed for printed text, cannot support the inherent irregularities of handwriting. Some of the features of handwritten notes, like inconsistent spacing, non-standard orthographic patterns, and stylistic variability, are deemed as anomalies by the OCR-based systems. Due to these limitations, the conventional OCR methods remained inapplicable for educational tools like contextual understanding, automated grading, and advanced academic data analysis. To address these issues, recent research works exploit deep learning models for robust recognition.

Though convolutional neural networks have achieved notable success in computer vision, the technology is computationally expensive [3]. Similarly, recurrent neural networks also lack the ability to properly model the semantic contexts essential for language understanding [4]. With the emergence of large language models (LLMs), new possibilities are created for context-aware interpretation beyond plain character recognition [5]. While using such models, attention should be paid to proper cleaning and tokenization of the noisy and irregular handwritten text [6]. As stated in [7], the Swin transformer facilitates representation of multi-scale features for effective capture of spatial dependencies and variable stroke patterns of handwritten text. The transformer not only outperforms the traditional CNN approach in text recognition but also supports long-range dependencies and shape irregularities [8]. Therefore, there is a need to incorporate Swin as a visual encoder in the handwritten text recognition (HTR) pipeline. In this way, character-level precision is enhanced with is necessary for semantic alignment and contextual understanding of handwritten documents; hence, this research.

With this study, a unified framework is presented for handwritten text recognition, semantic interpretation, and uncertainty estimation. For effective modeling of spatial dependencies, detection and transcription of handwritten text is performed by the Swin Transformer-based HTR engine. Microsoft's Phi-2 [9] language model is employed in the semantic understanding engine (SUE) for leveraging contextual refinement. To minimize computational cost, parameter-efficient fine-tuning is employed through Low-Rank Adaptation (LoRA) [10] and Decomposition Rank Adaptation (DoRA) [11]. The targeted adaptation in the encoder–decoder architecture aids in minimal use of parameters while learning diverse handwritten styles, thereby reducing training overhead.

Unlike the contemporary approaches, uncertainty estimation is incorporated through Bayesian inference [12]. The Monte

*e-mail: 18694057748@163.com

Manuscript submitted 2025-11-20, revised 2026-05-13, initially accepted for publication 2026-06-09, published in September 2026.

© 2026 The Author(s). This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

Carlo Dropout (MC-Dropout) estimation technique is exploited to trace the inherent inconsistency of handwritten text and to quantify confidence in predictions, see Fig. 1. The model is trained and tested on the IAM Handwriting database [13] in conjunction with real-world handwritten text samples. For effective assessment, the system is subjected to systematic evaluation against the ground-truth annotations. Extensive evaluations are conducted on handwritten educational corpora derived from essays, classroom notes, and examination scripts. Experimental outcomes demonstrate satisfactory performance across multiple metrics, including character-level accuracy, error rates, and calibration measures. The results highlight the effectiveness of the approach in handling variability in handwriting and its applicability in automated grading, document analysis, and intelligent archival. Using the test set of the IAM dataset, a promising accuracy of 0.96 is obtained, whereas a satisfactory accuracy of 0.94 is attained on samples of real-world handwritten text. Similarly, noteworthy rates are achieved for CER (2.41%) and WER (8.13%), demonstrating robust performance of the framework. Uncertainty calibration through Bayesian inference suggests an Expected Calibration Error (ECE) of 1.7%. As a whole, the results underline the reliability of the system in detecting ambiguous handwritten scripts and the applicability of the proposed approach in automated grading, intelligent document processing, and assignment analysis.

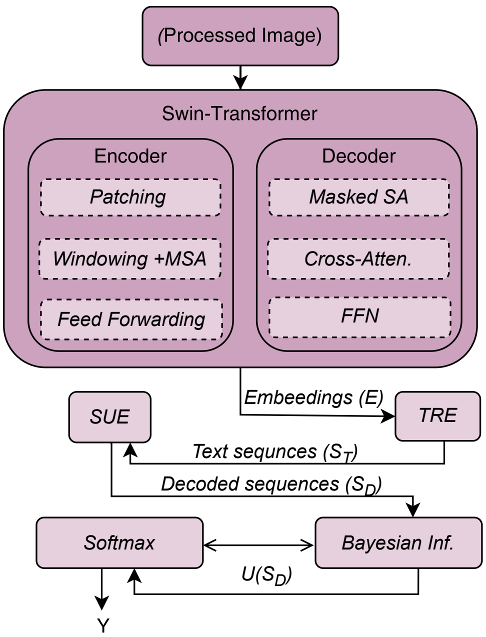

Fig. 1. Schematic of the EDU framework

2. LITERATURE REVIEW

Handwritten text recognition has been in focus as advancements are made in computing technology, particularly in image processing. The initial work of handwritten text recognition (HTR) mainly relied on handcrafted feature extraction methods. Rules-based systems are designed using the techniques of template matching and structural–syntactic analysis [14]. These approaches depend primarily on predefined heuristics to analyze character shape. While effective for constrained datasets, their performance degraded significantly in the presence of writing variability, noise, and cursive scripts. To overcome the challenge, statistical modeling like Hidden Markov Models (HMMs) was suggested [15]. Hybrid systems integrating HMMs and discriminative classifiers were also explored to enhance sequence modeling [16]. The generalization capability of such systems, however, was not up to the mark. Moreover, HMMs are limited by Markov assumptions and are inappropriate to capture complex visual patterns and long-range dependencies [17].

With the advent of machine learning, efficient systems based on Support Vector Machines (SVMs), k-Nearest Neighbors (kNN), and Random Forests were introduced for classification of characters and words [18]. Such machine learning models, though they leverage enhanced discriminative feature spaces, relied mostly on manual feature engineering, which hinders adaptability to diverse handwriting styles. The rise of deep learning brought a paradigm shift in the domain of HTR. Convolutional Neural Networks (CNNs) substituted the extraction of handcrafted features by learning hierarchical visual representations [19]. Recurrent Neural Networks (RNNs) and LSTM networks are used to model sequential dependencies among the tokens [20]. Recently, Transformer-based architectures, like Vision Transformers (ViTs) and Swin Transformers, are used for explicit character segmentation [21]. The performance of the transformers is leveraged by self-attention mechanisms for modeling visual sequences [22]. In NRTR, Sheng *et al.* [23] used a Transformer architecture with minimal convolutional layers for text recognition. Yang *et al.* [24] replaced LSTM with a transformer decoder to enhance accuracy. In SRN, Yu *et al.* [25] combined a Transformer encoder with a convolutional feature network for image encoding and text decoding. Similarly, language models such as GPT, BERT, and Phi-2 offer powerful capabilities in natural language understanding [9].

3. METHODOLOGY

Unlike printed text, handwritten characters are often characterized by discontinuities and overlapping. Owing to differences in style and stroke of the handwritten text, significant intra-class variability is exhibited. To effectively address these challenges, the proposed framework makes use of handwritten-oriented feature modeling where emphasis is placed on local curvature and spatial stroke continuity. To distinguish between visually similar handwritten characters, multi-scale feature extraction is employed to encode fine-grained stroke details along with global structural patterns. An input image $I \in \mathbb{R}^{H\times W}$ is transformed into a sequence of embeddings $E \in \mathbb{R}^{N\times d}$, where N represents the total number of patches and d the embedding dimension. This representation ensures robustness against image distortions and writer-dependent inconsistencies.

The architecture of the EDU framework consists of three units or engines: the embedding generation engine (EGE), text recognition engine (TRE), a nd semantic understanding engine (SUE). Following preprocessing, features are transformed into embeddings in the EGE. The embeddings are fed to the TRE, where a transformer-based encoder-decoder approach is followed. A semantic sequence of the tokens representing handwritten text is performed in the SUE. To obtain an encoder sequence, an input image I is partitioned into non-overlapping patches in the Embedding Generation Engine (EGE). The patches are linearly projected to obtain the appropriate embedding. The encoder contains N layers of multi-head self-attention and feed-forward networks. Without full model retraining, the DoRA module is exploited, which augments the embeddings for lightweight adaptation of visual features. The decoder utilizes LoRA adapters in the attention layer to reduce the number of trainable parameters while maintaining expressive capacity. During inference, the encoder processes the image only once, and the decoder autoregressively generates sub-word tokens. The hierarchical architecture, as illustrated in Fig. 2, facilitates the seamless integration of feature extraction, sequence modeling, and semantic interpretation, thereby transforming low-level image representations into high-level language understanding.

