A Controlled Comparison of BiLSTM and Transformer Encoders for Arabic Handwritten Word Recognition

Imane Bounour [1,*], Alae Ammour [2], Ghizlane Khaissidi [1], and Mostafa Mrabti [1]

[1] Laboratory LIPI ENS, USMBA Fez, Morocco
[2] Euromed Research Center, School of Digital Engineering and Artificial Intelligence (EIDIA), Euromed University of Fes, UEMF, Morocco
Email: imane.bounour@usmba.ac.ma (I.B.); a.ammour@ueuromed.org (A.A.); ghizlane.khaissidi@usmba.ac.ma (G.K.); mostapha.mrabti@usmba.ac.ma (M.M.)
*Corresponding author

***Abstract*—Despite significant advances in Arabic Handwritten Word Recognition (AHWR), the specific contribution of sequential encoders remains unclear because most prior studies change several architectural elements simultaneously, making comparisons difficult to interpret. This work presents the first carefully controlled evaluation of Bidirectional Long Short-Term Memory (BiLSTM) and Transformer encoders within an identical Convolutional Neural Network-Connectionist Temporal Classification (CNN-CTC) framework. All external factors, including preprocessing, the truncated ResNet-50 backbone, CTC alignment, Word Beam Search (WBS) decoding, and dataset splits, are held constant to isolate the effect of the sequence modeling mechanism itself, a dimension not explicitly analyzed in previous literature. The study also evaluates three augmentation strategies, including uniform augmentation and frequency-aware schemes at both the word and character levels, which address the distributional imbalance inherent in Arabic handwriting datasets. Experiments on IFN/ENIT show that the Transformer consistently achieves higher accuracy, up to 98.91%-Character Accuracy (CAR) and 98.41%-Word Accuracy (WAR), while the BiLSTM offers substantially faster inference. These findings provide the first reproducible quantification of the accuracy-efficiency trade-off between recurrent and attention-based encoders for Arabic cursive handwriting under fully controlled conditions.**

***Keywords*—Arabic handwritten word recognition, ResNet-50, transformer, Bidirectional Long Short-Term Memory (BiLSTM), Connectionist Temporal Classification (CTC), Word Beam Search (WBS), data augmentation**

I. Introduction

Optical Character Recognition (OCR) plays a central role in a wide range of applications, including the digitization of historical archives, automated document processing, and the preservation of cultural heritage [1]. Within this domain, Arabic handwritten text recognition remains particularly challenging due to the intrinsic properties of the script. Arabic handwriting is cursive by nature, exhibits context-dependent character shapes, frequent ligatures, and optional diacritics, all of which introduce strong spatial and sequential dependencies and make recognition especially difficult in unconstrained, real-world settings [2].

Early approaches to Arabic handwritten text recognition relied on traditional machine learning techniques such as Hidden Markov Models (HMMs) [3, 4] and Support Vector Machines (SVMs) [5, 6], which depended heavily on handcrafted features and were limited in their ability to generalize across writing styles and writers [7]. The advent of deep learning has since shifted the field toward end-to-end architectures capable of learning hierarchical visual and sequential representations directly from raw image data, significantly improving recognition performance for cursive scripts.

Among these approaches, the Convolutional Neural Network-Recurrent Neural Network-Connectionist Temporal Classification (CNN-RNN-CTC) pipeline has become a widely adopted framework for handwritten word recognition [8, 9]. In this paradigm, CNN is used to extract high-level visual features [7], RNN-most commonly Bidirectional Long Short-Term Memory (BiLSTM) units-model sequential dependencies [9, 10], and the CTC objective enables alignment-free sequence prediction without explicit character segmentation [11]. While effective, recurrent sequence modeling is inherently sequential, limiting parallelization and increasing training and inference time. Moreover, capturing long-range dependencies remains challenging, particularly for Arabic handwriting, where character interpretation can depend on distant contextual cues across ligatures and elongated strokes [12, 13].

In recent years, Transformer-based architectures have emerged as a powerful alternative to recurrent models for

Manuscript received January 12, 2026; revised January 27, 2026; accepted March 6, 2026; published June 17, 2026.

sequence modeling, relying on self-attention mechanisms to capture global dependencies while enabling full parallelization during training [13]. CNN-Transformer-CTC pipelines have demonstrated strong performance in handwriting and scene text recognition, especially for Latin scripts [14, 15]. However, despite their growing adoption, the behavior of Transformer encoders for Arabic cursive handwriting remains comparatively underexplored. In particular, it is still unclear to what extent self-attention provides advantages over recurrent modeling for Arabic script, and at what computational cost these gains are achieved.

Rather than proposing a novel architecture, this work adopts a hypothesis-driven and comparative perspective. We investigate whether Transformer encoders are better suited than BiLSTM-based models for capturing long-range dependencies, ligature structures, and diacritic-dependent character distinctions inherent to Arabic cursive handwriting, and whether the resulting performance gains justify the associated computational overhead [16].

To enable a rigorous and interpretable comparison, we deliberately focus on isolated word recognition. Although recent studies increasingly emphasize text-line recognition, word-level modeling remains a fundamental setting for analyzing sequence modeling mechanisms in cursive scripts [17, 18]. At the word level, long-range dependencies, ligatures, and diacritic-driven ambiguities between visually similar characters are already present, while additional confounding factors such as inter-word spacing, layout variability, and line segmentation are avoided. This choice allows us to isolate the impact of the sequence encoder and to conduct a controlled analysis that would be more difficult to achieve in a full text-line scenario [19].

To this end, we design a carefully controlled experimental framework in which BiLSTM- and Transformer-based sequence encoders are compared under identical conditions [20, 21]. Both models share the same CNN backbone (truncated ResNet-50), CTC formulation, decoding strategy based on Word Beam Search (WBS), data augmentation schemes, and dataset splits, allowing us to isolate the impact of the sequence modeling component itself [22, 23]. This setup enables a fair and reproducible comparison between recurrent and attention-based encoders.

In addition to architectural comparison, this study examines the role of data distribution in Arabic handwritten recognition. Due to the highly imbalanced frequency of characters and word forms, standard uniform data augmentation may not sufficiently address rare or diacritic-rich patterns [24, 25]. We therefore evaluate adaptive augmentation strategies based on character and word frequency and analyze how these strategies interact with recurrent and attention-based encoders. This analysis provides insight into how frequency-aware augmentation can improve generalization across writers and word forms.

An important aspect of this work is the explicit analysis of computational efficiency alongside recognition accuracy. In addition to Character Accuracy Rate (CAR) and Word Accuracy Rate (WAR), we report detailed training and inference time measurements for all experimental configurations. This analysis allows us to quantify the accuracy-efficiency trade-offs between BiLSTM and Transformer encoders under consistent experimental conditions, providing practical insights into their suitability for large-scale or time-sensitive handwriting recognition scenarios.

The experimental evaluation was conducted on the IFN/ENIT dataset under multiple standard training-testing protocols using CAR and WAR metrics [26]. The consistency of the experimental setup and the use of multiple protocols ensured that the reported findings are robust, reproducible, and directly comparable.

In summary, the contributions of this paper are threefold:

(i) a controlled, hypothesis-driven comparison of BiLSTM- and Transformer-based sequence modeling for Arabic cursive handwritten word recognition, with explicit consideration of ligatures and diacritic-related ambiguities;

(ii) an analysis of adaptive frequency-aware data augmentation strategies and their interaction with recurrent and attention-based encoders;

(iii) a systematic evaluation of computational efficiency, highlighting the trade-offs between recognition accuracy and computational cost.

The remainder of this paper is organized as follows. Section II reviews related work on Arabic Handwritten Word Recognition (AHWR). Section III presents the theoretical foundations of the proposed approach. Section IV describes the dataset and the data augmentation strategies. Section V details the unified methodology and model architectures. Section VI reports the experimental results, while Section VII provides a discussion of the findings, including qualitative error analysis. Finally, Section 8 concludes the paper and outlines directions for future research.

II. Related Work

AHWR has attracted sustained research interest due to the linguistic complexity of the Arabic script and its importance in document analysis applications. Recent advances in deep learning have led to substantial performance improvements, with models evolving from holistic convolutional approaches toward sequence-aware architectures capable of capturing the cursive and context-dependent nature of Arabic handwriting [17, 18]. Early CNN-based models demonstrated that convolutional features are effective for extracting discriminative visual patterns from handwritten word images [19, 21]. However, comparative evaluations on benchmarks such as IFN/ENIT revealed that increasing network depth alone does not consistently yield performance gains under limited data conditions [21, 22]. Moreover, purely CNN-based approaches treat words as static patterns and are unable to explicitly model sequential dependencies arising from ligatures, contextual letter shaping, and kashida elongations [23, 27].

To overcome these limitations, recurrent sequence modeling frameworks became a dominant paradigm in AHWR. End-to-end CNN-RNN-CTC architectures, typically based on BiLSTM encoders, explicitly model handwritten words as character sequences and enable alignment-free prediction through CTC decoding [28, 29]. Extensive studies on IFN/ENIT and AHDB established CNN-BiLSTM-CTC systems combined with WBS decoding as strong baselines, particularly when data augmentation strategies addressing character-level imbalance are employed [24, 30]. While these architectures effectively capture inter-character dependencies without requiring explicit segmentation, their inherently sequential nature limits parallelization and can hinder efficient modeling of long-range dependencies [31]. Attention mechanisms have also been considered within CNN-BiLSTM-CTC frameworks as a complementary modeling strategy, enabling the network to dynamically weight visually or contextually relevant regions of the input image [32, 33]. This selective weighting is particularly beneficial for Arabic handwriting, where visually similar characters differ mainly by diacritics and long ligatures span distant image regions [34]. Consequently, attention-enhanced CNN-RNN-CTC frameworks have been explored to better capture long-range dependencies beyond what can be modeled through recurrent memory alone [35, 36].

More recently, Transformer-based architectures relying on self-attention have gained increasing attention in handwritten text recognition due to their ability to capture global dependencies while enabling parallel computation [37, 38]. In the context of Arabic handwritten text recognition, several studies have reported promising results using Transformer-based approaches, particularly at the text-line or page levels, where self-attention jointly models visual features and implicit linguistic context [39, 40]. Hybrid CNN-Transformer architectures, as well as lightweight and ensemble-based variants, have further demonstrated the potential of attention-based modeling in multi-script and low-resource scenarios, including Arabic and related cursive scripts [41, 42]. However, most of these approaches integrate linguistic modeling or rely on heterogeneous experimental settings, making it difficult to isolate the intrinsic contribution of self-attention relative to recurrent sequence modeling [43]. This limitation is particularly relevant given that CNN-BiLSTM-CTC architectures remain a highly competitive and widely deployed paradigm in practical settings [36, 44, 45], where robustness, data efficiency, and computational constraints are critical.

In contrast, the present study deliberately adopts a CTC-based framework to analyze the sequence modeling mechanism itself, independent of external language priors. By conducting a carefully designed, hypothesis-driven comparison between BiLSTM- and Transformer-based encoders under identical experimental conditions and by focusing on isolated word recognition—where no rich linguistic context is available and performance depends primarily on visual and morphological cues—this work aims to assess whether self-attention provides inherent advantages for modeling the long-range dependencies characteristic of Arabic cursive handwriting, and whether such gains justify the additional computational complexity in practical settings.

III. Theoretical Background and Model Components

The studied architecture follows a standard end-to-end AHWR pipeline composed of a truncated ResNet-50 backbone, a sequence modeling module (either a BiLSTM or a Transformer encoder) [46], and a CTC-based alignment with WBS decoding, as illustrated in Fig. 1. The comparison focuses exclusively on the sequence modeling component, while all other elements of the recognition pipeline remain identical. All components are well established in the handwriting recognition literature. Consequently, this section focuses on the theoretical description of the overall pipeline and on the architectural choices that are directly relevant to the comparison between recurrent and attention-based sequence encoders. Implementation details related to data preprocessing, training configuration, and optimization are intentionally deferred to the model configuration section (Section V).

A. CNN Backbone (Truncated ResNet-50)

For visual feature extraction, a ResNet-50 convolutional backbone [47] is employed, as it has been widely adopted in handwritten text recognition. The convolutional and residual structure of ResNet-50 enables the progressive extraction of local and mid-level visual patterns, such as strokes, contours, and small structural details, through hierarchical feature learning [48]. In our implementation, the network is truncated after the fourth residual stage, before the final downsampling operations, in order to limit excessive spatial resolution reduction. This design choice ensures that the extracted feature maps retain sufficient vertical resolution to encode fine-grained elements such as cursive connections, ligatures, and diacritic marks, which play a critical role in character discrimination in AHWR [49].

B. Sequence Modeling

1) Recurrent Sequence Modeling (BiLSTM)

BiLSTM networks [50] are widely used baselines for handwritten text recognition, as they model local sequential dependencies in both forward and backward directions [17, 35, 36]. In CNN-RNN-CTC pipelines, the convolutional backbone first converts a handwritten word image into a sequence of feature vectors by traversing the feature maps along the horizontal axis. The BiLSTM then processes this sequence bidirectionally, allowing each hidden state to aggregate contextual information from neighboring positions.

Such bidirectional modeling is well aligned with the contextual nature of Arabic handwriting, where character identity emerges from connected strokes, ligatures, and diacritic patterns rather than isolated shapes. Numerous studies have shown that CNN-BiLSTM-CTC architectures provide strong performance for AHWR [17, 35, 36]. However, BiLSTM encoders operate sequentially and

require recurrent computations that scale linearly with sequence length, commonly expressed as O(T) in terms of time steps. In practice, the computational cost also depends on the hidden-state dimensionality and the inherently sequential nature of recurrent processing, which limits parallelization on modern hardware. These characteristics make BiLSTMs an appropriate reference baseline for comparison with attention-based sequence encoders under identical feature representations.

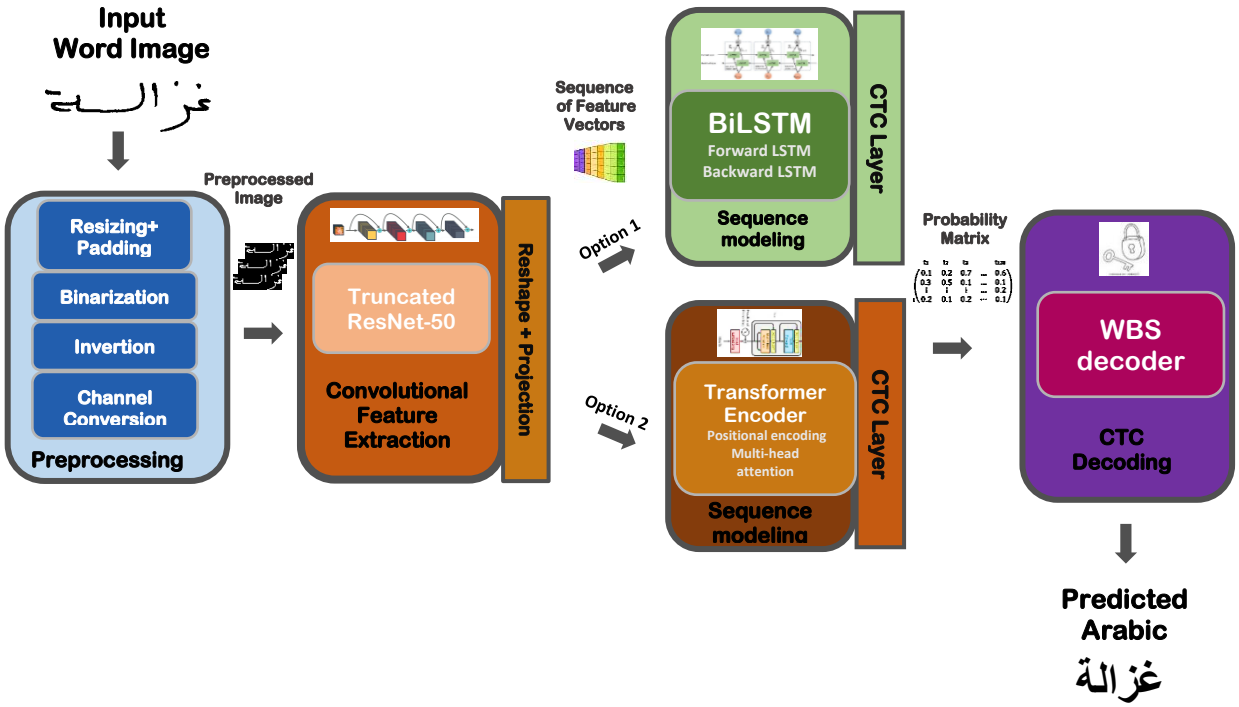


Fig. 1. Handwritten word recognition pipeline with BiLSTM and Transformer encoder.

2) Attention-based modeling (transformer encoder)

Transformer encoders model sequences through self-attention, allowing each position to directly attend to all others regardless of distance [46]. In a CNN-Transformer-CTC pipeline for handwritten text recognition, the convolutional backbone first converts the input text image into a sequence of feature vectors by scanning the feature maps along the horizontal axis, which are then processed by a stack of Transformer encoder layers. Positional encodings are added to preserve the left-to-right ordering of the sequence [51]. It is worth noting that alternative Transformer-based text recognition models relying on autoregressive decoding have also been proposed and have demonstrated strong performance, particularly at the page and text-line level [38, 39].

In AHWR, the use of a Transformer encoder for sequence modeling is particularly relevant due to the cursive nature of the script, where character identity depends on long-range intra-word dependencies arising from extended ligatures, delayed diacritic placement, and context-dependent character shapes [52]. Self-attention provides a direct mechanism for modeling such global contextual relationships, in contrast to recurrent architectures that propagate information sequentially. However, this comes at the cost of quadratic theoretical computational complexity $O(T^2)$ with respect to sequence length, due to pairwise interactions between all positions. These complexity expressions refer to asymptotic sequence-length scaling; in practice, actual runtime performance also depends on feature dimensionality and hardware parallelism. Because self-attention operations can be efficiently parallelized on GPUs, the observed computational overhead is typically moderate despite the higher theoretical complexity.

In this work, the Transformer encoder is used as a direct replacement for the BiLSTM within an otherwise identical CNN-CTC framework, enabling a controlled evaluation of the trade-off between modeling capacity and computational cost for Arabic cursive word recognition.

C. Alignement-Free Training (CTC)

CTC is a standard alignment-free training criterion for handwritten word recognition, where the temporal correspondence between visual features and target characters is unknown and variable [53]. In our framework, CTC enables segmentation-free learning by marginalizing over all valid monotonic alignments between the encoder output sequence and the target transcription [54].

Specifically, the CNN-encoder stack (BiLSTM or Transformer) produces a sequence of feature vectors that are projected onto character probabilities, including a blank symbol. CTC then computes the likelihood of the target word by considering all possible alignments consistent with the character order, without requiring explicit frame-level supervision [55].