Let an input image I_p be the output from preprocessing. In the TRE, the HTR function maps the image to a sequence of textual tokens- words, sub-words, and letters of the vocabulary V as shown in Fig. 3.

$$S_T = f_{\mathrm{HTR}}\left(I_p\right), \tag{1}$$

$$f_{\mathrm{HTR}}\left(I_p\right) = \{t_1, t_2, \ldots, t_n\}, \tag{2}$$

where $t_i \in V$, semantic analysis of S_T is performed by the Phi2 language model to obtain the semantic or decoded sequence S_D:

$$S_D = f\left(S_T\right). \tag{3}$$

Complex patterns of extracted handwritten notes are transformed into accurate digital representations. The representation is then subsequently used for downstream language understanding and semantic reasoning. The Bayesian interpretation is attained by performing multiple forward stochastic passes over the S_D

$$\left\{S_D^1, S_D^2, S_D^3, \ldots, S_D^n\right\} = f_{\mathrm{phi}}^{\mathrm{drop}}\left(S_T\right). \tag{4}$$

Uncertainty is estimated based on variability in predictions obtained through Monte Carlo (MC) dropout sampling for the set of predictions; $S_D = \{y_1, y_2, \ldots, y_n\}$ obtained from n stochastic forward passes, uncertainty U is defined as

$$U\left(S_T\right) = \frac{\left|\mathrm{unique}\left(S_D^i\right)\right|_{i=1}^{n}}{n}, \tag{5}$$

where unique $\left(S_D\right)$ represents the distinct predictions.

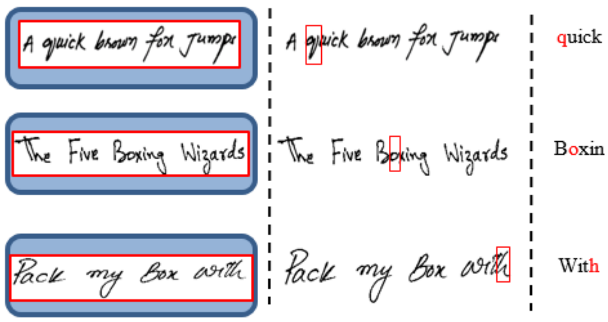

Fig. 3. Mapping illustration of visual tokens to textual tokens

The formulation ensures higher uncertainty for greater diversity in predictions and vice versa. Consistency across multiple stochastic forward passes indicates reliable model behavior, while variation among predictions reflects ambiguity in the inferred output.

To understand the academic contexts of handwritten text, the base system with parametric function f of the model taking input I is represented as

$$f_{\vartheta}(I) = Y, \tag{6}$$

where $Y = \{y_1, y_2, \ldots, y_n\}$ is the predicted sequence and ϑ is the model parameter. Details about the framework architecture are presented in the following subsections.

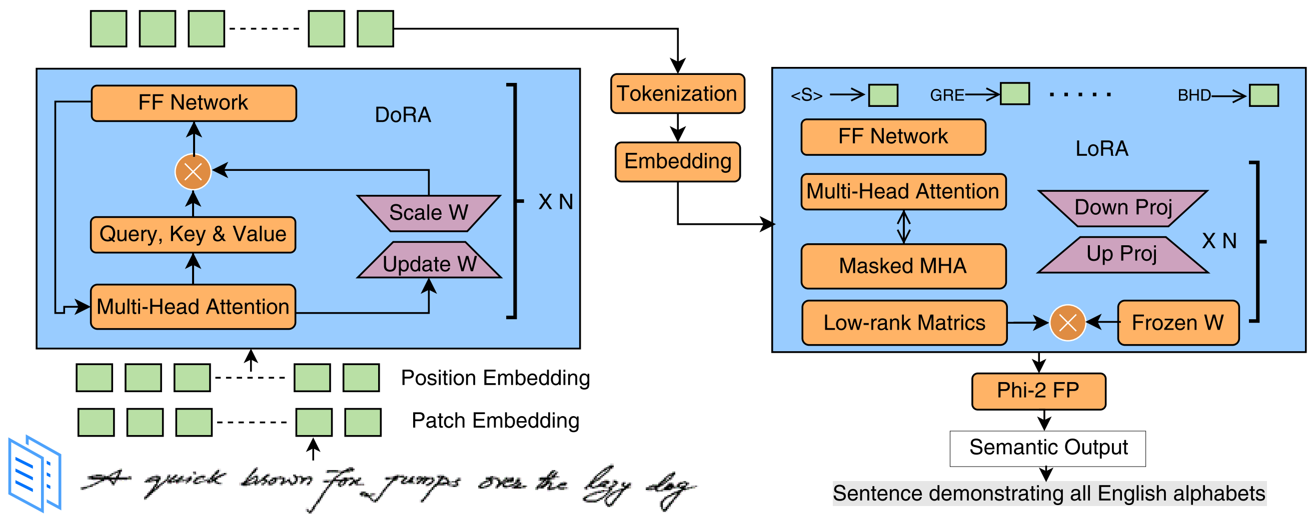

Fig. 2. Detailed architecture of the framework

3.1. Preprocessing

Preprocessing is performed to ensure robust recognition of the textual contents contained in an input image. Initially, a raw input image I, $I \in \mathbb{R}^{h \times w \times c}$ is resized (r), gray-scaled (g), and normalized (n) to have a standard grayscale image I_{pr}

$$I_p = f_n\left(f_g\left(f_r(I)\right)\right). \tag{7}$$

To obtain a consistent-size image, image resizing is performed. Using a standard scaling factor F, image I of height h and width w is resized to a height h' and width w'

$$F = \frac{h/w}{h'/w'}. \tag{8}$$

As shown in Fig. 4, proper pixel intensity is retained with bicubic interpolation (B_I)

$$I_r = B_I(I, h', w'). \tag{9}$$

The standard luminance weight vector V is applied to the input image to obtain the gray-scale image – I_G

$$I_{G(x,y)} = V.I(x,y). \tag{10}$$

To have computationally efficient transformer-based encoding, illumination artifacts are to be avoided. Therefore, denoising (D) is performed after contrast enhancement with the Gaussian function G_{Cont} as

$$I_{G(x,y)} = D\left(G_{\text{Cont}}\left(I_{G(x,y)}\right)\right). \tag{11}$$

The z-score normalization method is exploited to represent the pixel values of I_G into a standard range of $[0,1]$. Taking image I_G with channel c, a normalized image I_N is obtained as

$$I_N = \frac{I'_G - \mu}{\sigma}, \tag{12}$$

where

$$\mu = \frac{1}{N}\sum_c I'_G, \tag{13}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_c \left(I'_G - \mu\right)^2}. \tag{14}$$

Binary mask with Sauvola threshold over window w and dynamic range r is computed as

$$I_B = \mu_w\left(I'_G\right)\left[1 + k\left(\frac{\sigma_w\left(I'_G\right)}{r} - 1\right)\right]. \tag{15}$$

The standard Swin transformer is exploited to obtain visual embeddings. Unlike the other vision transformers, which perform global self-attention, Swin employs a hierarchical structure with shifted windows. Thus, efficient modeling of the local and global dependencies is ensured. The decoder converts visual features from the Swin encoder into a discrete token sequence. With its pretrained variants of handwritten text, EDU exploits the encoder–decoder paradigm, and textual tokens are generated from the input image features.