Beyond alignment, CTC plays a methodological role in this study by allowing a fair comparison between recurrent

and attention-based sequence encoders. Since decoding does not rely on autoregressive generation or language modeling, any observed performance differences can be attributed more directly to the sequence modeling capacity of the encoder itself.

D. Probabilistic Decoding (WBS)

The WBS algorithm is adopted as the decoding strategy in our CTC-based recognition framework. In this work, WBS is used in a fully unconstrained (open-vocabulary) setting, without relying on an explicit lexicon or external language model [55]. Consequently, recognition is not restricted to a fixed list of candidate words, and all predicted sequences are generated solely based on CTC posterior probabilities.

WBS selects the most probable transcription from the CTC output matrix by maintaining multiple candidate sequences (beams) instead of following a single greedy path. Each beam corresponds to a partial transcription associated with its cumulative probability, and beams are iteratively expanded and pruned to retain only the most promising hypotheses. By exploring multiple candidate paths in parallel, WBS reduces the impact of early decision errors and improves robustness to local ambiguities, such as visually similar character shapes or uncertain CTC predictions [56]. This multi-hypothesis decoding strategy has been shown to outperform best-path and token-passing decoding, particularly in Arabic handwritten text recognition, where cursive variability and character coarticulation amplify local prediction uncertainty [17].

IV. Dataset and Data Augmentation

A. IFN/ENIT Dataset

The IFN/ENIT dataset is a widely used benchmark for AHWR, developed by the Institut Für Nachrichtentechnik (IFN) in Germany and the École Nationale d'Ingénieurs de Tunis (ENIT) [57]. It comprises over 32,000 binary images of Tunisian town and village names written by more than 400 writers, with a vocabulary of approximately 900 distinct words and a character inventory of 123 contextual character classes.

The dataset is partitioned into five subsets (a–e) following standard evaluation protocols. Sets a–d contain overlapping word forms written by different writers, exposing models to diverse handwriting styles, while set e includes only unseen words, providing a more challenging generalization scenario. In our experiments, one subset is used for testing and the remaining subsets for training (e.g., a+b+c for training and d for testing). Fig. 2 shows representative handwritten word samples from the IFN/ENIT dataset, highlighting the diversity of writing styles across different writers.

In this dataset, words are labeled using contextual character classes that encode both character identity and positional form. In Arabic, character shapes vary depending on whether they appear in initial, medial, final, or isolated positions, and handwritten text often includes ligatures that further alter their visual appearance. The adopted coding scheme captures this contextual morphology and connected structure, enabling recognition models to better learn the diverse character shapes and ligature patterns encountered in real handwritten words.

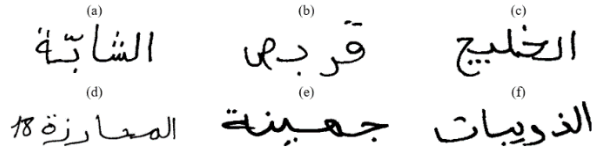


Fig. 2. Sample Arabic handwritten word images from the IFN/ENIT dataset. Subfigures (a)–(f) show randomly selected word samples written by different writers.

Table I illustrates the morphological variations of the Arabic character "م" (*Meem*) according to its position within a word. For each form, the table shows the contextual label used in the dataset, its corresponding visual shape, and an example word demonstrating that specific form in context.

TABLE I. Positional Forms and Labels of the Arabic Character "Meem" with Example Words

Char/ position	Char Label	Visual Form	Example Word	Word Label
م - Isolated	maA	م	أم العظام	aeA|**maA**|aaA|laB|ayM| zaM|aaE|**maA**|
م - Initial	maB	مـ	مارث	**maB**|aaE|raA|thA|
م - Median	maM	ـمـ	ثلاخ	shB|**maM**|aaE|khA|
م - Final	maE	ـم	العلم	aaA|laB|ayM|laM|**maE**|

B. Data Augmentation

AHWR is challenging due to strong cursivity, context-dependent letter shapes, frequent ligatures, and small diacritics that vary widely across writers, resulting in high intra-class variability and sensitivity to geometric distortions. These difficulties are compounded by severe word- and character-level imbalance in datasets such as IFN/ENIT, which limits the model's exposure to rare patterns and adversely affects CTC-based alignment learning. Data augmentation is therefore essential [24, 30, 58]; prior studies on Arabic and related scripts have shown that augmentation improves generalization by increasing visual diversity, mitigating data scarcity, and compensating for skewed class distributions [30]. These effects are particularly important for high-capacity encoders such as ResNet-BiLSTM and ResNet-Transformer, as well as for CTC-based sequence models that require sufficient occurrences of each character to learn stable alignment behaviors.

In this study, augmentation is not only used to enhance accuracy but also to analyze how frequency-aware strategies interact with different sequence encoders under consistent experimental conditions. To this end, we evaluate three complementary approaches: (i) Traditional Uniform Augmentation (TUA), which increases visual variability without modifying class frequencies; (ii) Word-Frequency Augmentation (WFA), which allocates more synthetic samples to underrepresented words; and (iii) Character-Frequency Augmentation (CFA), which targets rare characters and ligatures to promote more stable CTC alignment. Fig. 3 summarizes

the sampling and weighting mechanisms employed by these strategies. In the frequency-aware approaches, normalized weights derived from inverse word frequencies or average inverse character frequencies are used to determine the number of augmented samples allocated to each word, thereby increasing the representation of rare patterns during training. The geometric transformations used in all approaches include elastic deformation, rotation, perspective distortion, Gaussian noise, and blur [59], selected to simulate realistic handwriting variability without compromising legibility (Table II). Fig. 4 illustrates how a single word from the IFN/ENIT dataset is transformed through these five augmentation operations, visually demonstrating the range of distortions introduced during training. Augmentation is applied exclusively to the training sets, with each protocol expanded to a fixed size of 50,000 samples, while test sets remain unchanged to ensure fair and unbiased evaluation (Table III).

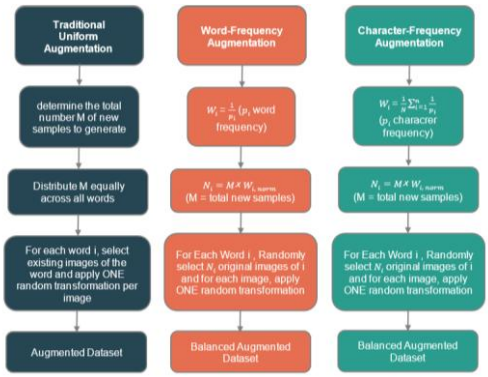


Fig. 3. Schematic diagram illustrating the three data augmentation pipelines used in the study.

TABLE II. CHARACTERISTICS OF TRANSFORMATIONS APPLIED FOR DATA AUGMENTATION

Transformation	Description	Parameters
Elastic Deformation	Simulates natural variations in handwriting and paper distortion	$\sigma = 3$, $\alpha = 19$
Random Rotation	Compensates for orientation variations during scanning	$\theta \in [-3°, +3°]$
Perspective Distortion	Simulates perspective effects and document tilting	margin = 5 pixels
Gaussian Noise	Reproduces scanning artifacts and quality variations	$\sigma_{noise} = 0.04$
Blur	Simulates slight defocusing or smudging, reproducing common scanning flaws	kernel size = 5×5

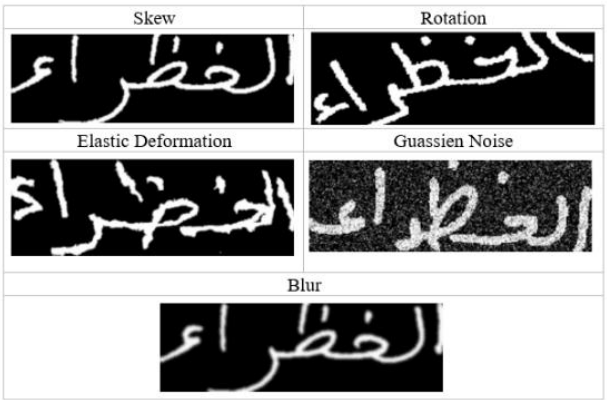


Fig. 4. Visual representation of a word from the dataset after applying the five data augmentation transformations.

TABLE III. DATASET SIZES BEFORE AND AFTER AUGMENTATION FOR EACH EXPERIMENTAL PROTOCOL

Protocol	Role	Before Augmentation	After Augmentation
a+b+c	Training	10,442	50,000
d	Testing	3751	3751
b+c+d	Training	10,621	50,000
a	Testing	3572	3572
a+b+c+d	Training	14,193	50,000
e	Testing	3150	3150

V. PREPROCESSING AND EXPERIMENTAL SETUP

This section presents the preprocessing pipeline, training setup, experimental environment, and evaluation metrics used in our experiments. These elements are identical across all architectures to ensure a fair and reproducible comparison.

A. Preprocessing

To prepare the word images for ResNet-50 processing, each sample is normalized to a fixed size of 96×256 pixels. The height is set to 96 pixels, the width is scaled proportionally, and any remaining space is padded with a white background to preserve the original aspect ratio, a strategy well suited to handwritten words whose widths vary significantly while heights remain relatively consistent [60]. The resized images are then binarized to enhance foreground-background contrast and inverted so that the text appears in black on a white background, matching the expected input polarity of the recognition pipeline. Because ResNet-50 requires three input channels, the processed single-channel images are replicated to form a 96×256×3 tensor [61]. Training is performed both without augmentation and with the three augmentation schemes described previously to assess robustness under consistent experimental conditions.

B. Implementation Details

Both models share the same visual feature extractor, namely a ResNet-50 pretrained on ImageNet [62, 63] and truncated after Stage 4 (conv4_block6_out). Given an input image of size 96×256×3, the stem convolution and subsequent downsampling in Stages 2–4 progressively reduce the feature map to 6×16×1024. A 1×1 convolution (kernel size 1, stride 1, padding "same") projects this tensor to 256 channels. The resulting 6×16×256 feature map is then reshaped into a sequence of 96 time steps with 256-dimensional feature vectors, which serve as the common input to both sequence encoders.

The recurrent baseline employs a BiLSTM encoder composed of two stacked bidirectional LSTM layers, each

with 128 hidden units per direction, yielding a 256-dimensional representation at each time step. A recurrent dropout rate of 0.25 is applied between layers to improve generalization. The Transformer-based model replaces recurrence with a three-layer Transformer encoder augmented with sinusoidal positional encodings. Each layer consists of four-head self-attention followed by a feed-forward network of size 512→256, with residual connections, layer normalization, and a dropout rate of 0.1.

In both architectures, the encoder outputs are projected to 124 character classes (including the CTC blank) through a dense layer and decoded using the same CTC alignment and WBS decoding strategy. The BiLSTM-based architecture contains approximately 9.7 million trainable parameters, whereas the Transformer-based architecture contains approximately 12.8 million parameters. Table IV summarizes the main architectural differences between the two encoders.

TABLE IV. COMPARATIVE OVERVIEW OF THE BILSTM AND TRANSFORMER ENCODERS WITHIN THE SHARED CNN-CTC FRAMEWORK

Component	BiLSTM-based Model	Transformer-based Model
Preprocessing	Image normalization and resizing	Image normalization and resizing
CNN Backbone	ResNet-50 (Truncated at Stage 4)	ResNet-50 (Truncated at Stage 4)
CNN Output	6×16×1024	6×16×1024
Channel Projection	1×1 Conv → 256 channels	1×1 Conv → 256 channels
Sequence Construction	Reshape 6×16 → 96 time steps	Reshape 6×16 → 96 time steps
Sequence Input to Encoder	96×256	96×256
Encoder Type	2-layer BiLSTM	3-layer Transformer Encoder
Hidden / Model Dimension	128 units per direction (256 total)	Model dimension = 256
Context Modeling	Sequential forward/backward recurrence	Global self-attention
Multi-Head Attention	Not applicable	4 attention heads
Positional Encoding	Implicit (via recurrence)	Sinusoidal positional encoding
Feed-Forward Network	Not applicable	512 → 256
Dropout	0.25 (recurrent)	0.1 (after attention & FFN)
Output Projection	Linear → 256	Identity (already 256)
CTC Classifier	Dense (124 classes)	Dense (124 classes)
Alignment & Decoding	CTC + WBS	CTC + WBS
Total Trainable Parameters	~9.7M	~12.8M

C. Training Setup and Environement

The models were trained using the Adam optimizer with an initial learning rate of 0.001. To ensure stable convergence across all experimental configurations, a learning rate decay strategy was applied during training, allowing larger parameter updates in early epochs and finer adjustments in later stages. A batch size of 32 was used for all experiments, and training was performed for up to 50 epochs with early stopping based on the validation CTC loss to prevent overfitting. For each configuration, the training data were split into 80% for parameter optimization and 20% for validation.

All experiments were implemented in PyTorch (version 1.12) using Python 3.9 and were executed on a workstation equipped with an NVIDIA RTX 3090 GPU (24 GB VRAM), 32 GB of system memory, and running Ubuntu 20.04.

D. Evaluation Metrics

To evaluate recognition performance and compare the two sequence encoders within a unified AHWR framework, we report two standard metrics widely used in handwritten text recognition, namely CAR and WAR.

These metrics provide complementary character-level and word-level evaluations and are particularly relevant for Arabic handwriting, where character-level errors directly affect word correctness due to the cursive nature of the script.

CAR measures the proportion of correctly recognized characters [64] and is defined as:

$$CAR(\%) = \left[1 - \frac{S+I+D}{N}\right] \times 100\ \% \quad (1)$$

where N denotes the total number of characters in the ground truth, and S, D, and I represent the numbers of substitutions, deletions, and insertions, respectively. All error types are equally weighted, following the standard CAR formulation.

WAR provides a stricter evaluation by measuring the proportion of words that are entirely recognized without any character-level errors:

$$WAR(\%) = \frac{W_c}{W_t} \times 100\ \% \quad (2)$$

where W_t is the total number of words in the ground truth, and W_c the number of words exactly matched by the model (i.e., words with no character-level errors).

VI. RESULTS AND ANALYSIS

The quantitative results obtained from the experimental protocol are reported in Tables V and VI for the ResNet-50-BiLSTM-CTC and ResNet-50-Transformer-CTC architectures, respectively. Table VII provides complementary information regarding computational efficiency, including total training time, average epoch duration, and inference latency per image. To assess the statistical robustness of the primary encoder comparison, additional multi-run experiments were conducted for the CFA configuration under the abcd-e protocol, and mean ± standard deviation values along with paired statistical testing are reported separately. The results are analyzed along five main axes: (i) comparison of sequential encoders under consistent experimental conditions, (ii) impact of data augmentation strategies on each encoder, (iii) generalization behavior across the three train-test protocols, (iv) computational efficiency and cost-accuracy trade-offs, and (v) robustness analysis across random

seeds. This structured analysis enables a comprehensive assessment of recognition performance, statistical stability, and practical deployment considerations.

A. Encoder-To-Encoder Performance Comparison

Across all experimental configurations, the Transformer encoder consistently achieves higher CAR and WAR scores than the BiLSTM model. This pattern is observed uniformly across the three train-test protocols and under all augmentation settings. On the original (non-augmented) dataset, improvements appear immediately: in the abc-d protocol, CAR increases from 84.30% with BiLSTM to 86.91% with the Transformer, and WAR increases from 81.40% to 83.77%. Similar differences are observed in the bcd-a protocol, where CAR rises from 83.52% to 86.14%, and WAR increases from 80.31% to 83.02%. These numerical trends confirm a consistent margin of approximately +2–3% CAR and +1.6–2.2% WAR in favor of the Transformer across the baseline conditions.

Performance differences become more substantial when applying CFA. Under CFA, the BiLSTM WAR increases from 81.40% to 95.75%, whereas the Transformer improves from 83.77% to 98.41% in the abc-d protocol. Similar results are found in the bcd-a and abcd-e configurations, where the Transformer reaches a maximum WAR of 98.91%, compared to 95.75% for BiLSTM. These values indicate that, across all protocols and augmentation scenarios, the Transformer consistently achieves higher performance ceilings than the BiLSTM, and the performance gap does not diminish as augmentation strength increases.

B. Impact of Data Augmentation Strategies

The evaluation of the three augmentation strategies (TUA, WFA, and CFA) shows a consistent increase in performance for both encoders across all train-test configurations. TUA leads to noticeable improvements compared to the original dataset. For example, in the abc-d protocol, BiLSTM WAR increases from 81.40% to 89.10%, while the Transformer WAR increases from 83.77% to 90.91%. Similar trends are observed in the bcd-a and abcd-e protocols.

WFA results in further performance gains for both models. In the abc-d configuration, BiLSTM WAR increases to 93.83%, and the Transformer WAR reaches 95.06%. Comparable improvements are observed across the other protocols, indicating that WFA consistently outperforms TUA for both encoders.

The highest recognition rates are obtained with CFA. Under CFA, BiLSTM WAR reaches 95.75% in the abc-d protocol, while the Transformer achieves 98.41%, corresponding to an absolute improvement of +14.64 percentage points over the original dataset. Across all configurations, CFA consistently yields the highest CAR and WAR values for both models.

C. Generalization Behaviour across the Three Train–Test Protocols

The three evaluation protocols—abc-d, bcd-a, and abcd-e—are designed to assess recognition performance under different train-test splits involving disjoint subsets of writers and word forms. The abc-d and bcd-a configurations evaluate performance when training and testing are performed on different subsets, while the abcd-e configuration evaluates generalization to a subset (e) containing word forms that do not appear in the training data.

Across all augmentation settings, both models exhibit lower recognition accuracy on abcd-e compared to abc-d and bcd-a, reflecting the increased difficulty of generalizing to unseen word forms. However, for all configurations and augmentation schemes, the Transformer consistently achieves higher WAR values than the BiLSTM. Without augmentation, BiLSTM WAR decreases from 81.40% in abc-d to 78.92% in abcd-e, while the Transformer WAR decreases from 83.77% to 81.55%. A similar pattern is observed with augmented datasets. Under CFA, BiLSTM WAR decreases from 95.75% (abc-d) to 94.01% (bcd-a) and further to 92.77% (abcd-e), whereas the Transformer WAR decreases from 98.41% to 96.11% and 94.89%, respectively.

Overall, the reduction in performance across protocols is consistently smaller for the Transformer encoder than for the BiLSTM encoder. This trend is observed under all augmentation strategies and indicates that the Transformer maintains more stable recognition accuracy when evaluated on disjoint subsets, including those containing previously unseen word forms.