The processed image I_p is fed to the HTR engine of the framework, where a hybrid pipeline of Transformer and DoRA is used for robust extraction of handwritten text. The HTR function maps the image to a sequence of textual tokens-words, subwords, and letters of the vocabulary V. To enable the capturing of local and global context from the images, the multi-head self-attention (MSA) approach is followed in encoding and decoding. Therefore, the input image is partitioned into 'm' shifted windows – w_m. The windows obtained are non-overlapping with a shifting mechanism as illustrated in Fig. 5. The input image of

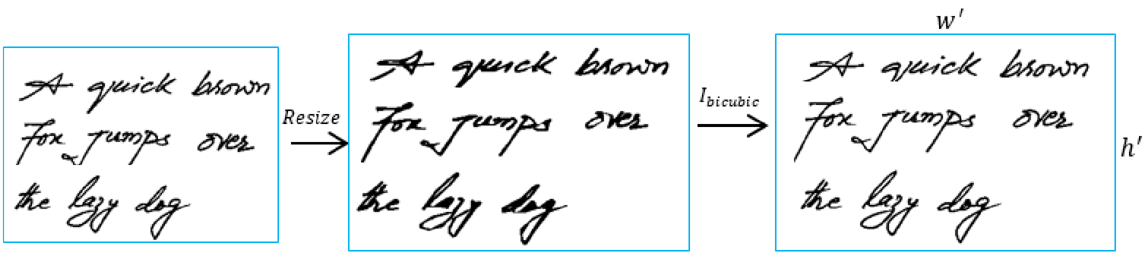

Fig. 4. Image resizing followed by bicubic interpolation

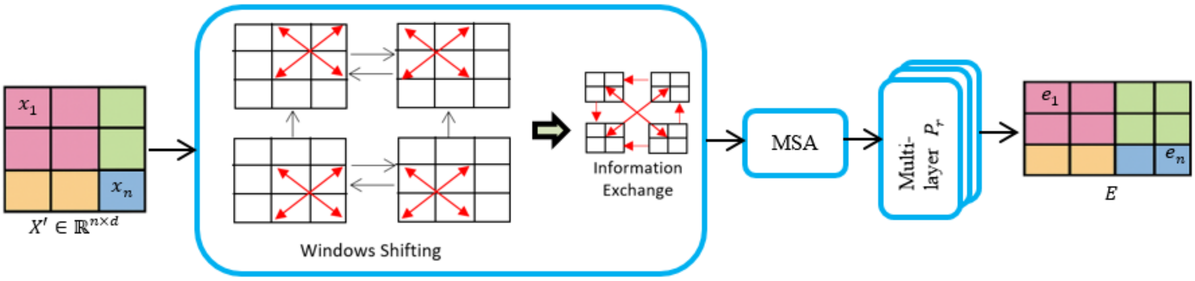

Fig. 5. Visual embedding via shifted-window and multi-head self-attention

a handwritten text is $I_p \in \mathbb{R}^{h\times w\times c}$, where w, h and c represent height, width, and the number of color channels, respectively.

3.2. Embedding generation engine

After preprocessing, as shown in Fig. 6, an image I_p is divided into patches

$$I_p \quad \rightarrow \quad \{p_1, p_2, \ldots, p_N\} \quad p \in \mathbb{R}^{p_s \times p},$$

where p_s is the patch size and $N = \dfrac{hw}{p^2}$.

Flattening and linear projection of each patch is performed to have a vector of patches – $v(p_i)$ and embeddings e_i for transformer-compatible processing

$$e_i = M_e \cdot v(p_i) + b, \tag{16}$$

where M_e is the projection matrix and b is the bias term. To capture patterns of higher order, passing the window w_i to MSA is followed by a multi-layer perceptron P_r. Layer normalization (ln) is employed to retain standardization in training while obtaining embedding e of d size

$$e_i = MSA(w_i) + P_r(\ln(w_i)). \tag{17}$$

The set of embeddings $E = \{e_1, e_2, \ldots, e_n\}$, where $e_i \in \mathbb{R}^d$ for each input sequence i, is utilized for effective recognition.

With the multi-head self-attention (MHSA), the encoder E_o maps the image to a sequence of latent visual embeddings

$$E_o(I) = \{e_1, e_2, \ldots, e_n\} \quad e_t \in \mathbb{R}^d, \tag{18}$$

where d denotes the embedding dimension, and t is the number of patches or tokens derived from the image. For each MHSA, the set of embeddings $E = \{e_1, e_2, \ldots, e_n\}$ is projected into an attention space D, with query q, key k and value v, as

$$q = EM_q, \quad k = EM_k, \quad v = EM_v,$$

where M_q, M_K, and M_v are the learnable projection matrices. The attention coefficient $\mu i \in [0,1]$ is computed using the scaled dot-product of the queries and keys for each node $n_i \in N$ as

$$\mu_i = \text{Softmax}\left(\frac{\langle q_T^i, k_T^i \rangle}{\sqrt{s_\mu}}\right), \tag{19}$$

where s_μ is the stabilizing gradient.

This allows the model to learn a direct mapping from visual patches to character-level tokens, leveraging both multi-head attention and cross-attention mechanisms.

3.3. Text recognition engine

The Transformer-based Text Recognition Engine (TRE) is designed as a sequence prediction module that converts visual representations into a sequence of characters. The engine takes as input the embeddings of visual tokens obtained from the Swin-based feature extractor. The embeddings enable modeling of the global contextual relationships among spatially distributed stroke components.

To cope with the spatial variability, long-range dependencies across the visual tokens are captured through attention mechanisms. The processed representations are subsequently utilized to predict the corresponding character sequence.

To attain adaptability and computational efficiency, Low-Rank Adaptation (LoRA) modules are integrated within the attention layers. Without full model retraining, low-rank updates are introduced to the attention weights instead. The hierarchical, multi-scale feature maps of the Swin encoder are projected and organized into a sequence of tokens. The decoder consumes the embeddings via cross-attention. The set of dense spatial feature maps obtained as a result of the Swin function is represented as

$$f_{\text{swin}}(I_p) = \{t_1, t_2, \ldots, t_n\}. \tag{20}$$

The tokens are subsequently transformed into the decoder embedding space through reshaping and linear projection. Spatial flattening and linear projection are performed to bring the output of the transformer into the embedding space d of the decoder

$$\overline{E}_x = \text{reshape}(T_x, d_x), \tag{21}$$

$$P_T = \overline{E}_x \emptyset_P + 1b_p, \tag{22}$$

where $\emptyset_P$ and b_p are the learnable projection parameter and projection bias.

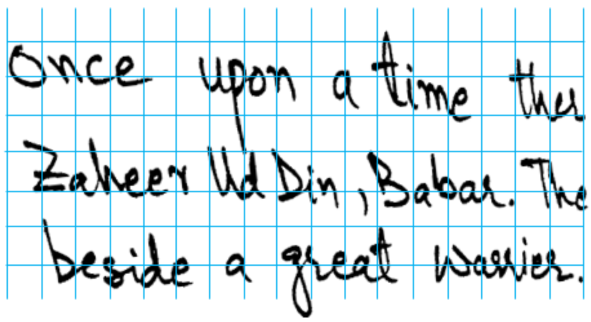

Fig. 6. Splitting the input image into patches for onward compatible processing

At each stage i, a 2-D positional embedding vector ε is added to attain spatial position – $\overline{P}_{T_i}$

$$\overline{P}_{T_i} = P_{T_i} + \varepsilon_i. \tag{23}$$

Tokens of the multiple stages are combined into a single order sequence C by Cross-scale pooling using the learnable query matrix M_q

$$C_i = \text{softmax}\left(M_{q_i}\left(\overline{P}_{T_i}\right)\right) P_T. \tag{24}$$

Let the ground-truth text sequence be S: $S = \{s_1, s_2, \ldots, s_N\}$ where $s_i \in V$, embedding is performed by one-hot encoding

$$e_i = M_e \,\text{One}-\text{hot}(s_i), \tag{25}$$

where M_e is the embedding matrix which represents the learnable vector representation of the tokens of dimension d; $M_e \in \mathbb{R}^{d\times|V|}$. The work of the text recognition engine is presented in Fig. 7.

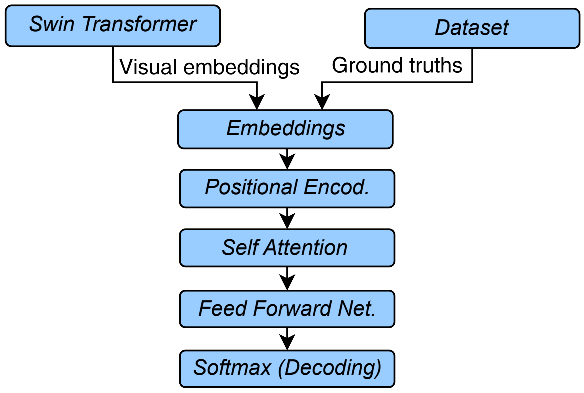

Fig. 7. Workflow of the text recognition engine

Order information is injected through positional encoding p_e for the sequence of length T

$$x_i = e_i + p_e\,, \quad i = \{1,2,3,\ldots,T\}\,.$$

If q, k, v and w are the query, key and projection matrix, the auto-regression at layer l with input set $X^l = \{x_1, x_2, \ldots, x_T\}$ is represented as

$$\begin{aligned} q^l &= X^{l-1} w_q\,, \quad k^l = X^{l-1} w_k\,, \quad v^l = X^{l-1} w_v\,, \\ A^l &= \text{Softmax}\left(\frac{q^l k^l}{\sqrt{d}} + M\right) v^l, \end{aligned} \tag{26}$$

where M is the mask matrix, and d is the dimensional space.

The final update h at layer l using the feed-forward network *FFN* is given as

$$h^l = A^l + \text{FFN}\left(A^l\right). \tag{27}$$

Overall, TRE establishes a structured transformation from visual representation to textual sequence. The joint capture of spatial dependencies and relationships of sequential character improves recognition of complex handwritten scripts.

3.4. Fine-tuning with LoRA and DoRA

Due to a larger number of parameters associated with handwritten text, full fine-tuning remains computationally expensive and prone to overfitting. Therefore, following the encoder-decoder architecture, parameter-efficient fine-tuning (PEFT) is exploited using the strategies of LoRA and DoRA. The technique supports low-rank adaptations and task-specific learning without updating the full set of model parameters. For this purpose, trainable low-rank matrices are injected into frozen pretrained weights, ensuring task-specific adaptation with fewer parameters. In the proposed framework, DoRA is employed in the Swin Transformer to stabilize image feature adaptation of visual features. With the DoRA-based encoder, variations in style and pattern of strokes are efficiently captured. Similarly, LoRA modules facilitate sequence generation by refining attention mechanisms during decoding. Employing LoRA in the Phi-2 language model, text representations are performed with minimal parameter overhead. The pre-trained Phi-2 weights are loaded into a standard half-precision of 16 (fp16). The weights are kept frozen during adaptation. Only a lightweight set of trainable matrices is used, which constitutes less than 1.9% of the total parameters. Thus, a substantial reduction in memory consumption is ensured to use larger batch sizes. The parameter-efficient adaptation design significantly reduces computational cost while maintaining high recognition performance. Additionally, the tendency of overfitting is minimized without compromising stability in the recognition of handwritten text, where dataset variability is always high.

3.4.1. DoRA-based encoding

Encoding is performed with DoRA, where the pretrained weight matrix W of the Swin transformer with input dimension I_d and output dimension O_d is factorized into row-wise magnitude and direction. The adaptation refines weight vectors and updates only the directional component, see Fig. 8. The i-th row in W; $W_i \in \mathbb{R}^{I_d}$, is fed to DoRA and decomposition is processed as

$$W_i = ||W_i|| \cdot \frac{W_i}{||W_i||}\,, \tag{28}$$

where $||W_i||$ is the scalar norm and $\dfrac{W_i}{||W_i||}$ being the unit vector.

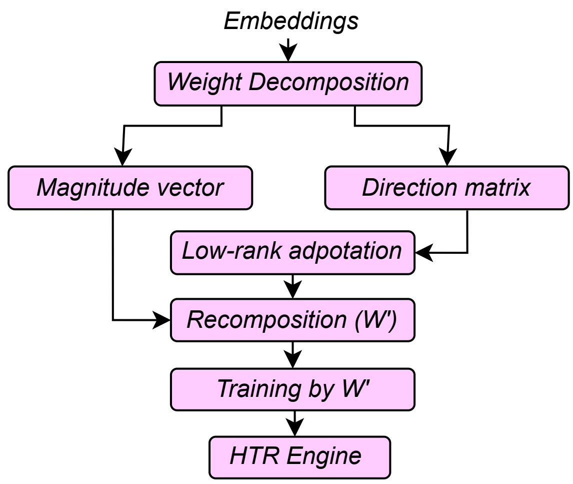

Fig. 8. DoRA based encoding of the textual embeddings

In the adaptation, the directional unit is modified as

$$\frac{W_i}{||W_i||} \to \frac{W_i}{||W_i||} + \Delta\gamma_i,$$

$$\Delta\gamma = \alpha\beta^T, \tag{29}$$

$$\check{W}_i = ||W_i|| \cdot \left(\frac{W_i}{||W_i||} + \Delta\gamma_i\right). \tag{30}$$

For further stability, re-normalization of the new direction is performed

$$\check{W}_i = \|W_i\| \cdot \left(\frac{\dfrac{W_i}{W_i\|} + \Delta\gamma_i}{\left\| \dfrac{W_i}{\|W_i\|} + \Delta\gamma_i \right\|} \right). \tag{31}$$

Thus, magnitude is preserved in adaptation while stacking all the rows of the weight matrix, yielding

$$\check{W} = \begin{bmatrix} \check{W}_1^T \\ \check{W}_2^T \\ \vdots \\ \check{W}_{O_d}^T \end{bmatrix}. \tag{32}$$

The residual adaptation mechanism guarantees mitigating the domain shifts between different handwritten styles and backgrounds in training and inference datasets.

3.4.2. LoRA-based decoding

The lightweight fine-tuning of the textual representation is performed via low-rank matrix updates. LoRA is applied to the self-attention and feed-forward projection matrices of the language model. During decoding, the low-rank adaptations are merged with the original weights for the modified representations required for downstream tasks, see Fig. 9. Instead of fully updating the pretrained weights W, LoRA freezes the matrices and introduces low-risk updates. Let I_d be the input dimension and O_d the output dimension

$$W \in \mathbb{R}^{I_d \times O_d}.$$

The low-range update term is computed through the skinny weight matrices α and β as

$$\Delta W = \alpha\beta^T, \tag{33}$$

where $\alpha \in \mathbb{R}^{I_d \times r}$, $\beta \in \mathbb{R}^{I_d \times r}$ and $r \ll \min(I_d, O_d)$.

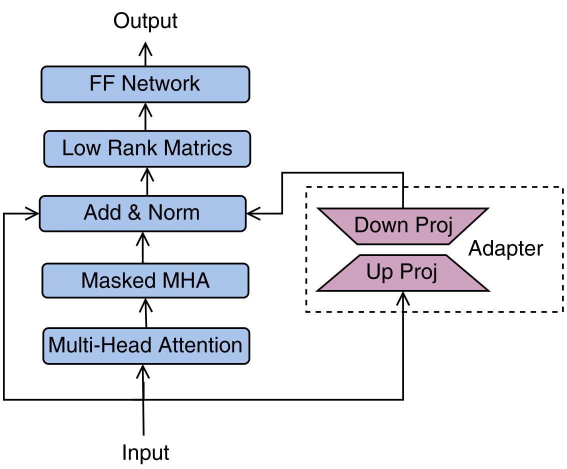

Fig. 9. LoRA based decoding with low-rank adaption

The weight used during fine-tuning is given as

$$\check{W} = W + \alpha\beta^T. \tag{34}$$

The output O during the forward pass is given as

$$O = \check{W}x = (W + \Delta W)\,x. \tag{35}$$

Thus, with intrinsic rank r: $r \ll \min(I_d, O_d)$, the small number of trainable parameters $O(r(I_d, O_d))$, task-specific adaptation is ensured to avoid overfitting while retaining generalization.

3.5. Semantic understanding engine

Once the text is extracted and recognized by the HTR engine, emphasis is paid on text refinement and understanding. Contextual dependencies across tokens are analyzed for semantic understanding. The standard language understanding model Phi-2 is exploited to serve as the language backbone for understanding the extracted text. Phi-2 is an autoregressive language model that learns the probability distribution over sequences of tokens. Initially, the output obtained from TRE is first converted into token IDs compatible with the Phi-2 tokenizer. Each token is then mapped into a high-dimensional embedding using the pretrained layer of Phi-2. The Lightweight matrices of LoRA are injected into the attention projection layers of Phi-2, see Fig. 10. With the multi-head self-attention, the adapted Phi-2 transformer processes the token sequences for generating semantically aware text.