D. Computational Efficiency and Cost-Accuracy Trade-offs

A detailed comparison of computational efficiency reveals clear differences between the two sequential encoders in terms of training time, epoch duration, and inference latency. Across all protocols and augmentation settings, the Transformer-based architecture requires substantially more computational resources than the BiLSTM model. For example, in the abc-d protocol using the original dataset, the average epoch duration rises from ~2.5 min for the BiLSTM to ~3.6 min for the Transformer, and total training time grows from approximately 1 h to 1.8 h. This pattern becomes more pronounced with larger augmented datasets: under CFA with 50,000 training samples, epoch time increases from ~5.2 min (BiLSTM) to ~9.2 min (Transformer), and total training duration nearly doubles. Inference speed follows the same trend: The BiLSTM achieves an average processing time of ~2.5 ms per image, whereas the Transformer requires ~4.5 ms per image, reflecting a consistent overhead in real-time scenarios. Under consistent experimental conditions, the Transformer incurs an estimated 1.8× to 2× computational cost relative to the BiLSTM, establishing a quantifiable trade-off between accuracy and computational efficiency across all evaluated configurations.

TABLE V. RESNET50-BILSTM-CTC—PERFORMANCE ACROSS ALL DATA AUGMENTATION SCENARIOS

Train-Test Configuration	Original Dataset		TUA		WFA		CFA	
	CAR (%)	WAR (%)	CAR (%)	WAR (%)	CAR (%)	WAR (%)	CAR (%)	WAR (%)
abc-d	84.30	81.40	90.75	89.10	95.61	93.83	96.83	95.75
bcd-a	83.52	80.31	89.46	87.67	94.29	92.15	95.95	94.01
abcd-e	81.79	78.92	88.21	85.24	93.07	92.86	94.62	92.77

TABLE VI. RESNET50-TRANSFORMER-CTC—PERFORMANCE ACROSS ALL DATA AUGMENTATION SCENARIOS

Train-Test Configuration	Original Dataset		TUA		WFA		CFA	
	CAR (%)	WAR (%)	CAR (%)	WAR (%)	CAR (%)	WAR (%)	CAR (%)	WAR (%)
abc-d	86.91	83.77	92.84	90.91	96.38	95.06	98.91	98.41
bcd-a	86.14	83.02	91.95	89.70	95.63	94.33	97.72	96.11
abcd-e	84.37	81.55	90.52	88.41	94.28	93.16	96.03	94.89

TABLE VII. TRAINING AND INFERENCE TIME COMPARISON ACROSS PROTOCOLS AND AUGMENTATION STRATEGIES

Architecture	Protocol	Augmentation Strategy	Training Samples	Training Time (h)	Avg. Epoch Time (min)	Inference Time / Image (ms)
ResNet50 + BiLSTM + CTC	**abc-d**	Original	10,442	~1.0	~2.5	~2.5
		TUA	50,000	~2.5	~5.0	~2.5
		WFA	50,000	~2.6	~5.2	~2.6
		CFA	50,000	~2.6	~5.2	~2.6
	bcd-a	Original	10,621	~1.0	~2.6	~2.5
		TUA	50,000	~2.5	~5.0	~2.5
		WFA	50,000	~2.6	~5.2	~2.6
		CFA	50,000	~2.6	~5.2	~2.6
	abcd-e	Original	14,193	~1.2	~3.0	~2.5
		TUA	50,000	~2.5	~5.0	~2.5
		WFA	50,000	~2.6	~5.2	~2.6
		CFA	50,000	~2.6	~5.2	~2.6
ResNet50 + Transformer + CTC	**abc-d**	Original	10,442	~1.8	~3.6	~4.5
		TUA	50,000	~4.5	~8.5	~4.5
		WFA	50,000	~4.7	~9.0	~4.6
		CFA	50,000	~4.8	~9.2	~4.6
	bcd-a	Original	10,621	~1.9	~3.8	~4.5
		TUA	50,000	~4.5	~8.5	~4.5
		WFA	50,000	~4.7	~9.0	~4.6
		CFA	50,000	~4.8	~9.2	~4.6
	abcd-e	Original	14,193	~2.0	~4.0	~4.5
		TUA	50,000	~4.5	~8.5	~4.5
		WFA	50,000	~4.7	~9.0	~4.6
		CFA	50,000	~4.8	~9.2	~4.6

E. Statistical Robustness Across Random Seeds

To evaluate the stability of the reported performance differences and address potential variance introduced by random initialization, three independent training runs were conducted for both encoders under the CFA configuration on the abcd-e protocol. This setting was selected as it represents the most challenging and representative evaluation scenario. Mean and standard deviation values for CAR and WAR are reported in Table VIII, along with paired statistical testing.

The results show low variance across runs for both architectures, indicating stable convergence behavior. The Transformer consistently outperforms the BiLSTM across all repetitions, maintaining a clear performance margin in both CAR and WAR. Paired t-tests [65] confirm that the observed differences are statistically significant ($p < 0.05$), demonstrating that the reported performance gains are not attributable to random seed effects. These findings reinforce the reliability of the encoder comparison and provide additional methodological rigor to the experimental evaluation.

TABLE VIII. STATISTICAL ROBUSTNESS ANALYSIS ACROSS RANDOM SEEDS (CFA, ABCD-E)

Model	CAR (%) (mean ± std)	WAR (%) (mean ± std)
ResNet50 + BiLSTM + CTC	94.58 ± 0.21	92.65 ± 0.29
ResNet50 + Transformer + CTC	96.03 ± 0.17	94.92 ± 0.23

VII. DISCUSSION

The objective of this study was to conduct a controlled and reproducible comparison between BiLSTM and Transformer encoders within a unified Arabic handwritten word recognition framework. Unlike many previous

works that modify multiple architectural components simultaneously, all elements of the recognition pipeline were kept identical except for the sequence encoder. Both models share the same truncated ResNet-50 backbone, preprocessing pipeline, CTC formulation, Word Beam Search decoding strategy, data augmentation schemes, and train-test splits, allowing performance differences to be attributed primarily to the sequence modeling mechanism [43, 55, 66].

Under consistent experimental conditions, the Transformer outperforms the BiLSTM in both CAR and WAR across all protocols and augmentation settings. Although the absolute performance gap varies depending on the training configuration, the relative ranking remains stable. Additional multi-run experiments conducted under the CFA configuration on the abcd-e protocol demonstrate low variance across random seeds and confirm that the observed performance differences are statistically significant. These results indicate that self-attention-based sequence modeling provides a systematic and statistically robust advantage over recurrent modeling in Arabic handwritten word recognition within an open-vocabulary, CTC-based setting [67, 68].

The analysis of data augmentation strategies reveals a clear hierarchy in their effectiveness. TUA improves recognition accuracy compared to the original dataset but is outperformed by frequency-aware strategies. WFA yields additional gains, while CFA produces the highest CAR and WAR values for both encoders. Across all configurations, the Transformer benefits more strongly from frequency-aware augmentation, with the performance gap between the two encoders widening under CFA.

Generalization across the three evaluation protocols further highlights differences between the encoders. The abcd-e configuration, which evaluates recognition on unseen word forms, results in a performance decrease for both models. However, across all augmentation settings, the Transformer maintains higher WAR values and exhibits a smaller performance drop than the BiLSTM, indicating more stable behavior under disjoint and unseen-word evaluation conditions.

To better understand the origin of performance differences, a qualitative error analysis was conducted on all misrecognized samples from subset e under the abcd-e configuration with CFA. Errors were manually annotated according to predefined criteria and grouped into five categories: (i) diacritic-related confusions, (ii) visually similar characters, (iii) ligatures and cursive connections, (iv) kashida elongations, and (v) CTC alignment errors. The annotation followed clear guidelines to ensure consistency across samples. When multiple error types co-occurred within the same word, each sample was assigned to a single category corresponding to the most structurally dominant source of misrecognition to avoid double counting. This prioritization rule was applied consistently across all samples to reduce subjective variability. Table IX reports both raw error counts and percentage distributions for each category. The analysis indicates that the Transformer produces fewer diacritic-related and visually ambiguous character errors than the BiLSTM, whereas both models remain challenged by extreme ligatures and highly cursive segments. A relatively higher proportion of CTC alignment errors is observed for the Transformer. Representative examples illustrating these trends are provided in Table X.

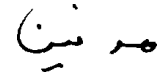
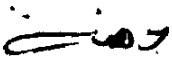
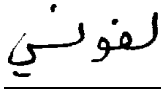
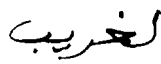
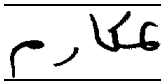
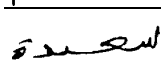
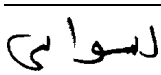
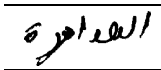
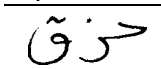
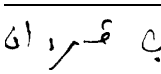
TABLE IX. DISTRIBUTION OF ERROR TYPES ON IFN/ENIT SUBSET E (ABCD-E, CFA)

Error Type	Description	BiLSTM (count /%)	Transformer (count /%)	Observation
Diacritic-related confusions	Incorrect number or placement of dots for characters sharing the same base shape (e.g., ب/ت/ث, ح/خ/ج).	439 (36%)	198 (22%)	The Transformer shows fewer errors, suggesting improved handling of spatially distant or weakly written diacritics.
Visually similar characters	Confusions between characters with similar skeletal shapes due to faint, merged, or displaced diacritics.	293 (24%)	135 (15%)	The Transformer reduces confusion between visually similar character shapes.
Ligatures and cursive connections	Errors caused by ambiguous character boundaries in highly cursive segments.	244 (20%)	234 (26%)	Both models struggle with extreme ligatures, with slightly higher incidence in the Transformer.
Kashida elongations	Elongated strokes misinterpreted as repeated or inserted characters.	98 (8%)	108 (12%)	Both models are affected by elongated strokes, with a slightly higher proportion of errors in the Transformer.
CTC alignment errors	Insertions, deletions, or shifts in dense or highly cursive word segments.	146 (12%)	226 (25%)	CTC alignment errors remain a major source of misrecognition, with a higher proportion observed in the Transformer.
Total Errors		1220 (100%)	901 (100%)	-

These observations suggest that self-attention enhances global interactions among distant visual cues by allowing each position to integrate information from the entire word [46, 52]. This mechanism is particularly effective for associating weak or spatially displaced discriminative features in Arabic handwriting [67], such as diacritics and

subtle shape variations, thereby reducing confusion between visually similar characters. However, this global aggregation may reduce sensitivity to fine-grained local structures and short-range dependencies [69, 70], which are essential for implicit character boundary modeling in highly cursive scripts. As a result, dense connections, elongated kashida strokes, and extreme ligatures are more likely to induce CTC alignment errors, character insertions or deletions, and misinterpretations [44, 71, 72]. Nevertheless, despite these limitations, self-attention-based sequence modeling outperforms recurrent BiLSTM-based approaches, particularly when combined with frequency-aware character-level augmentation strategies [15, 34, 73], indicating that the benefits of global contextual integration outweigh its limitations in this setting.

TABLE X. REPRESENTATIVE EXAMPLES OF TYPES AND COMPARATIVE PREDICTIONS OF BILSTM AND TRANSFORMER ENCODERS

Image Word	Ground Truth	BiLSTM Prediction	Transformer Prediction	Error Category	Comparative Observation
	مدنين	مدس	مدنين	Diacritic-related confusion	The BiLSTM fails to link distant diacritics to character shapes. The Transformer correctly models these long-range visual dependencies
	جهينة	حت	جمنت	Ligatures and cursive connections	Strong cursive connections blur boundaries, causing BiLSTM errors, while the Transformer leverages diacritics
	الفوني	الفولي	الفوني	Visually similar characters confusion	The BiLSTM confuses visually similar characters with subtle shape differences. The Transformer correctly preserves character identity
	الغريب	الخريب	الغريب	Visually similar characters confusion	The BiLSTM confuses structurally similar cursive characters, while the Transformer correctly distinguishes subtle visual differences
	دغومس	دغوس	دغومس	Ligatures and cursive connections	BiLSTM omits a distorted cursive character, while the Transformer maintains a stable sequence representation
	المكارم	اعكارم	الكارم	Ligatures and cursive connections	Both models struggle with strongly fused ligatures that obscure character boundaries, leading to segmentation errors
	السعيدة	السعيدة	السعييدة	Kashida elongation	An elongated stroke causes the Transformer to insert an extra character, whereas the BiLSTM remains stable
	السواسي	السواي	السواسي	Ligatures and cursive connections	Strong cursive connection obscures character boundaries, causing omission in the BiLSTM, while the Transformer remains accurate
	الهدارة	الهدسسك	الهدكسة	CTC alignement	Both models fail due to alignment disruption caused by a severely degraded character
	حزق	حدق	حزق	Diacritic-related confusion	BiLSTM ignores a weak diacritic cue; the Transformer correctly distinguishes the character
	بن قرءان	ن قرر ان	بن قزان	CTC alignment	A weak hamza disrupts alignment, causing severe insertions and deletions in the BiLSTM, while the Transformer maintains better sequence stability
	جميلة	جميلة	جعبيلت	Ligatures and cursive connections	A strong ligature obscures character boundaries, causing segmentation errors in the Transformer, while the BiLSTM remains accurate.
	الجو	اكو	الجو	Diacritic-related confusion	A spatially distant diacritic, reinforced by an initial ligature, causes character confusion in the BiLSTM, while the Transformer correctly associates the distant visual cue

When evaluating prior work on the IFN/ENIT benchmark, two distinct experimental settings can be identified: closed-vocabulary word classification [22, 74–76] and open-vocabulary sequence recognition [17, 29, 35, 36, 77] (see Table XI). In the closed-vocabulary setting, handwritten words are treated as predefined classes, which can lead to very high recognition rates. For example, one study reports a perfect word accuracy rate of 100% on the abc-d protocol [76]. However, such results are obtained under constrained conditions where recognition is limited to a fixed set of candidate words. In contrast, CTC-based approaches operate in an open-vocabulary setting, allowing the model to predict unrestricted character sequences without relying on a predefined lexicon [17, 29, 35, 36, 77]. Within this more challenging and generalizable framework, the proposed architectures remain highly competitive. In particular, the Transformer-based configuration achieves performance comparable to or exceeding the best reported open-vocabulary results across the abc-d, bcd-a, and abcd-e protocols, while maintaining consistent accuracy across all evaluation splits.

TABLE XI. COMPARISON WITH OTHER STATE-OF-THE-ART SYSTEMS EVALUATED ON THE IFN/ENIT DATABASE

Author	Architecture	Vocabulary setting	Augmentation	WAR (abc-d)	WAR (bcd-a)	WAR (abcd-e)
[74]	DBN	Closed vocabulary	-	83.7	-	-
[75]	CNN + DBN + SVM	Closed vocabulary	-	91.55	-	-
[76]	ResNet50	Closed vocabulary	-	100	-	-
[22]	ResNet18	Closed vocabulary	-	96.11	-	-
[29]	RNN-GRU + CTC + Dropout	Open vocabulary	-	86.49	-	-
[77]	MLDTSTM + CTC	Open vocabulary	Uniform	92.46	-	-
[17]	BiLSTM + CTC	Open vocabulary	CFA	**98.99**	95.05	93.57
[36]	CNN + BiLSTM + CTC	Open vocabulary	-	-	-	92.21
[35]	CNN + BiLSTM + CTC	Open vocabulary	-	98.09	-	-
Present Work	ResNet50 + BiLSTM + CTC	Open vocabulary	CFA	95.75	94.01	92.77
	ResNet50 + Transformer + CTC	Open vocabulary	CFA	98.41	**96.11**	**94.89**

Finally, the computational analysis highlights a clear accuracy–efficiency trade-off. The Transformer incurs approximately 1.8× to 2× higher computational cost in training and inference compared to the BiLSTM under identical conditions. This trade-off should be considered when selecting a sequence encoder for deployment scenarios with constrained resources.

Despite the controlled and reproducible nature of the proposed evaluation, the scope of this study remains intentionally constrained to ensure a fair and interpretable comparison between sequence encoders. The analysis is limited to isolated word recognition on IFN/ENIT to eliminate confounding factors related to segmentation, layout variability, and inter-word spacing, so that performance differences can be attributed primarily to the encoder design. A CTC-based formulation and frequency-aware augmentation strategies are consistently adopted to maintain identical alignment and training conditions. While the main results are reported following standard IFN/ENIT evaluation protocols, additional repeated runs with different random seeds were conducted for the CFA configuration under the abcd-e protocol to assess robustness and variance across training initializations. These experiments confirm the stability of the observed performance differences. Extending multi-run statistical validation to all configurations, as well as exploring alternative alignment mechanisms or longer text units, represents a natural continuation of this work.

Overall, this study demonstrates that, under consistent experimental conditions, Transformer encoders provide consistent performance advantages over BiLSTMs for Arabic handwritten word recognition, particularly when combined with frequency-aware data augmentation.

The qualitative error analysis shows that these gains are closely linked to improved handling of diacritic-related and visually ambiguous characters, which are central challenges in Arabic handwriting. At the same time, the increased computational cost associated with self-attention must be carefully considered. Together, these results provide a clearer understanding of how different sequence modeling paradigms behave in the presence of the intrinsic variability of Arabic cursive handwriting and establish a strong open-vocabulary baseline for future research.

VIII. CONCLUSION

This study presents a carefully designed comparison between BiLSTM and Transformer encoders for Arabic handwritten word recognition. By keeping all components of the recognition pipeline consistent except for the sequence encoder, the analysis isolates the effect of sequence modeling and demonstrates a consistent performance advantage of attention-based modeling across evaluation protocols and augmentation strategies.

Beyond overall accuracy gains, the Transformer exhibits greater robustness to diacritic-related and visually ambiguous characters, although both models remain challenged when handling highly cursive structures. This improvement comes at the cost of increased computational complexity, highlighting a measurable accuracy-efficiency trade-off within an open-vocabulary, CTC-based setting.

These findings establish a reproducible benchmark for encoder-level comparison in Arabic handwriting recognition and provide a foundation for future extensions to more complex scenarios, including line-level and multi-word recognition, where longer sequences and structural variability may further influence the relative behavior of recurrent and attention-based encoders.

DATA AVAILABILITY STATEMENT

The datasets used for training and evaluation in this study are publicly available. Specifically, the IFN/ENIT Arabic handwriting database can be accessed at http://www.ifnenit.com/ or upon request from the original dataset authors. Additional processed data or code used during the experiments are available from the corresponding author upon reasonable request.

CONFLICT OF INTEREST

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

IB. and AA. conceived and designed the proposed model, implemented the architecture, and set up the experimental framework. They conducted all experiments,

processed and analyzed the results, and prepared the initial draft of the manuscript. GK. and MM. provided scientific supervision throughout the study, guided the technical methodology, and validated the proposed model and experimental results. They also critically reviewed, edited, and improved the manuscript to ensure its scientific rigor and clarity. All authors discussed the results, contributed to refining the study, and approved the final version of the manuscript.

ACKNOWLEDGMENT

The authors gratefully acknowledge the researchers at ENSA and EuroMed University for their valuable support, insightful discussions, and assistance throughout this study. We also acknowledge the institutional and technical support provided by our laboratory and university, which made this work possible. We are grateful to the reviewers and editors for their constructive comments that helped improve the quality of this manuscript.