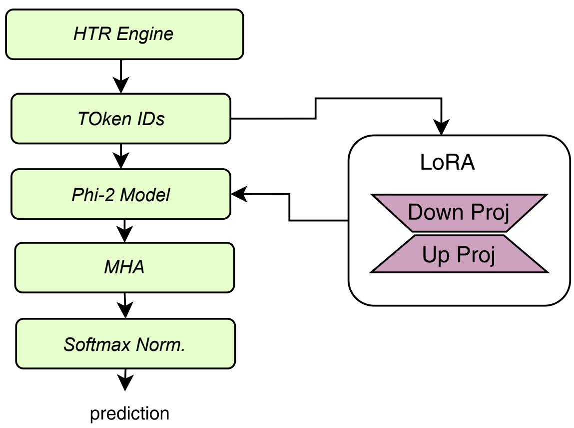

Fig. 10. The key processes involved in the working of SUE

The set of text sequences recognized by the HTR engine – $S = \{s_1, s_2, .., ., s_T\} \mid s_t^T \in V$, is mapped into a continuous embedding space with M_{emb} matrix of embeddings as

$$e_t = M_{\mathrm{emb}} \cdot \mathrm{one\_hot}(s_t) \in \mathbb{R}^d. \tag{36}$$

As positional encoding significantly matters in language understanding of 'he eats' ≠ 'eats he', the vector of positional encoding vpe_t is added to the corresponding embedding e_t to have mixed embedding m_t

$$m_t = e_t + vpe_t. \tag{37}$$

The set of mixed embeddings ME, fed to the first layer of the Phi-2 transformer, is given as

$$ME^0 = [m_1, m_2, \ldots, m_T]. \tag{38}$$

A two-layer feed-forward network (FFN) is used to process each position independently. Taking embedding m_t, the decoder block with cross attention (CA) and self-attention (SA) for the l-th decoder layer (D^l) is given as

$$D^l = \mathrm{FFN}\left(CA\left(SA\left(D^{l-1}\right), m_t\right)\right). \tag{39}$$

At each decoding step, a token with maximum probability from the distribution is produced by the decoder for embedding vector v with the latent space LS

$$y = \underset{m_t \in ME^0}{\arg\max}\, P\left(m_t \middle| m_{<t}, LS\right). \tag{40}$$

The probability $P(m_t||m_{<t}, LS)$ is computed by the Softmax decoder function as

$$P\left(m_t \middle| m_{<t}, LS\right) = \frac{\exp\left(N_\theta(m_t, m_{<t}, LS)\right)}{\sum\limits_{y \in V} \exp\left(\exp\left(N_\theta\left(y, m_{<t}, LS\right)\right)\right)}, \tag{41}$$

where N_θ is the encoder network, parameterized by θ.

At the last layer L, the decoder projects the hidden state of the last token $D^L_{m_t}$ into vocabulary space

$$P\left(s_{m_t+1} \middle| s_{1:m_t}, ME\right) = \mathrm{Softmax}\left(M_o D^L_{m_t}, +b\right), \tag{42}$$

where M_o is the projection matrix and b is the bias term, and

$$P\left(m_t \middle| ME\right) = \pi^N_{n=1} P\left(m_t \middle| m_{<t}, ME; \emptyset\right). \tag{43}$$

If y^*_t is the ground token at time t, and ω is the set of trainable parameters, the collective loss $L(\omega)$ is minimized as

$$L(\omega) = -\sum_{t=1}^{K} \log P\left(m^*_t \middle| m^*_{<t}, S_T; \omega\right), \tag{44}$$

where K is the length of the ground-truth sequence.

3.6. Modeling uncertainty

To augment reliability in handwritten recognition, Bayesian inference via Monte Carlo Dropout (MC-Dropout) is integrated into the decoding phase. Bayesian inference is the standard approach to estimate uncertainty of a neural network model. Instead of a single deterministic forward pass, dropout is activated during inference, and the model parameters are transformed into probabilistic distributions, yielding stochastic predictions. Therefore, to estimate the posterior distribution Ω, the dataset D and parameter π of the model are used

$$\Omega(\pi|D) = \frac{\Omega(D|\pi)\,\Omega(\pi)}{\Omega(D)}. \tag{45}$$

The model output – y_i is evaluated with each stochastic forward pass for each new input I_i

$$\Omega(y_i|I_i, D) = \int \Omega(y_i|I_i, \pi)\,\Omega(\pi|D)\,\mathrm{d}\pi. \tag{46}$$

As the dropout is applied at inference time, with each pass, samples with a different set of parameters are obtained, see Fig. 11. If $Y^{\vartheta_t} = \{y^{\vartheta_1}, y^{\vartheta_2}, \ldots, y^{\vartheta_T}\}$ is the set of predictions where y^{ϑ_i} is computed with a different dropout mask, the predictive mean is computed as

$$\mu_y = \frac{1}{\vartheta_T} \sum_{\vartheta_t=1}^{\vartheta_T} y^{\vartheta_t}. \tag{47}$$

By taking μ_y of the last layer, the epistemic uncertainty is obtained through predictive variance given as

$$\sigma_y^2 = \frac{1}{\vartheta_T} \sum_{\vartheta_t=1}^{\vartheta_T} \left(y^{\vartheta_t} - \mu_y\right)^2. \tag{48}$$

Optimizing the dedicated components – EGE, TRE, and SUE in a unified architecture, the EDU framework enhances both recognition accuracy and contextual coherence. Special emphasis is placed on learning discriminative features by prioritizing stroke-relevant regions during the context-aware modeling. Moreover, uncertainties of ambiguous inputs are adjusted using the Bayesian inference mechanism to improve robustness. The integrated strategy promises improved resilience to noise, and stylistic variations commonly observed in handwritten text.

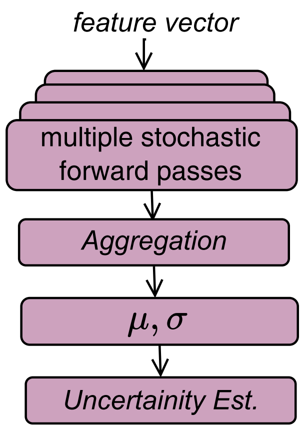

Fig. 11. The process of uncertainty estimation using the Bayesian inference

4. EXPERIMENTATION AND ANALYSIS

Systematic experiments are conducted to accurately assess the performance of the framework. There are two distinct phases of the evaluation. In the first phase, recognition performance is assessed on the benchmark dataset. In the second phase, reliability of the system is gauged by real-world handwritten scripts. The deterministic metrics of Character Error Rate (CER), Word Error Rate (WER), and Expected Calibration Error (ECE) are used for examining the performance of the system. This experimental design provides not only a fair comparison with existing state-of-the-art models, but also a comprehensive understanding of how architectural refinements and uncertainty modeling contribute to robustness and generalizability. The two standard error rates, CER and WER, are utilized. The CER measures the proportion of character-level errors between the predicted transcription Y^{ϑ} and the ground-truth sequence Y. It is computed as the normalized Levenshtein distance

$$\mathrm{CER} = \frac{S_c + D_c + I_c}{N_c}, \tag{49}$$

where N_c is the total number of characters in the Y and S, D and I are the number of substitutions, deletions, and insertions. WER assesses the word-level transcription quality for catching the semantic correctness and is defined as

$$\mathrm{WER} = \frac{S_w + D_w + I_w}{N_w},. \tag{50}$$

The metric Expected Calibration Error (ECE) is employed to measure how much the model confidence (Cc) aligns with empirical accuracy (Ac). If n is the total number of predictions and J the total number of bins-B, ECE is defined as

$$\mathrm{ECE} = \sum_{j=1}^{J} \frac{|B_i|}{n} \left| Ac\left(B_j\right) - Cc\left(B_j\right) \right|. \tag{51}$$

Each bin B_i is computed using the predicted confidence as

$$B_j = \left\{ i \middle| pc_i \in \left(\frac{j-1}{J}, \frac{j}{J} \right) \right\}, \quad j = 1, 2, \ldots, J, \tag{52}$$

where the confidence and accuracy for the bin are given as

$$Cc(B_j) = \frac{1}{|B_j|} \sum_{i=B_j} pc_i, \tag{53}$$

$$Ac(B_j) = \frac{1}{|B_j|} \sum_{i=B_j} 1\left(y_i^{\vartheta} = y_i\right). \tag{54}$$

4.1. Datasets description

Besides the standard IAM handwriting dataset, a custom collection of real-world handwritten samples was used in the study. The custom dataset capturing different writing styles comprised educational documents such as classroom notes, essays, and examination scripts. The data acquisition process involved collecting handwritten samples from multiple contributors, including students and educators, ensuring diversity in handwriting patterns. The sampling process was conducted in a proper way to ensure readability while ensuring variability in handwritten text. Contributors were asked to write naturally without constraining writing style or spacing. The custom dataset contains 73 documents, each containing two separate paragraphs with balanced representation of pangrams.

All images of the dataset were digitized at a resolution of 300 DPI and normalized to a fixed size of 220×220 pixels to ensure consistency. Standard preprocessing steps were applied to reduce variations in scale, illumination, and noise. The dataset was partitioned into training, validation, and testing subsets using a standard split ratio of 80:10:10. The stratified division guarantees balanced representation across the subsets for reliable performance evaluation.