REFERENCE

[1] A. Beg, F. Ahmed, and P. Campbell, "A clustering technique for digital communications channel equalization using radial basis function networks," in *Proc. 2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks*, IEEE, 2010, pp. 101–105. https://doi.org/10.1109/CICSyN.2010.36

[2] 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, pp. 1–37, 2025.

[3] R. Al-Hajj, C. Mokbel, and L. Likforman-Sulem, "Combination of HMM-based classifiers for the recognition of Arabic handwritten words," in *Proc. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, IEEE, 2007, pp. 959–963. https://doi.org/10.1109/ICDAR.2007.4377057

[4] A. Kundu, T. Hines, B. Huyck, J. Phillips, and L. V. Guilder, "Arabic handwriting recognition using variable duration HMM," in *Proc. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007),* vol. 2, IEEE, 2007, pp. 644–648. https://doi.org/10.1109/icdar.2007.4376994

[5] S. A. Azeem and H. Ahmed, "Effective technique for the recognition of offline Arabic handwritten words using hidden Markov models," *IJDAR*, vol. 16, pp. 399–412, 2013. https://doi.org/10.1007/s10032-013-0201-8

[6] M. Khalifa and Y. BingRu, "A novel word based Arabic handwritten recognition system using SVM classifier," in *Communications in Computer and Information Science*, Springer Berlin Heidelberg, 2011, pp. 163–171. doi: 10.1007/978-3-642-20367-1_26

[7] S. A. Mahmoud and S. O. Olatunji, "Handwritten Arabic numerals recognition using multi-span features & support vector machines," in *Proc. 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*, IEEE, 2010, pp. 618–621. https://doi.org/10.1109/isspa.2010.5605423

[8] M. Bisht and R. Gupta, "Offline handwritten Devanagari word recognition using CNN-RNN-CTC," *SN Comput. Sci.*, vol. 4, 88, 2022. https://doi.org/10.1007/s42979-022-01461-x

[9] R. Ahmed *et al.*, "Offline Arabic handwriting recognition using deep machine learning: A review of recent advances," in *Advances in Brain Inspired Cognitive Systems Lecture Notes in Computer Science*, J. Ren *et al.*, Eds., Springer International Publishing, 2020, pp. 457–468. doi: 10.1007/978-3-030-39431-8_44

[10] S. K. Jemni, Y. Kessentini, S. Kanoun, and J.-M. Ogier, "Offline Arabic handwriting recognition using BLSTMs combination," in *Proc. 2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, IEEE, 2018, pp. 31–36. https://doi.org/10.1109/DAS.2018.54

[11] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin, "GTC: Guided training of CTC towards efficient and accurate scene text recognition," *AAAI*, vol. 34, pp. 11005–11012, 2020. https://doi.org/10.1609/aaai.v34i07.6735

[12] L. Kang, P. Riba, M. Rusiñol, A. Fornés, and M. Villegas, "Pay attention to what you read: non-recurrent handwritten text-line recognition," *Pattern Recognition*, 129, 108766, 2022.

[13] A. Mostafa *et al.*, "An end-to-end OCR framework for robust Arabic-handwriting recognition using a novel transformers-based model and an innovative 270 million-words multi-font corpus of classical Arabic with diacritics," arXiv preprint, arXiv:2208.11484, 2022. https://doi.org/10.48550/arXiv.2208.11484

[14] S. Khamekhem Jemni, S. Ammar, M. A. Souibgui, Y. Kessentini, and A. Cheddad, "ST-KeyS: Self-supervised transformer for keyword spotting in historical handwritten documents," *Pattern Recognition*, vol. 170, 112036, 2026.

[15] B. Rabhi *et al.*, "A novel multi-head attention and long short-term network for enhanced inpainting of occluded handwriting," *Cogn. Comput.*, vol. 17, 6, 2025. https://doi.org/10.1007/s12559-024-10382-1

[16] M. Li *et al.*, "TrOCR: Transformer-based optical character recognition with pre-trained models," in *Proc. the AAAI Conference on Artificial Intelligence,* vol. 37, no. 11, 2023, pp. 13094–13102.

[17] M. Eltay, A. Zidouri, and I. Ahmad, "Exploring deep learning approaches to recognize handwritten Arabic texts," *IEEE Access*, vol. 8, pp. 89882–89898, 2020. https://doi.org/10.1109/ACCESS.2020.2994248

[18] Q. Al-nuzaili, D. Mohamad, N. A. Ismail, and M. S. Khalil, "Feature extraction in holistic approach for Arabic handwriting recognition system: A preliminary study," in *Proc. 2012 IEEE 8th International Colloquium on Signal Processing and its Applications*, IEEE, 2012, pp. 335–340. https://doi.org/10.1109/CSPA.2012.6194745

[19] M. E. Mustafa and M. Khalafallah, "A deep learning approach for handwritten Arabic names recognition," *IJACSA*, vol. 11, 2020. https://doi.org/10.14569/IJACSA.2020.0110183

[20] A. Boukharouba and A. Bennia, "Recognition of handwritten Arabic literal amounts using a hybrid approach," *Cogn. Comput.*, vol. 3, pp. 382–393, 2011. https://doi.org/10.1007/s12559-010-9088-6

[21] T. M. Ghanim, M. I. Khalil, and H. M. Abbas, "Comparative study on deep convolution neural networks DCNN-based offline Arabic handwriting recognition," *IEEE Access*, vol. 8, pp. 95465–95482, 2020. https://doi.org/10.1109/ACCESS.2020.2994290

[22] M. Awni, M. I. Khalil, and H. M. Abbas, "Offline Arabic handwritten word recognition: A transfer learning approach," *Journal of King Saud University—Computer and Information Sciences*, vol. 34, pp. 9654–9661, 2022. https://doi.org/10.1016/j.jksuci.2021.11.018

[23] A. Lamsaf, M. Ait Kerroum, S. Boulaknadel, and Y. Fakhri, "Recognition of Arabic handwritten words using convolutional neural network," *IJEECS*, vol. 26, 1148, 2022. https://doi.org/10.11591/ijeecs.v26.i2.pp1148-1155

[24] M. Eltay, A. Zidouri, I. Ahmad, and Y. Elarian, "Improving handwritten Arabic text recognition using an adaptive data-augmentation algorithm," in *Proc. Document Analysis and Recognition – ICDAR 2021 Workshops Lecture Notes in Computer Science*, E. H. Barney Smith and U. Pal, Eds., Springer International Publishing, 2021, pp. 322–335. https://doi.org/10.1007/978-3-030-86198-8_23

[25] A. F. De Sousa Neto, B. L. D. Bezerra, G. C. D. De Moura, and A. H. Toselli, "Data augmentation for offline handwritten text recognition: A systematic literature review," *SN Comput. Sci.*, vol. 5, 258, 2024. https://doi.org/10.1007/s42979-023-02583-6

[26] H. El Abed and V. Margner, "The IFN/ENIT-database—A tool to develop Arabic handwriting recognition systems," in *Proc. 2007 9th International Symposium on Signal Processing and Its Applications*, IEEE, pp. 1–4, 2007. https://doi.org/10.1109/ISSPA.2007.4555529

[27] A. M. Azmi and A. Alsaiari, "A calligraphic based scheme to justify Arabic text improving readability and comprehension," *Computers in Human Behavior*, vol. 39, pp. 177–186, 2014. https://doi.org/10.1016/j.chb.2014.07.003

[28] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene

text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, pp. 2298–2304, 2017. https://doi.org/10.1109/TPAMI.2016.2646371

[29] L. Chen, R. Yan, L. Peng, A. Furuhata, and X. Ding, "Multi-layer recurrent neural network based offline Arabic handwriting recognition," in *Proc. 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, IEEE, 2017, pp. 6–10. https://doi.org/10.1109/ASAR.2017.8067749

[30] M. A. Chadli, R. B. Bouiadjra, A. Fekir, J. Martínez-Gómez, and J. A. Gámez, "Data augmentation for offline Arabic handwritten text recognition using moving least squares," *RIA*, vol. 38, pp. 1–9, 2024. https://doi.org/10.18280/ria.380101

[31] L. Johnston, V. Patel, Y. Cui, and P. Balaprakash, "Revisiting the problem of learning long-term dependencies in recurrent neural networks," *Neural Networks*, vol. 183, 106887, 2025. https://doi.org/10.1016/j.neunet.2024.106887

[32] 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, pp. 520–540, 2021. https://doi.org/10.3390/forecast3030033

[33] T. B. A. Gader and A. K. Echi, "Attention-based deep learning model for Arabic handwritten text recognition," *MG&V*, vol. 31, pp. 49–73, 2022. https://doi.org/10.22630/MGV.2022.31.1.3

[34] B. Imane, A. Alae, K. Ghizlane, and M. Mrabti, "Enhancing Arabic handwritten word recognition: A CNN-BiLSTM-CTC architecture with attention mechanism and adaptive augmentation," *Discov. Appl. Sci.*, vol. 7, 460, 2025. https://doi.org/10.1007/s42452-025-06952-z

[35] L. Mosbah, I. Moalla, T. M. Hamdani, B. Neji, T. Beyrouthy, and A. M. Alimi, "ADOCRNet: A deep learning OCR for Arabic documents recognition," *IEEE Access*, vol. 12, pp. 55620–55631, 2024. https://doi.org/10.1109/ACCESS.2024.3379530

[36] R. Maalej and M. Kherallah, "Convolutional neural network and BLSTM for offline Arabic handwriting recognition," in *Proc. 2018 International Arab Conference on Information Technology (ACIT)*, IEEE, 2018, pp. 1–6. https://doi.org/10.1109/ACIT.2018.8672667

[37] T. Zhuo and Q. Sang, "Application of attention mechanism and composite convolution in handwriting recognition," *Journal of Frontiers of Computer Science & Technology*, vol. 16, no. 4, 2022.