4.2. Implementation details

For the implementation and testing, a system running with an Intel Core i7-12400F CPU, 16 GB RAM, and a 12 GB VRAM NVIDIA GPU was used. For image processing, the library of OpenCV is used, whereas for the ML-based operations, PyTorch is utilized. The model was trained using a batch size of 16, indicating that in each iteration 16 samples were processed. Training was conducted for 800 epochs, where each epoch corresponds to a complete pass over the training split. The use of parameter-efficient fine-tuning through LoRA and DoRA significantly reduced computational complexity, thereby securing a reduction in training time. With individual epochs lasting for 0.75 minutes, the training was completed in approximately 10 hours. The Adam optimizer was employed with an initial learning rate of 1×10^{-4} and a weight decay of 1×10^{-5} to improve generalization. Moreover, to preserve impartiality, a separate test set entirely independent from the training set was used for the final performance evaluation.

4.3. Results and analysis

Experimental evaluation of the framework is conducted using the standard IAM handwriting Database along-with the real-world handwritten text. Initially, the system is assessed by the test set of the dataset using the basic metrics of precision, recall, accuracy, and F1 score. As shown in Table 1, the scores obtained against the sole implementation are comparatively low in F1 score and accuracy. However, progressive improvement is achieved with each enhancement to the baseline model. Moreover, the integration of Bayesian inference via MC-Dropout further improved the model robustness, leading to the highest F1 score of 0.95. These results confirm that the proposed methods improve recognition accuracy and enhance the model ability to manage the inherent variability in handwritten text.

Table 1
Statistics of the evaluation performed using the test set of the IAM dataset

Model variant	Precision ↑	Recall ↓↑	F1 ↑	Accuracy ↑
LoRA	0.92	0.912	0.916	0.91
DoRA	0.928	0.918	0.922	0.92
LoRA + DoRA	0.939	0.929	0.934	0.93
LoRA + DoRA + Bayesian Inference	0.963	0.954	0.958	0.95

Discriminative ability of the model is depicted by the ROC–AUC analysis across various decision thresholds. As shown in the plotted curves (Fig. 12), the progressive enhancement over the baseline reflects consistent improvements in true

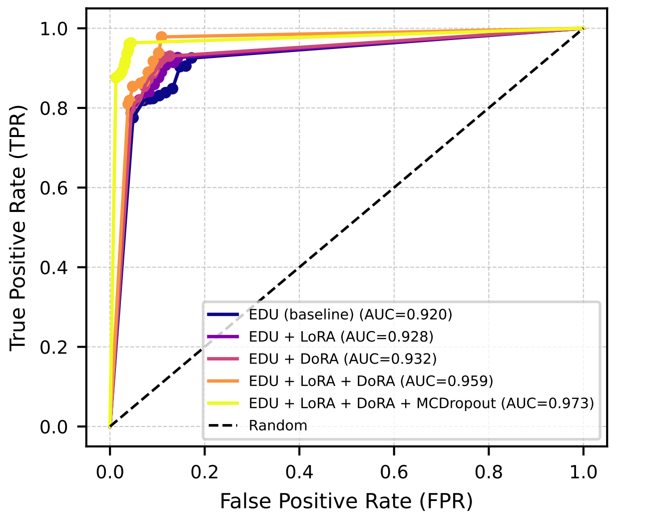

Fig. 12. Results of the discriminative ability using the ROC–AUC curve

positive rates and reduction in false positive rates. Notably, the variant incorporating LoRA, DoRA, and Bayesian inference achieved a macro AUC of 0.97. As a whole, the analysis indicates a promising separability between correct and incorrect character recognition. This high AUC score justifies that the model performs well not only at a single threshold but maintains robust classification across a range of thresholds with strong generalization capability.

For further in-depth assessments, the customary metrics of language recognitions CER, WER and ECE are exploited. The quantitative outcomes for CER and WER are analyzed separately using the following formulas:

$$\mathrm{CER} = \frac{\sum_{i=1}^{N} d_c(y_i, y_i')}{\sum_{i=1}^{N} |y_i^c|}, \tag{55}$$

$$\mathrm{WER} = \frac{\sum_{i=1}^{N} d_w(y_i, y_i')}{\sum_{i=1}^{N} |y_i^w|}, \tag{56}$$

where d_c and d_w denote the character-level and word-level Levenshtein distances.

The computed scores are analyzed against the two standard baseline systems of [26] and [27]. Statistics of the analysis are presented in Table 2.

Besides, the performance of the system is evaluated using real-world handwritten English text. To cover all the characters, pangram sentences like '*a quick brown fox jumps over the lazy dog*' containing all the alphabets are written by 213 college students on white paper. With an ordinary camera, images of the handwritten texts are captured and fed to the framework. Out of 112 038 characters from 426 sentences, 106 212 characters are

Table 2
Results of the evaluation against the baseline methods using the key language metrics (CER and WER are in %)

Model variant	CER ↓	WER ↓	ECE ↓	Entropy ↓
Base architecture (AVg)	2.1	8.2	–	–
EDU with LoRA	4.14	11.02	0.081	0.388
EDU with DoRA	3.07	10.91	0.077	0.362
EDU with LoRA & DoRA	2.93	9.88	0.072	0.341
With LoRA, DoRA and Bayesian Inference (MC-Dropout)	**2.41**	**8.13**	**0.061**	**0.319**

properly recognized, obtaining a promising accuracy of 94.83% and an F1 score of 0.925 (see the confusion matrices in Fig. 13 and Fig. 14).

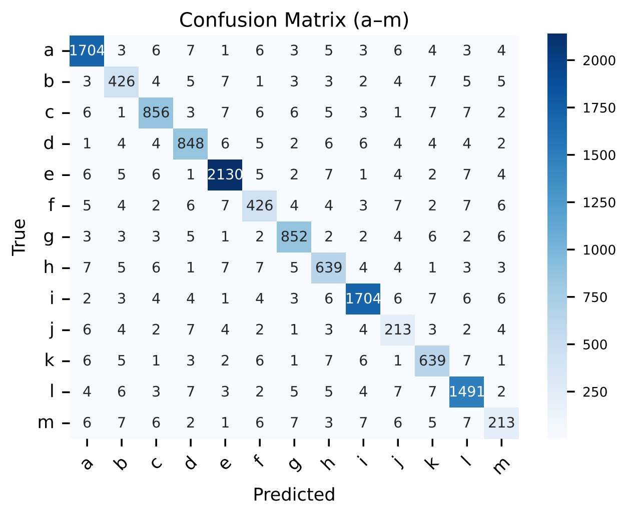

Fig. 13. Confusion matrix illustrating predictive performance of the system for the characters a-m

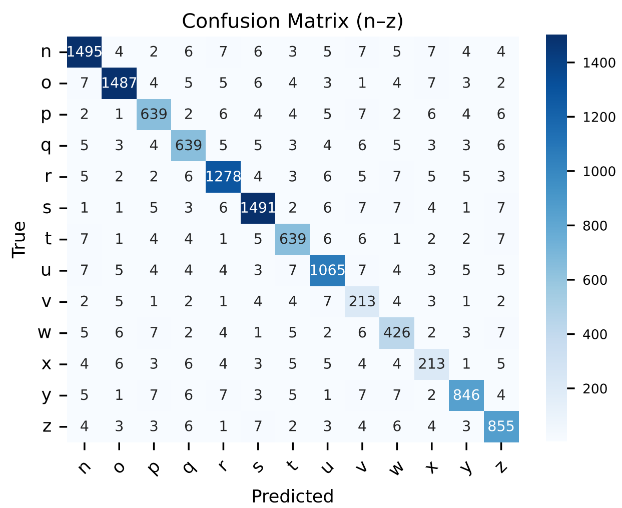

Fig. 14. Confusion matrix illustrating predictive performance of the system for the characters n-z

With the Receiver Operating Characteristic (ROC) curve, the discriminative capability of the framework for all the character classes is evaluated. As depicted by the curve (see Fig. 15), most classes exhibit high TPR values with a negligible FPR. The class-wise ROC items, visualized by distinct markers, reveal that lower recall points reside near the ideal region of the curve. This confirms the overall reliability and stability of the framework. The satisfactory AUC value of 0.86 indicates that the model performs considerably better than random guessing. Though overall a robust recognition performance is achieved, alphabets having similar shapes are misrecognized by the system.

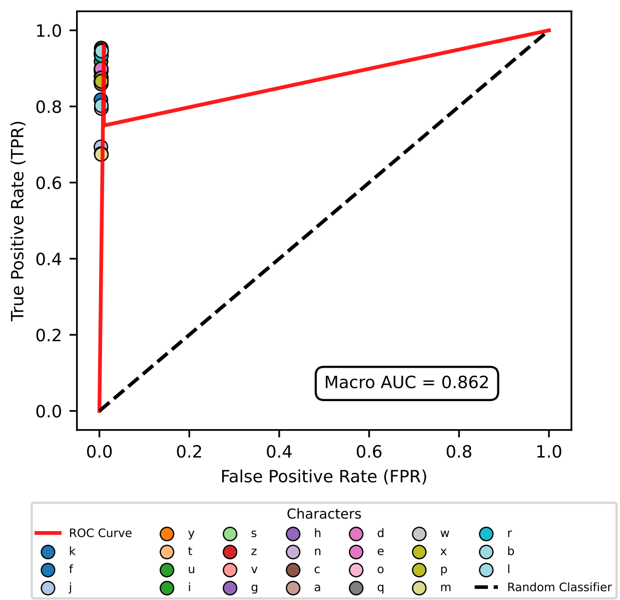

Fig. 15. Characters classification analysis using ROC

4.4. Ablation study

A comprehensive ablation study is conducted to assess the contribution of individual components in architecture. The two latest and standard research works of [26] and [27] are taken as the baselines. We systematically removed key modules LoRA, DoRA, and Bayesian Inference, while retaining all the other settings. As revealed in Table 3, omitting LoRA or DoRA leads to a significant drop in performance. This highlights the critical role of fine-tuning parameters for the encoder and decoder. However, the absence of Bayesian Inference marginally affects the efficiency of the framework. In the analysis, the base system with parametric model f taking input I is represented as

$$f_{\vartheta} = I \;\rightarrow\; Y, \tag{57}$$

where $Y = \{y_1, y_2, \ldots, y_n\}$ is the predicted sequence and ϑ is the model parameter.

If M is the error indicator, the error rate $E(M)$ for the ablation study is computed as

$$E(M) = \frac{1}{N}\sum_{i=1}^{N} 1\left[y_i \neq y_i'\right]. \tag{58}$$

The Bayesian inference using the Shannon entropy S, estimates uncertainty of the model prediction for the n-th token as

$$S(\widehat{y}_n) = -\sum_{c=1}^{C} P\left(y_n = c \middle| I, \vartheta\right) \log P(y_n = c | I, \vartheta). \tag{59}$$

To further assess the effectiveness of PEFT on recognition accuracy, the standard Character Accuracy Rate (CAR) is computed using the relation: $\mathrm{CAR} = (1 - \mathrm{CER}) \times 100$. The impact of LoRA, DoRA and Bayesian inference is checked independently and jointly.

The analysis indicates that with full-parameter, EDU achieves a high CAR. With the isolated use of LoRA or DoRA, a noticeable degradation is reported, which indicates that low-rank adaptation alone may constrain the representation capacity. However, when LoRA and DoRA are combined, a satisfactory recovery is observed, showing a more stable convergence in handwritten English text recognition. As a whole, the analysis validates the significance of each module in the EDU framework, see Fig. 16.

4.5. Comparative analysis

Since the last decade, multiple prominent HTR systems have been added in the literature. To contextualize the performance of the proposed framework, some domain-specific state-of-the-art methods are selected for comparative analysis. The studies provide a diverse set of baselines for rigorous judgement. For instance, the DNTextSpotter presented by Xie *et al.* [28] demonstrates robust performance in arbitrary-shaped scene text spotting, with noteworthy precision. Similarly, the transformer-based system of Huang *et al.* [7] underscores the effectiveness

Table 3
Performance comparison of the model variants based on the standard metrics

Model variant	CER ↓	WER ↓	ECE ↓	Entropy ↓	Precision ↑	Recall ↑	F1 ↑	Accuracy ↑
Baseline (Avg.)	2.12	8.23	1.321	–	0.902	0.89	0.896	–
LoRA	5.14	14.02	0.081	0.325	0.92	0.912	0.916	0.91
DoRA	5.07	13.91	0.077	0.318	0.928	0.918	0.922	0.92
LoRA + DoRA	4.93	12.88	0.072	0.304	0.939	0.929	0.934	0.93
LoRA + DoRA + Bayesian Inference	2.41	8.13	0.061	0.291	0.963	0.954	0.958	0.95

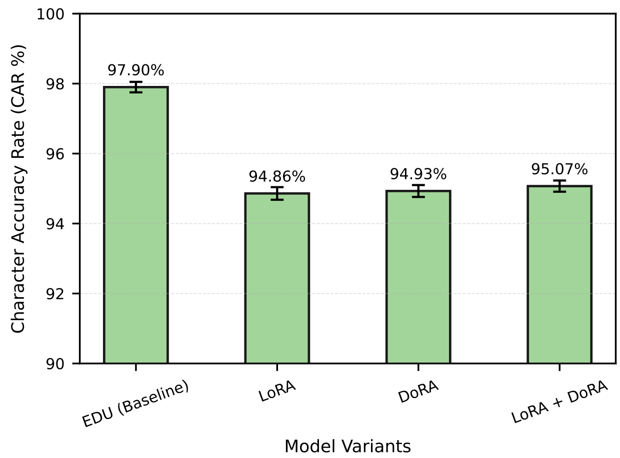

Fig. 16. Significance of modules in terms of CAR

of Swin backbones in structured textual tasks. As reviewed in Table 4, EDU accomplishes superior precision, recall, and F1-score, outperforming the prominent transformer-based models. The systems of TrOCR [21] and ABINet++ [29] rely on large-scale pretraining despite their strong language modeling capabilities. Similarly, PARSeq [30] demonstrates strong flexibility, but lacks robustness in complex handwritten scripts. Although text spotters like DNTextSpotter [28] and SwinTextSpotter [7] excel in handling arbitrary-shaped text, their performance often drops on cursive and low-resource scripts. Lukasik *et al.* [31] employed a CNN for Latin script recognition, while PLATTER [32] provides a valuable page-level benchmark for Indic scripts but requires extensive manual preprocessing. The LOGO system of Liu *et al.* [33] suffers from high computational overhead, which may limit its practicality for large-scale applications. The proposed EDU framework integrates end-to-end detection and recognition without extensive overhead. Although EDU exhibits an insignificant weakness in disambiguating similar shape characters, the system is robust and computationally efficient to be used for large-scale text recognition and understanding.

5. CONCLUSIONS AND FUTURE WORK

Handwritten text recognition (HTR) is substantially important in academic documents understanding, and information retrieval. Due to the diversity in writing styles and irregular layouts, proper recognition of human handwriting is significant challenge to be resolved. Most of the existing systems are limited by large-

Table 4
Statistics of the comparative analysis of the standard approaches (NR means Not Reported)

S. No.	Paper (Ref)	Year	Dataset(s) used	Precision (%)	Recall (%)	F1/H-mean (%)	Remarks
1	TrOCR: Transformer-based Optical Character Recognition [21]	2023	IAM, Bentham, RIMES, real-world scans	95.1	94.5	94.8	Performs well across scripts but requires large pre-training data.
2	PARSeq: Permutation invariant autoregressive scene text recognizer [30]	2022	IAM, IIIT5K, SVT, Synth-Text	93.8	92.1	93	Flexible sequence modeling but lacks explicit layout modeling.
3	ABINet++: Autonomous, bidirectional, and iterative text recognizer [29]	2023	IAM, IIIT5K, ICDAR2015	94	92.5	93.2	Incorporates visual-language modeling for robustness but is computationally intensive for handwritten scripts.
4	DNTextSpotter: Arbitrary-shaped scene text spotting via improved denoising training [28]	2024	CTW1500, ICDAR15, Inverse-Text	92.5	87	90.2	Strong on arbitrary shapes; however, recall drops on curved text.
5	SwinTextSpotter: Scene text spotting via better synergy [7]	2022	SCUT-CTW1500, ICDAR2015, ReCTS	88.0	NR	88	Good for multi-oriented text but mostly relies on shared backbone.
6	PLATTER: Page-Level Handwritten Text Recognition System [32]	2025	CHIPS, Kaggle HTR, IAM	NR	NR	NR	Focuses on page-level baselines but requires manual alignment
7	Recognition of handwritten Latin characters with diacritics using CNN [31]	2020	Polish Handwritten Characters Database (PHCD)	NR	NR	NR	Latency is promising; however, low accuracy for similar-shape characters.
8	LOGO: Video text spotting with language collaboration [33]	2024	ICDAR2013/2015 Video, Minetto, DSText	85.8	67.4	75.5	Strong zero-shot spotting but requires high training cost.
9	EDU (the proposed framework)	2026	IAM + Real-world datasets	96	95	95	Robust and computationally efficient; struggles with similarly shaped characters.

scale pre-training, computational cost, or manual alignment. The emerging transformer-based models and text spotting frameworks offer promising avenues for improving recognition performance. With this study, a hybrid approach combining Swin transformer, and Phi-2 language model is presented for effective extraction and contextual understanding of handwritten English text in images.