[38] A. Hamza, S. Ren, and U. Saeed, "ET-Network: A novel efficient transformer deep learning model for automated Urdu handwritten text recognition," *PLoS ONE*, vol. 19, e0302590, 2024. https://doi.org/10.1371/journal.pone.0302590

[39] A. F. Ganai and F. Khursheed, "Computationally efficient recognition of unconstrained handwritten Urdu script using BERT with vision transformers," *Neural Comput. & Applic.*, vol. 35, pp. 24161–24177, 2023. https://doi.org/10.1007/s00521-023-08976-1

[40] M. Dhiaf, A. C. Rouhou, Y. Kessentini, and S. B. Salem, "MSdocTr-Lite: A lite transformer for full page multi-script handwriting recognition," *Pattern Recognition Letters*, vol. 169, pp. 28–34, 2023. https://doi.org/10.1016/j.patrec.2023.03.020

[41] M. R. Al-Maamari, R. Ramteke, A. M. Al-Hejri, and S. S. Alshamrani, "Integrating CNN and transformer architectures for superior Arabic printed and handwriting characters classification," *Sci. Rep.*, vol. 15, 29936, 2025. https://doi.org/10.1038/s41598-025-12045-z

[42] Y. Li, D. Chen, T. Tang, and X. Shen, "HTR-VT: Handwritten text recognition with vision transformer," *Pattern Recognition*, vol. 158, 110967, 2025. https://doi.org/10.1016/j.patcog.2024.110967

[43] S. Momeni and B. BabaAli, "A transformer-based approach for Arabic offline handwritten text recognition," *SIViP*, vol. 18, pp. 3053–3062, 2024. https://doi.org/10.1007/s11760-023-02970-9

[44] S. Aabed and A. Khairaldin, "An end-to-end, segmentation-free, Arabic handwritten recognition model on KHATT," arXiv Preprint, arXiv: 2406.15329, 2024. https://doi.org/10.48550/arXiv.2406.15329

[45] A. Elbereky, H. Elshenhab, N. Maklad, and A. Fares, "Hybrid ResNet-transformer framework for Arabic handwritten OCR in exam grading and manuscript digitization," in *Proc. 2025 7th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, IEEE, 2025, pp. 113–116. https://doi.org/10.1109/NILES68063.2025.11232091

[46] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

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

[48] E. Haber and L. Ruthotto, "Stable architectures for deep neural networks," *Inverse Problems*, vol. 34, no. 1, 014004, 2017.

[49] M. M. R. Tusher *et al.*, "Development of a lightweight model for handwritten dataset recognition: Bangladeshi city names in Bangla script," *CMC*, vol. 80, pp. 2633–2656, 2024. https://doi.org/10.32604/cmc.2024.049296

[50] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, pp. 2673–2681, 1997. https://doi.org/10.1109/78.650093

[51] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proc. the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* vol. 2, 2018, pp. 464–468.

[52] I. Fatnassi, S. Khamekhem Jemni, S. Ammar, and Y. Kessentini, "ST-WID: Self-supervised transformer for writer identification in arabic handwritten scripts," *SIViP*, vol. 19, p. 1190, 2025. https://doi.org/10.1007/s11760-025-04771-8

[53] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020. https://doi.org/10.1109/ACCESS.2020.3012542

[54] H. Scheidl, S. Fiel, and R. Sablatnig, "Word beam search: A connectionist temporal classification decoding algorithm," in *Proc. 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, IEEE, 2018, pp. 253–258. https://doi.org/10.1109/ICFHR-2018.2018.00052

[55] M. Rabi, "Convolutional Arabic handwriting recognition system based BLSTM-CTC using WBS decoder," *IJASCA*, vol. 4, 2024. https://doi.org/10.47679/ijasca.v3i2.52

[56] A. Praneetha, S. Harisa, K. Satish, and B. Vivekananda, "Handwritten text recognition using word beam search," in *Proc. 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, 2023, pp. 1878–1882. https://doi.org/10.1109/ICACCS57279.2023.10112967

[57] M. Pechwitz, H. El Abed, and V. Märgner, "Handwritten Arabic word recognition using the IFN/ENIT-database," in *Guide to OCR for Arabic Scripts*, V. Märgner and H. El Abed, Eds., Springer London, 2012, pp. 169–213. https://link.springer.com/chapter/10.1007/978-1-4471-4072-6_8

[58] Y. Hamdi, H. Boubaker, and A. M. Alimi, "Data augmentation using geometric, frequency, and beta modeling approaches for improving multi-lingual online handwriting recognition," *IJDAR*, vol. 24, pp. 283–298, 2021. https://doi.org/10.1007/s10032-021-00376-2

[59] C. Luo, Y. Zhu, L. Jin, and Y. Wang, "Learn to augment: Joint data augmentation and network optimization for text recognition," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2020, pp. 13743–13752. https://doi.org/10.1109/CVPR42600.2020.01376

[60] A. A. Almisreb, N. Md Tahir, S. Turaev, M. A. Saleh, and S. A. M. Al Junid, "Arabic handwriting classification using deep transfer learning techniques," *JST*, vol. 30, pp. 641–654, 2022. https://doi.org/10.47836/pjst.30.1.35

[61] G. S. Nugraha, M. I. Darmawan, and R. Dwiyansaputra, "Comparison of CNN's architecture GoogleNet, AlexNet, VGG-16, Lenet-5, Resnet-50 in Arabic handwriting pattern recognition," *KINETIK*, 2023. https://doi.org/10.22219/kinetik.v8i2.1667

[62] N. A. Shiferaw *et al.*, "Handwritten Amharic character recognition through transfer learning: Integrating CNN models and machine learning classifiers," *IEEE Access*, vol. 13, pp. 52134–52148, 2025. https://doi.org/10.1109/ACCESS.2025.3553199

[63] S. Rouabhi and R. Tlemsani, "Multimodal classification of the Arabic alphabet: an artificial intelligence approach for handwriting, sign language, and braille," in *Advances in Educational Technologies and Instructional Design*, E. Cela, N. R. Vajjhala, and M. M. Fonkam, Eds., IGI Global, 2024, pp. 255–284. https://doi.org/10.4018/979-8-3693-7220-3.ch010

[64] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. 2013 12th International Conference on Document Analysis and Recognition*, IEEE, 2013, pp. 1484–1493. https://doi.org/10.1109/ICDAR.2013.221

[65] Student, "The probable error of a mean," *Biometrika*, vol. 6, p. 1–25, 1908. https://doi.org/10.2307/2331554
[66] A. Waly, B. Tarek, A. Feteha, R. Yehia, G. Amr, and A. Fares, "Arabic handwritten document OCR solution with binarization and adaptive scale fusion detection," in *Proc. 2024 6th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, IEEE, 2024, pp. 316–319. https://doi.org/10.1109/NILES63360.2024.10753216
[67] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8610–8617. https://doi.org/10.48550/ARXIV.1811.00751
[68] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. the 23rd International Conference on Machine Learning*, ACM Press, 2006, pp. 369–376. https://doi.org/10.1145/1143844.1143891
[69] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "On the relationship between self-attention and convolutional layers," arXiv Preprint, arXiv:1911.03584, 2019. https://arxiv.org/abs/1911.03584
[70] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," arXiv Preprint, arXiv:2010.11929, 2020. https://arxiv.org/abs/2010.11929
[71] P. Kaliosis and J. Pavlopoulos, "Learning to align: Addressing character frequency distribution shifts in handwritten text recognition," arXiv Preprint, arXiv:2506.09846, 2025. https://arxiv.org/abs/2506.09846
[72] N. Alrobah and S. Albahli, "Arabic handwritten recognition using deep learning: A survey," *Arab. J. Sci. Eng.*, vol. 47, pp. 9943–9963, 2022. https://doi.org/10.1007/s13369-021-06363-3
[73] M. E. Schubert, D. Langerman, and A. D. George, "Benchmarking inference of transformer-based transcription models with clustering on embedded GPUs," *IEEE Access*, vol. 12, pp. 123276–123293, 2024. https://doi.org/10.1109/ACCESS.2024.3426471
[74] M. Elleuch, N. Tagougui, and M. Kherallah, "Optimization of DBN using regularization methods applied for recognizing Arabic handwritten script," *Procedia Computer Science*, vol. 108, pp. 2292–2297, 2017. https://doi.org/10.1016/j.procs.2017.05.070
[75] M. Elleuch, R. Maalej, and M. Kherallah, "A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition," *Procedia Computer Science*, vol. 80, pp. 1712–1723, 2016. https://doi.org/10.1016/j.procs.2016.05.512
[76] M. M. AL-Taee, S. B. H. Neji, and M. Frikha, "Handwritten Arabic words detection using faster R-CNN in IFN/ENIT dataset," *Bulletin EEI*, vol. 13, pp. 3568–3578, 2024. https://doi.org/10.11591/eei.v13i5.8189
[77] R. Maalej and M. Kherallah, "New MDLSTM-based designs with data augmentation for offline Arabic handwriting recognition," *Multimed. Tools Appl.*, vol. 81, pp. 10243–10260, 2022. https://doi.org/10.1007/s11042-022-12339-8