The semantic backbone is supported by the parameter-efficient fine-tuning strategies of the Low-Rank Adaptation (LoRA) and Decomposition Rank Adaptation (DoRA). Moreover, the Bayesian inference with Monte Carlo dropout is exploited to measure predictive uncertainty. Apart from its valuable use in academia, the technique can be extended for use in medical report evaluation and legal document analysis. The implemented system of EDU is systematically evaluated using the standard metrics.

Collectively, the results indicate that the system designed on the proposed architecture achieves superior performance compared to the baseline approaches. Comparative analysis also demonstrates that EDU outperforms most of the standard approaches in terms of accuracy and cost. However, low precision is obtained for characters with similar visual structures. Our future work should focus on fusion and contrastive feature learning to enhance character disambiguation of visually similar graphemes. Additionally, we also plan to extend the framework to support cross-lingual learning and new writing styles.

DATA AVAILABILITY

Data is available upon request.

FUNDING

This study has been supported by the 2012 Hubei Provincial Department of Education Key Project in Educational Science Planning (Project No.: 2012A121), 2014 Hubei Vocational Education Research Center Project (Project No.: G2014B024), 2016 Hubei Provincial Department of Education Educational Science Planning Research Projec (Project No.: 2016GB271), 2021 Ministry of Education Vocational College Foreign Language Teaching Guidance Committee Research Project (Project No.: WYJZW-2021-2140), 2025 Ministry of Education.

REFERENCES

[1] T.M. Babczyński and R. Ptak, "Direct tensor voting in line segmentation of handwritten documents," *Int. J. Electron. Telecommun.*, vol. 70, no. 1, pp. 95–102, 2024, doi: 10.24425/ijet.2024.149519.

[2] M. Woda and G. Oliwa, "Authorial approach to the detection of selected psychological traits based on handwritten texts," *Int. J. Electron. Telecommun.*, vol. 70, no. 1, pp. 145–152, 2024, doi: 10.24425/ijet.2024.149524.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, USA, 2015, doi: 10.48550/arXiv.1409.0473.

[4] V. Lebedev and V. Lempitsky, "Speeding-up convolutional neural networks: A survey," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 66, no. 6, pp. 799–810, 2018, doi: 10.24425/bpas.2018.125927.

[5] H. Touvron *et al.*, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*, doi: 10.48550/arXiv.2302.13971.

[6] Y. Rao, W. Zhao, and X. Liu, "Bridging visual and linguistic modalities for handwritten document understanding," *Pattern Recognit. Lett.*, vol. 172, pp. 60–68, 2023, doi: 10.1016/j.patrec.2023.03.018.

[7] T. Huang, M. Lin, S. Huang, and E. Ding, "SwinTextSpotter: Scene text spotting via better synergy between text detection and text recognition," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, USA, 2022, pp. 4593–4602, doi: 10.1109/CVPR52688.2022.00456.

[8] H. Zhang, J. Li, and X. Wang, "Vision transformers for handwriting and document analysis: A survey," *Pattern Recognit.*, vol. 145, p. 109870, 2023, doi: 10.1016/j.patcog.2023.109870.

[9] M. Abdin *et al.*, "Phi-2: The surprising power of small language models," Microsoft Research, Tech. Rep., 2023. [Online]. Available: https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models

[10] E.J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," 2021, *arXiv:2106.09685*, doi: 10.48550/arXiv.2106.09685.

[11] X. Liu, Y. Li, S. Wang, L. Wang, and W. Chen, "DoRA: Weight-decomposed low-rank adaptation for efficient fine-tuning," 2024, *arXiv:2402.09353*, doi: 10.48550/arXiv.2402.09353.

[12] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, New York, USA, 2016, doi: 10.48550/arXiv.1506.02142.

[13] U.-V. Marti and H. Bunke, "The IAM-database: An English sentence database for offline handwriting recognition," *Int. J. Doc. Anal. Recognit.*, vol. 5, no. 1, pp. 39–46, 2002, doi: 10.1007/s100320200071.

[14] R. Plamondon and S.N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 63–84, 2000, doi: 10.1109/34.824821.

[15] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003, doi: 10.1162/153244303322533223.

[16] S. Madhvanath and V. Govindaraju, "The role of holistic paradigms in handwritten word recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 149–164, 2001, doi: 10.1109/34.908962.

[17] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, 2009, doi: 10.1109/TPAMI.2008.137.

[18] A. Vinciarelli *et al.*, "A survey on off-line cursive word recognition," *Pattern Recognit.*, vol. 35, no. 7, pp. 1433–1446, 2004, doi: 10.1016/S0031-3203(03)00201-1.

[19] D.C. Cireșan, U. Meier, L.M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, Beijing, China, 2011, pp. 1135–1139, doi: 10.1109/ICDAR.2011.229.

[20] T. Bluche, "Deep neural networks for large vocabulary handwritten text recognition," *Proc. Int. Conf. Document Analysis and*

Recognition (ICDAR), Nancy, France, 2015, pp. 488–493, doi: 10.1109/ICDAR.2015.7333947.

[21] M. Li et al., "TrOCR: Transformer-based optical character recognition with pre-trained models," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 11, pp. 13094–13102, 2023, doi: 10.1609/aaai.v37i11.26712.

[22] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vienna, Austria, 2021, doi: 10.48550/arXiv.2010.11929.

[23] F. Sheng, Z. Chen, and B. Xu, "NRTR: A no-recurrence sequence-to-sequence model for scene text recognition," *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, Sydney, Australia, 2019, pp. 781–786, doi: 10.1109/ICDAR.2019.00128.

[24] L. Yang, P. Wang, H. Li, Z. Li, and Y. Zhang, "A holistic representation guided attention network for scene text recognition," *Neurocomputing*, vol. 414, pp. 67–75, 2020, doi: 10.1016/j.neucom.2020.07.090.

[25] D. Yu *et al.*, "Towards accurate scene text recognition with semantic reasoning networks," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, USA, 2020, pp. 12110–12119, doi: 10.1109/CVPR42600.2020.01213.

[26] W. Gu, L. Gu, Z. Wang, C.Y. Suen, and Y. Wang, "DocTTT: Test-Time Training for Handwritten Document Recognition Using Meta-Auxiliary Learning," *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Waikoloa, USA, 2025, pp. 1904–1913.

[27] M. Hamdan, A. Rahiche, and M. Cheriet, "HTR-JAND: Handwritten Text Recognition with Joint Attention Network and Knowledge Distillation," 2024, *arXiv:2412.18524*, doi: 10.48550/arXiv.2412.18524.

[28] Y. Xie *et al.*, "DNTextSpotter: Arbitrary-shaped scene text spotting via improved denoising training," 2024, *arXiv:2408.00355*, doi: 10.48550/arXiv.2408.00355.

[29] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "ABINet++: Autonomous, bidirectional and iterative text recognizer for scene text spotting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7069–7082, 2023, doi: 10.1109/TPAMI.2022.3154079.

[30] D. Bautista and R. Atienza, "Scene text recognition with permuted autoregressive sequence models (PARSeq)," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Tel Aviv, Israel, 2022, pp. 178–196, doi: 10.1007/978-3-031-20047-2_11.

[31] E. Lukasik *et al.*, "Recognition of handwritten Latin characters with diacritics using CNN," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 69, no. 1, p. e136210, 2021, doi: 10.24425/bpasts.2020.136210.

[32] B.V. Kasuba *et al.*, "PLATTER: A page-level handwritten text recognition system for Indic scripts," 2025, *arXiv:2502.06172*, doi: 10.48550/arXiv.2502.06172.

[33] H. Liu, D. Sun, J. Wang, Y. Liu, and G. Pan, "LOGO: Video text spotting with language collaboration and glyph perception model," 2024, *arXiv:2405.19194*, doi: 10.48550/arXiv.2405.19194.